Introduction

CS2520/TELCOM2321

Wide Area Networks

Dr. Taieb Znati

Computer Science Department

University of Pittsburgh

Course Objectives

- The objective of the course is to provide an in-depth understanding of architectural principles, fundamental concepts and basic techniques underlying the design and implementation of the Internet and its protocols
  - Whenever possible and useful, a quantitative approach is used

Course Approach

- The focus is on advanced topics related to the design of large-scale, evolvable, reliable and robust internet
  - In-depth discussion of network design problems, models and approaches to a variety of design and implementation issues related to network protocols for large-scale internets
    - When possible, identify open research problems
  - Explore the Internet design principles and tradeoffs and understand adopted solutions in context – goals, assumptions and what problems are solved
  - Gain hands-on experience through problem solving and project development

Course Topics

- Network Architecture Fundamentals
  - Layering, Protocols and Interfaces
- Network Design Principles and Mechanisms
  - Design Tradeoffs and Performance
- Internet Architecture Overview
  - Naming, Addressing and Packet Forwarding
- Internet Routing Architecture
- Internet Congestion Control Framework
- Internet Traffic Management and QoS Support
  - Resource Allocation and Scheduling
  - Traffic Shaping and Monitoring
  - IntServ and DiffServ Frameworks
  - RSVP, MPLS, SDN and other protocols
- Future Internet Directions

Recommended Readings

- **James Kurose** and **Keith Ross**: *Computer Networking, A Top Down Approach*, Addison Wesley
- **William Stallings**: *Data & Computer Communications*, Prentice Hall
- **Larry Peterson and Bruce Davie**, : *Computer Networks: A Systems Approach*, Morgan Kauffman
- **Douglas E. Comer**, **David L. Stevens**: *Internetworking with TCP/IP*, Prentice Hall

Network Programming Useful References

- **W. Richard Stevens**, *TCP/IP Illustrated, Volume 1: The Protocols*
- **W. Richard Stevens, Bill Fenner** and **Andrew Rudoff,** *Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition)*,
- Online Resources
  - Search for “Socket Programming”

Course Requirements

- Homework Assignments
  - Problem solving and programming assignments
- Paper Reviews
  - Students are required to read state-of-the-art papers, write critique of the paper, and describe potential open problems for future research.
- Group Project
  - Two students per group
- Exams – Midterm and Final
- Participation in class discussions

Project Logistics

- The objective is to get a hands-on experience and a perspective on how to do design and deploy network protocols
- Requirements
  - Project Design,
  - Implementation, experimentation and Analysis, and
  - Final Report and Demonstration
- Grading
  - Project Preliminary Design (10%)
  - Project Implementation and Analysis (60%)
  - Project Final Report and Demonstration (20%)

Grading Policy

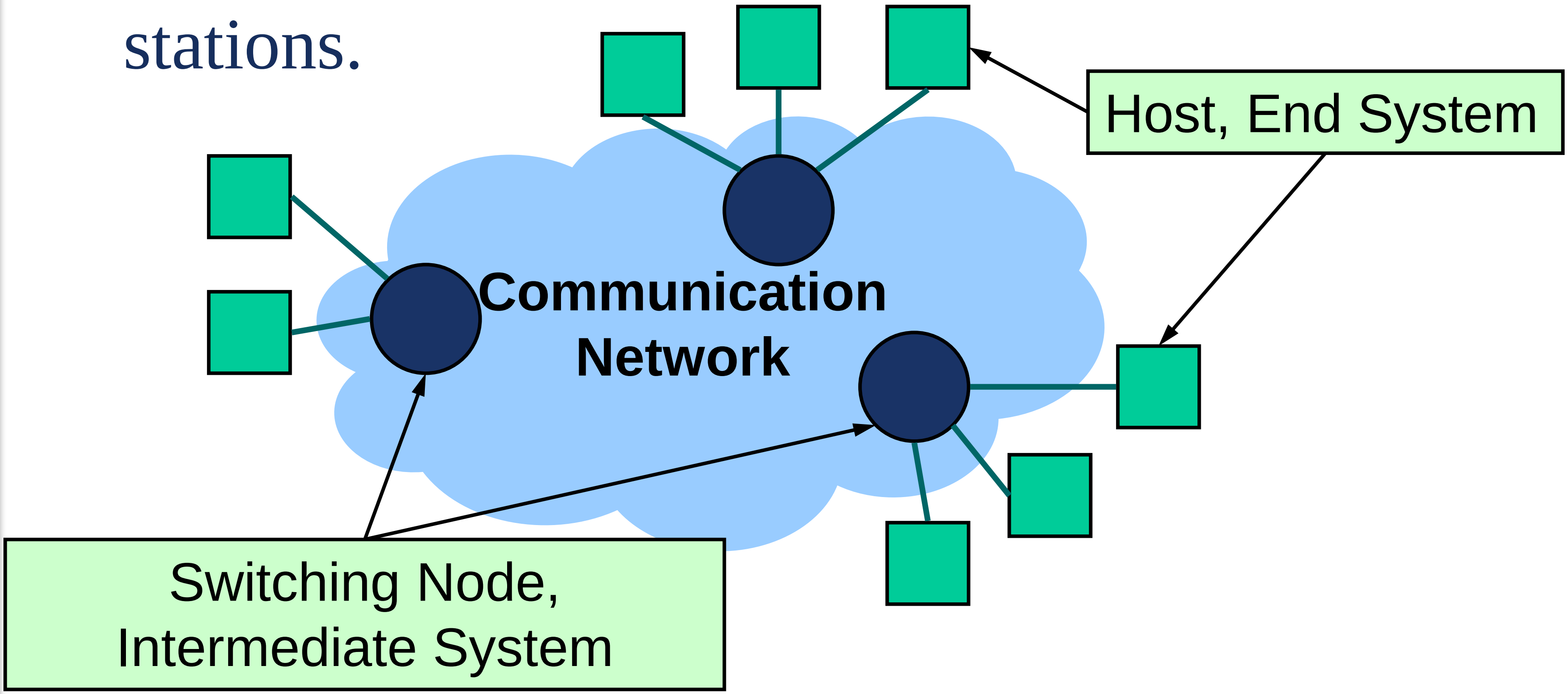
- Homework: 10%
- Project: 30%
- Midterm: 30%
- Final: 30%
- Class participation to round up or down!

Computer Networks

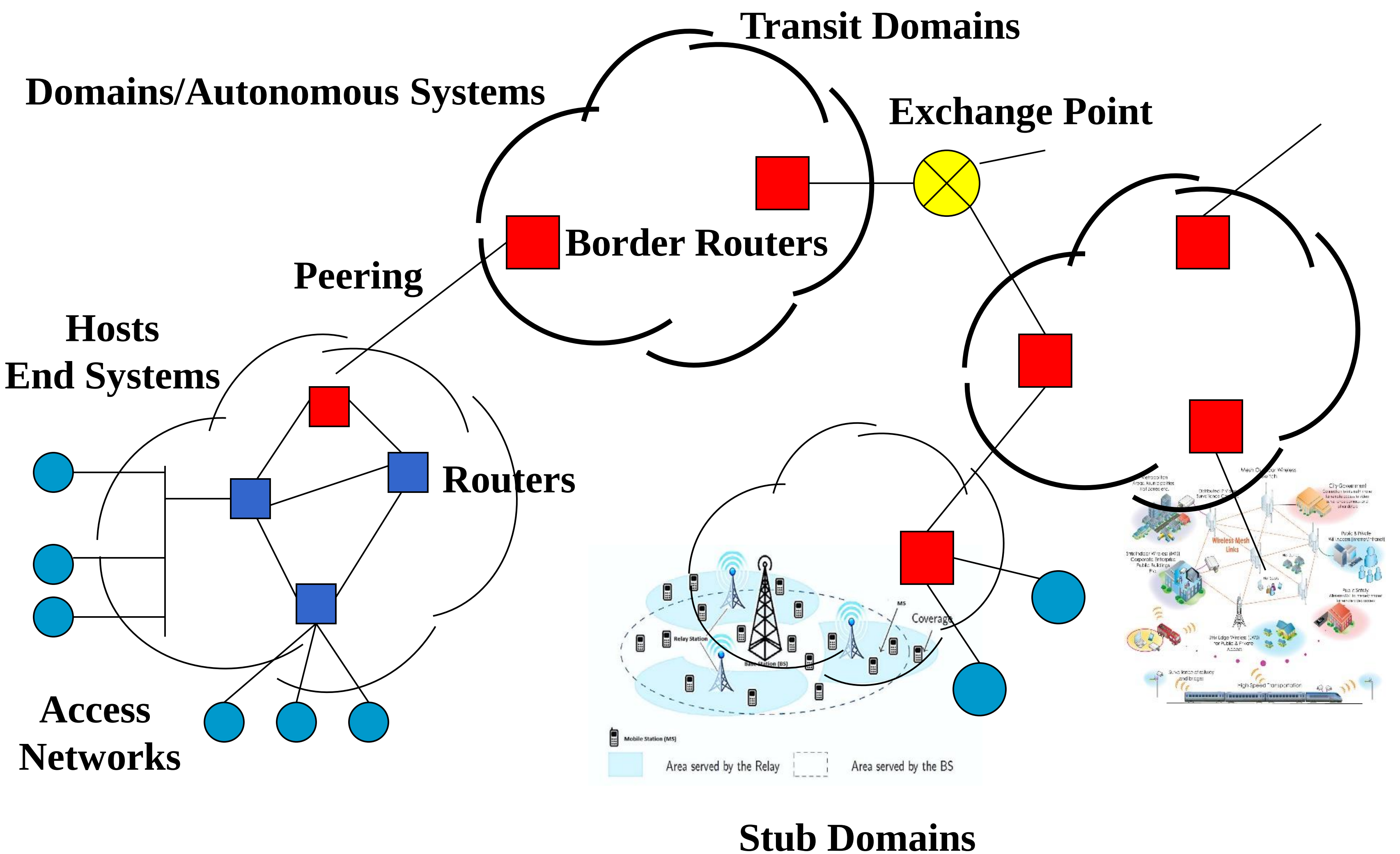
Architecture, Protocols and Design Principles

Communication Networks

- Network, a collection of nodes capable of transporting data between pairs of attached stations.



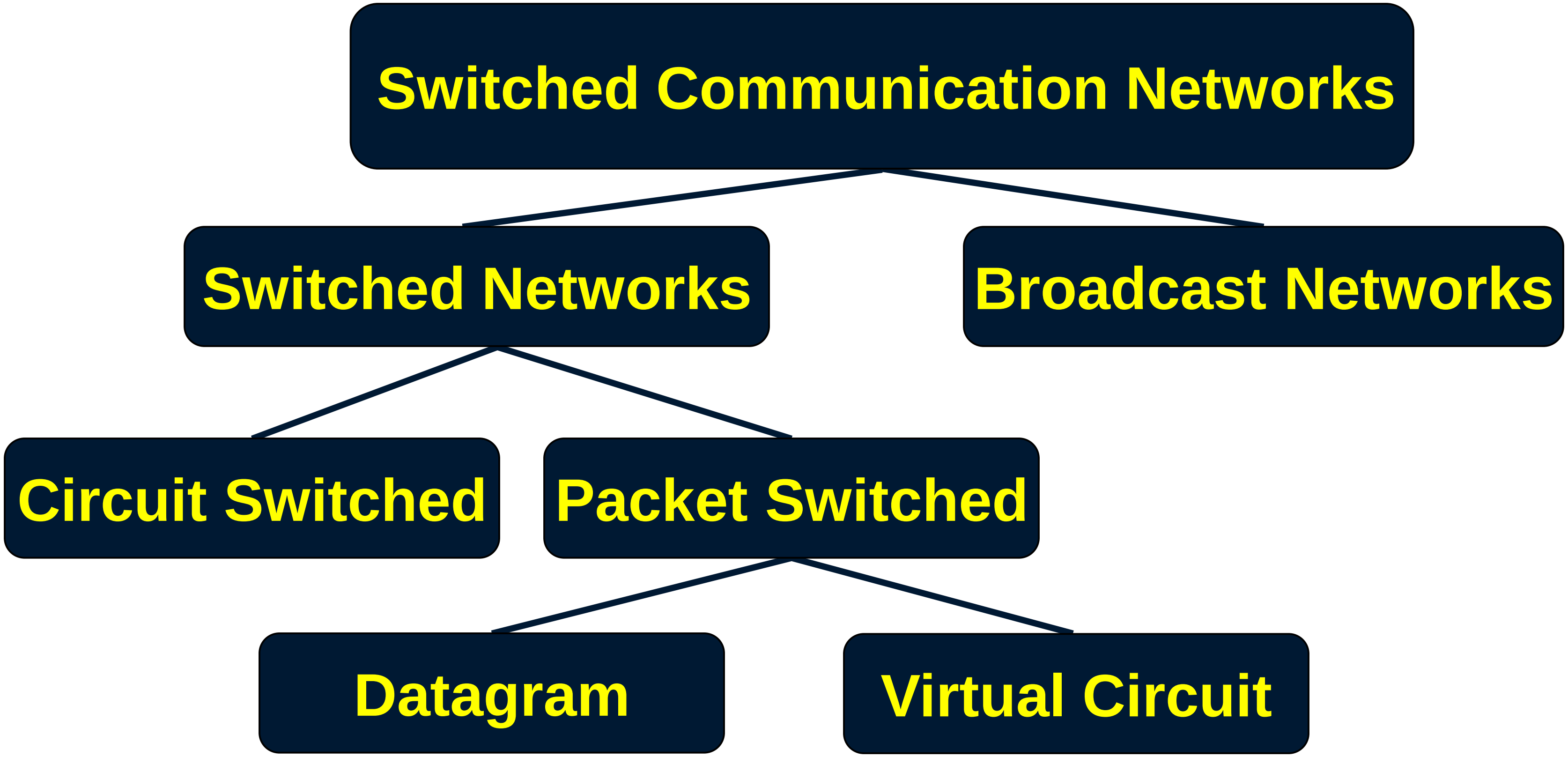
WAN Structure



COMPUTER NETWORKS

SWITCHING TECHNIQUES

Switching Techniques – Taxonomy



Broadcast Networks

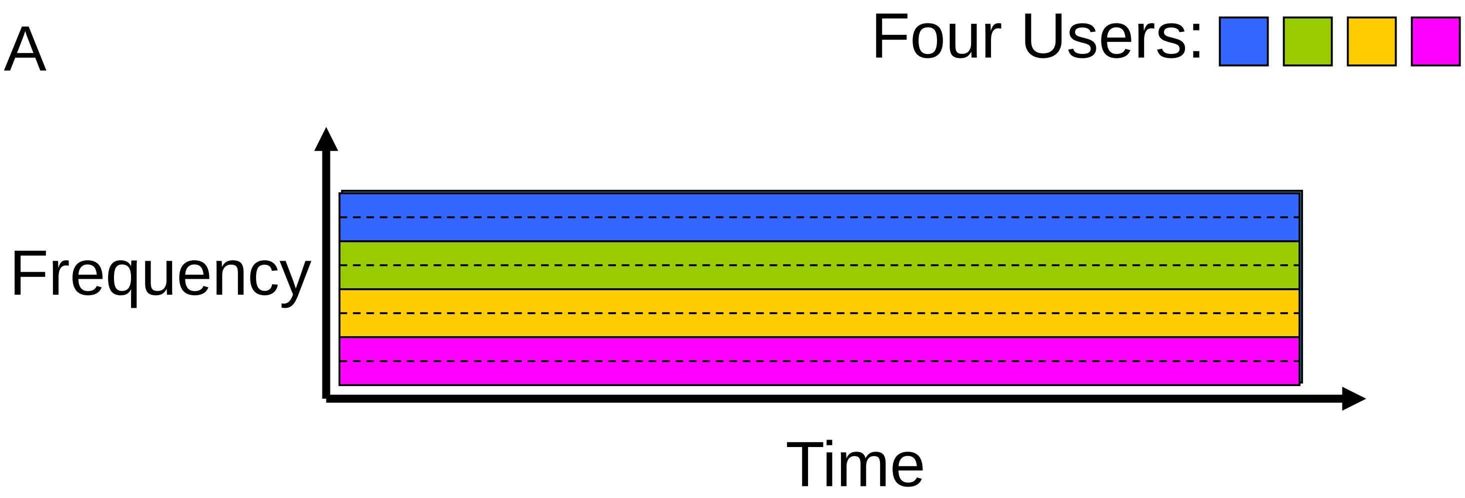
- Information transmitted by any node is received by every node
  - Typically used in LANs – wired and wireless
    - No need for intermediate node,
- Several limitations
  - Limited transmission range
  - Privacy of communication
  - Media Access Control is required to coordinate access to the shared communication medium

Circuit Switching

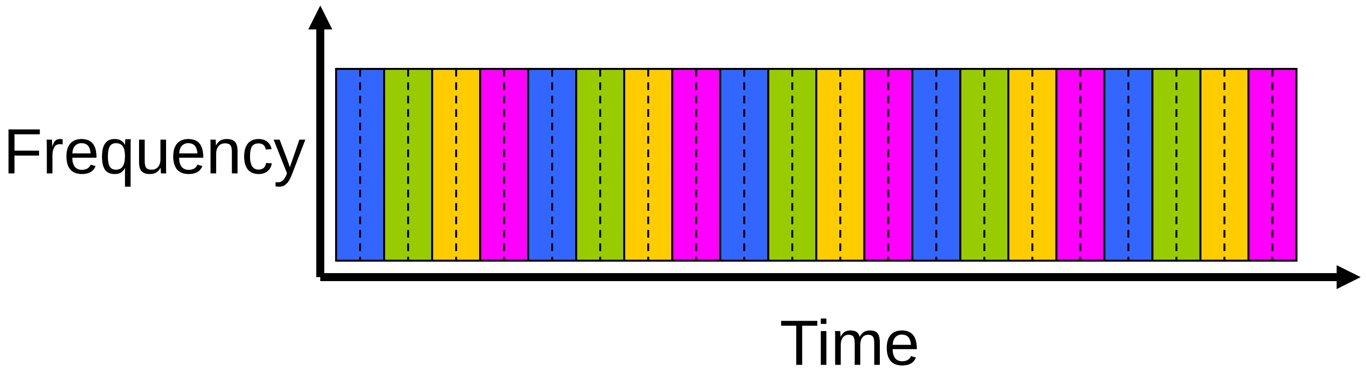
- Exclusive dedication of a portion of the available bandwidth to carry traffic between source and destination
  - Bandwidth is allocated using :
    - Frequency Division Multiplexing
    - Time Division Multiplexing
  - Incoming data are switched to the appropriate outgoing channel without delay
  - Telephone network, a typical example

Circuit Switching: FDMA and TDMA

FDMA

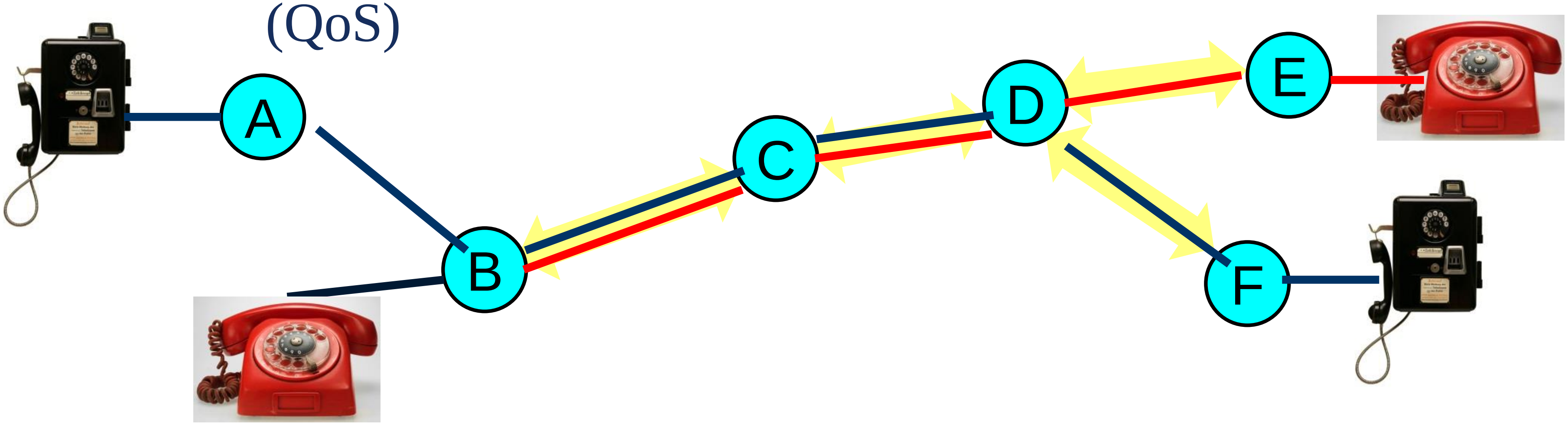


TDMA



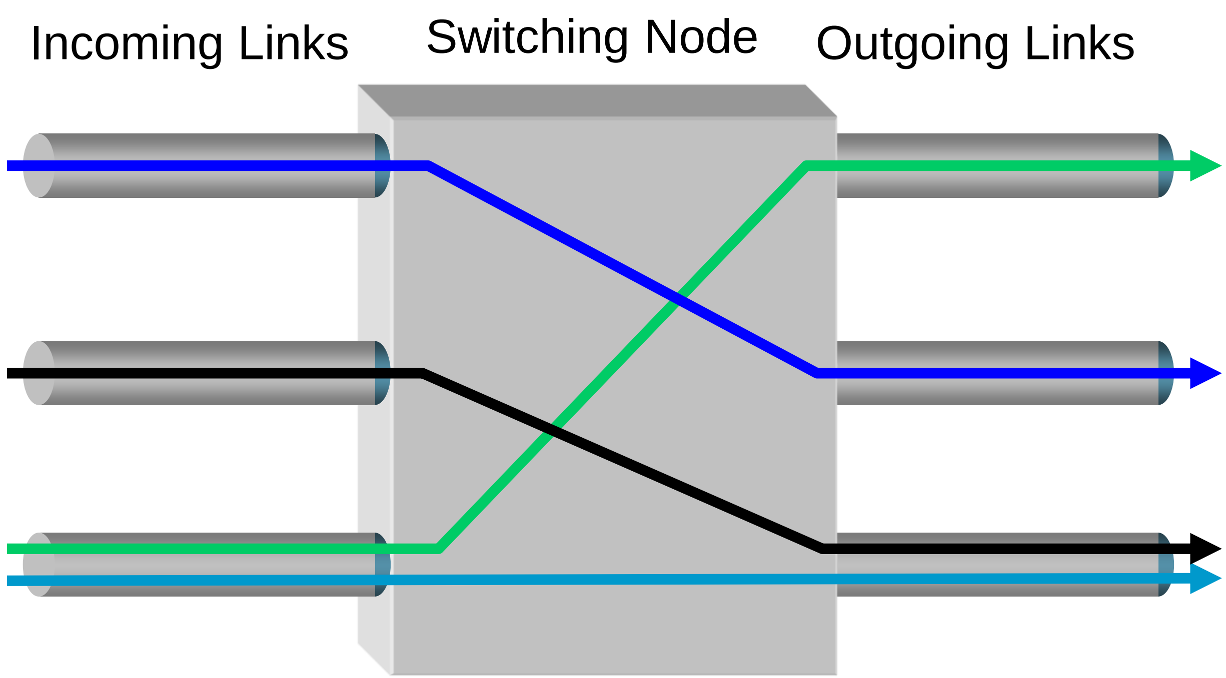
Circuit Switched Networks

- Typically, used in telephone networks – POTS
  - All resources needed by a “call”, such as communication links, buffers and processing, are reserved for the entire duration of the call
    - Resource reservation guarantees “quality of service” (QoS)

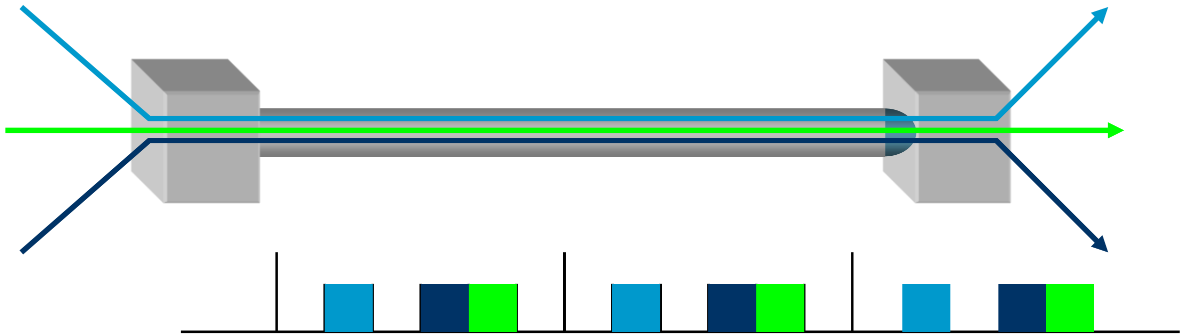


Circuit Switching

- A node (**switch**) in a circuit switching network switches data from incoming links to outgoing links

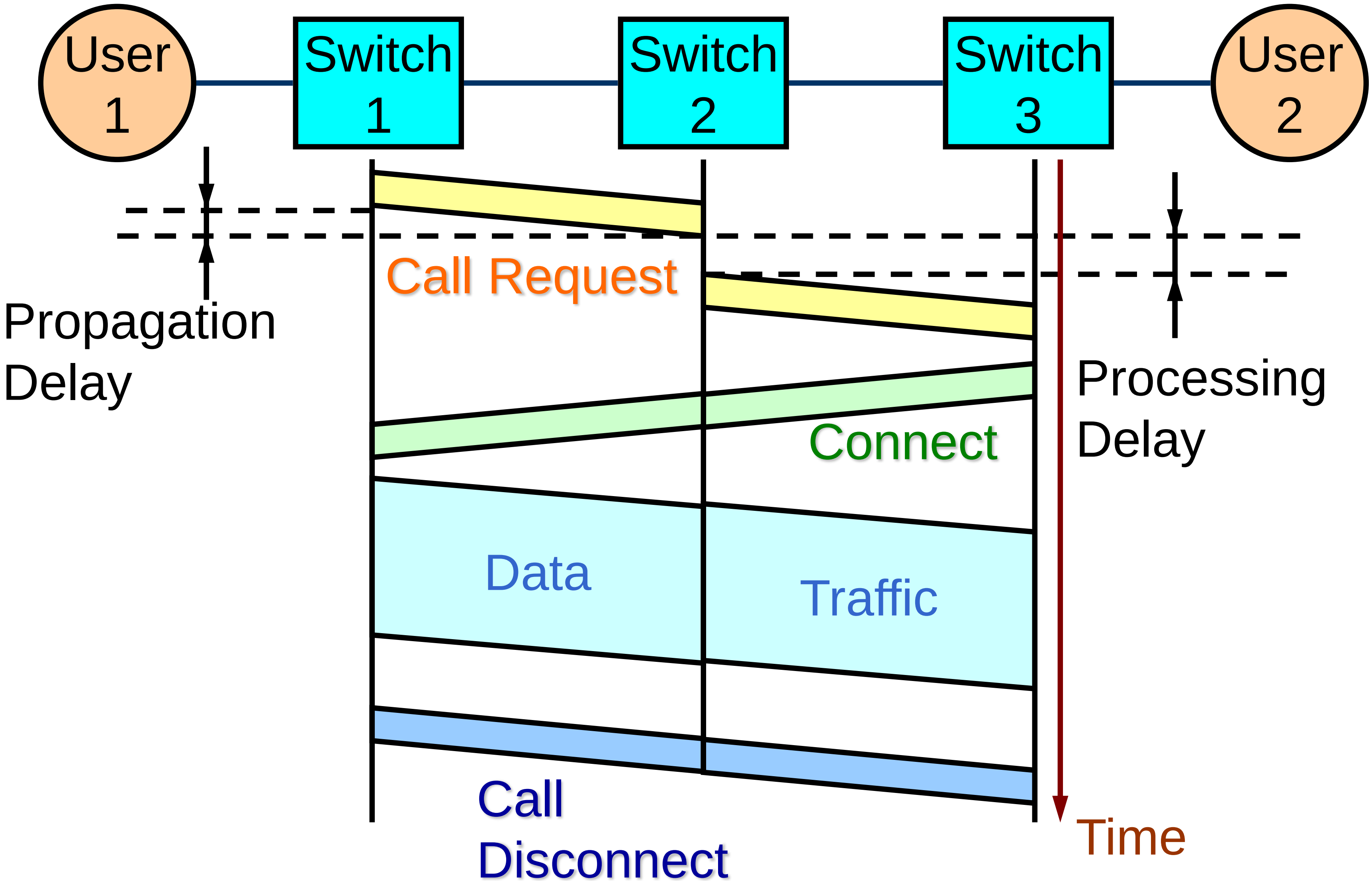


TDM-based Circuit Switching Multiplexing and De-Multiplexing



- Relative slot position inside a frame **determines** which conversation the data belongs to
  - Needs synchronization between sender and receiver
  - Slots may remain empty
  - Dynamic allocation of slot, possible but difficult

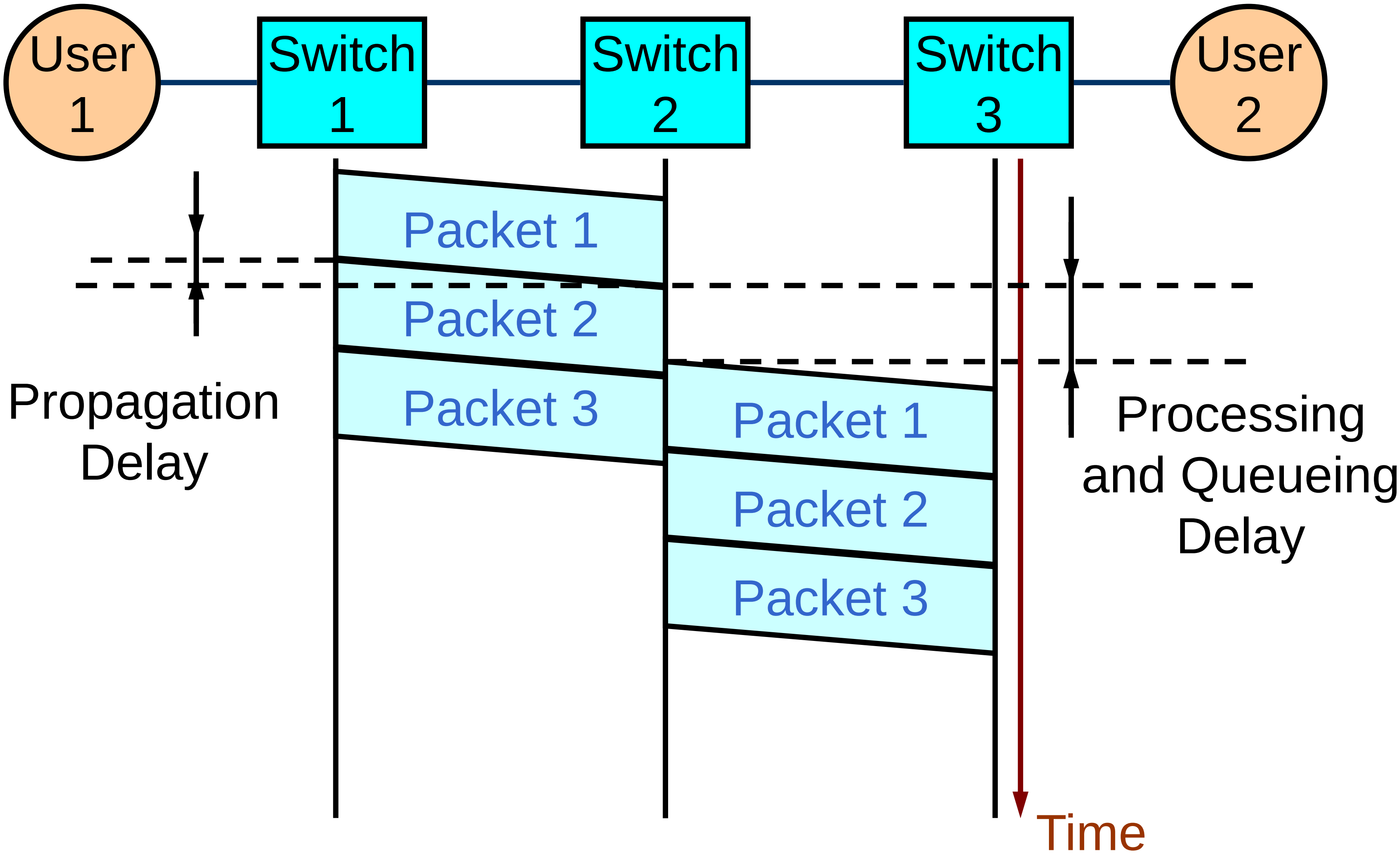
Circuit Switching Timing Diagram



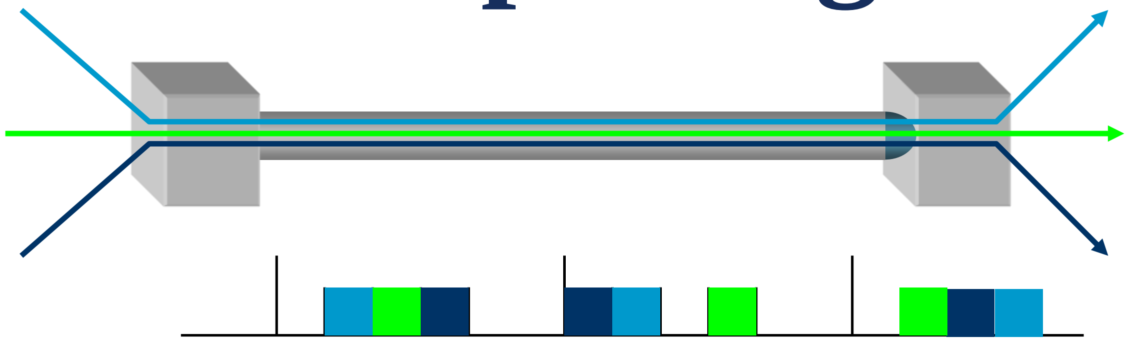
Packet Switching Networks

- Data enters the network in the form of "packets"
  - Maximum packet length for transmission is imposed.
  - Any data exceeding the maximum length is broken into shorter "packets".
- Packets traversing a network share netwrok resources with other packets – statistical sharing
- Demand for resources may exceed the amount of the available resources
  - Contention occurs, and may result in heavy congestion

Packet Switching Timing Diagram



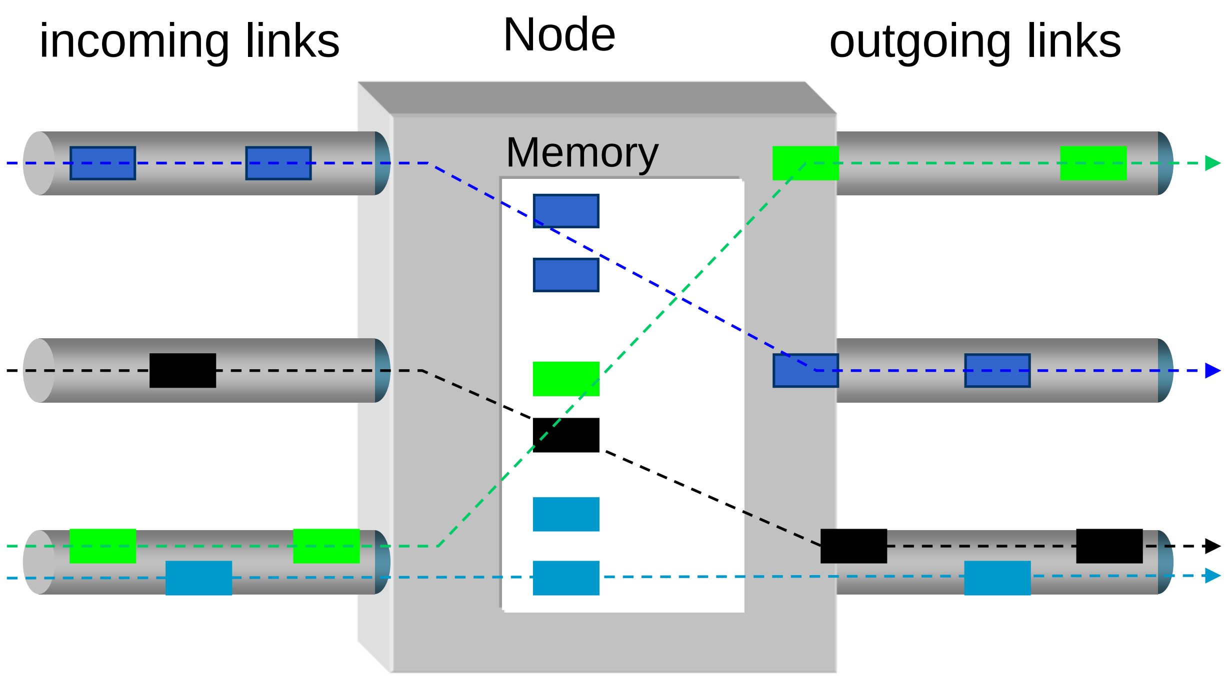
Packet Switching Multiplexing and Demultiplexing



- Data from any sender can be transmitted at any given time
- Headers are used to differentiate traffic from different senders

Packet Switching

- Store-and-Forward
  - At each node the entire packet is received, stored, and then forwarded to the next node



Packet Delays

- Propagation delay: time for the signal to propagate from the sender to the receiver
  - Determined by the wave propagation speed: 200 m/µs in a wire
- Transmission delay: time needed to transmit the signal representing a block of data
  - Determined by the data rate and the length of the packet,
- Processing and queueing delay: time needed to process the packet and the time the packet has to wait until the link becomes available: store-and-forward.
  - Determined by the router processing speed and the network load

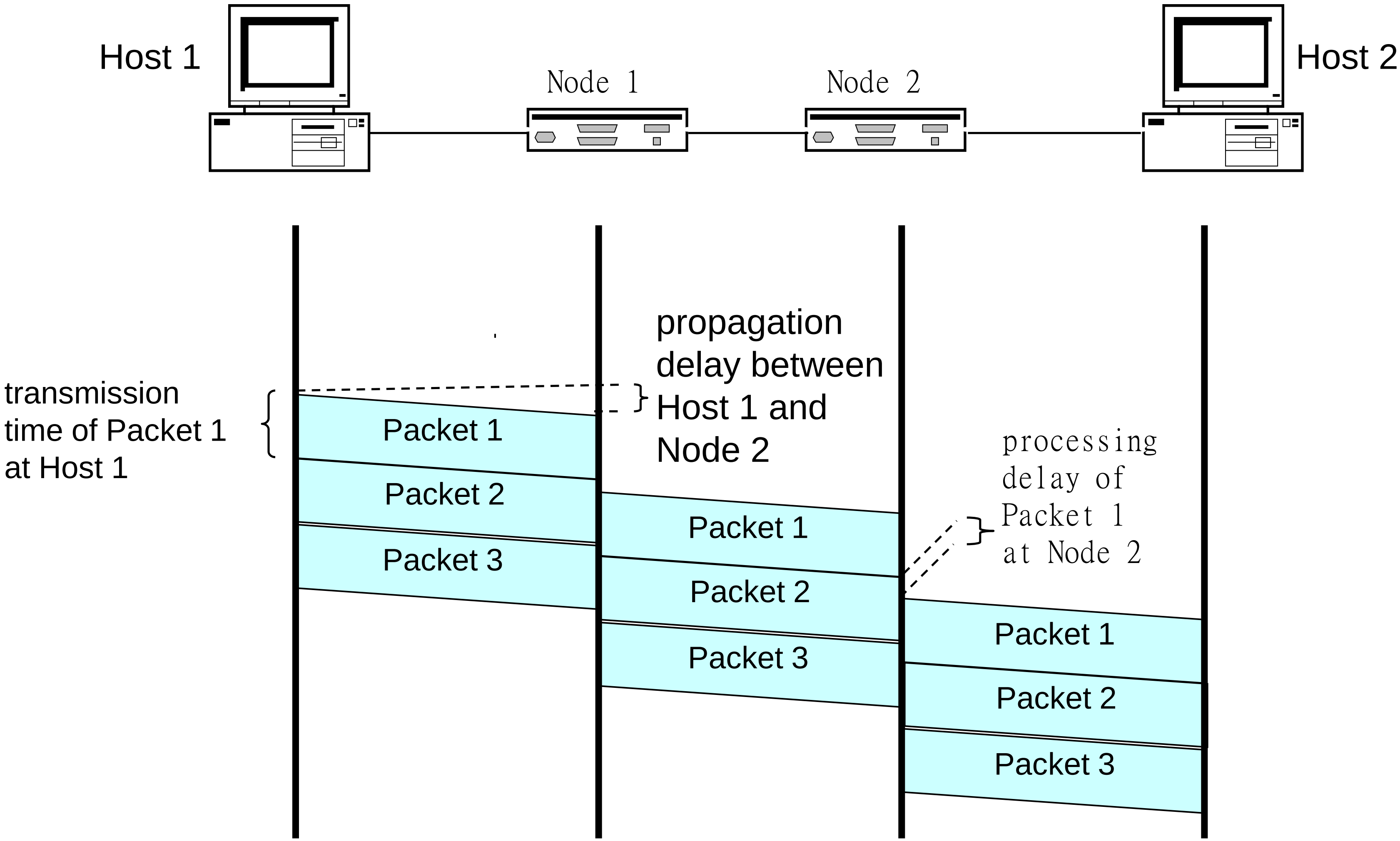
Packet-Switching vs. Circuit-Switching

- Ability of packet-switching to exploit statistical multiplexing:
  - Efficient bandwidth usage ─ ratio between peek and average rate is 3:1 for audio, and 15:1 for data traffic
    - Consider a 1 Mbps communication link, where each user requires 100 kbps when transmitting, but sends 10% of the time,
      - Circuit switching:
        - Each caller is allocated 100 kbps capacity,
        - At most 10 callers are supported.
      - Packet switching:
        - With 35 ongoing calls, probability that 10 or more callers are simultaneously active is about 0.00174,
        - Can support many more callers, with small probability of contension
- If user traffic is “bursty” (on/off), then packet switching can be more efficient than circuit switching.

Packet Switching Techiques

- Two basic approaches to packet switching are common:
  - Datagram packet switching,
  - Virtual circuit packet switching.

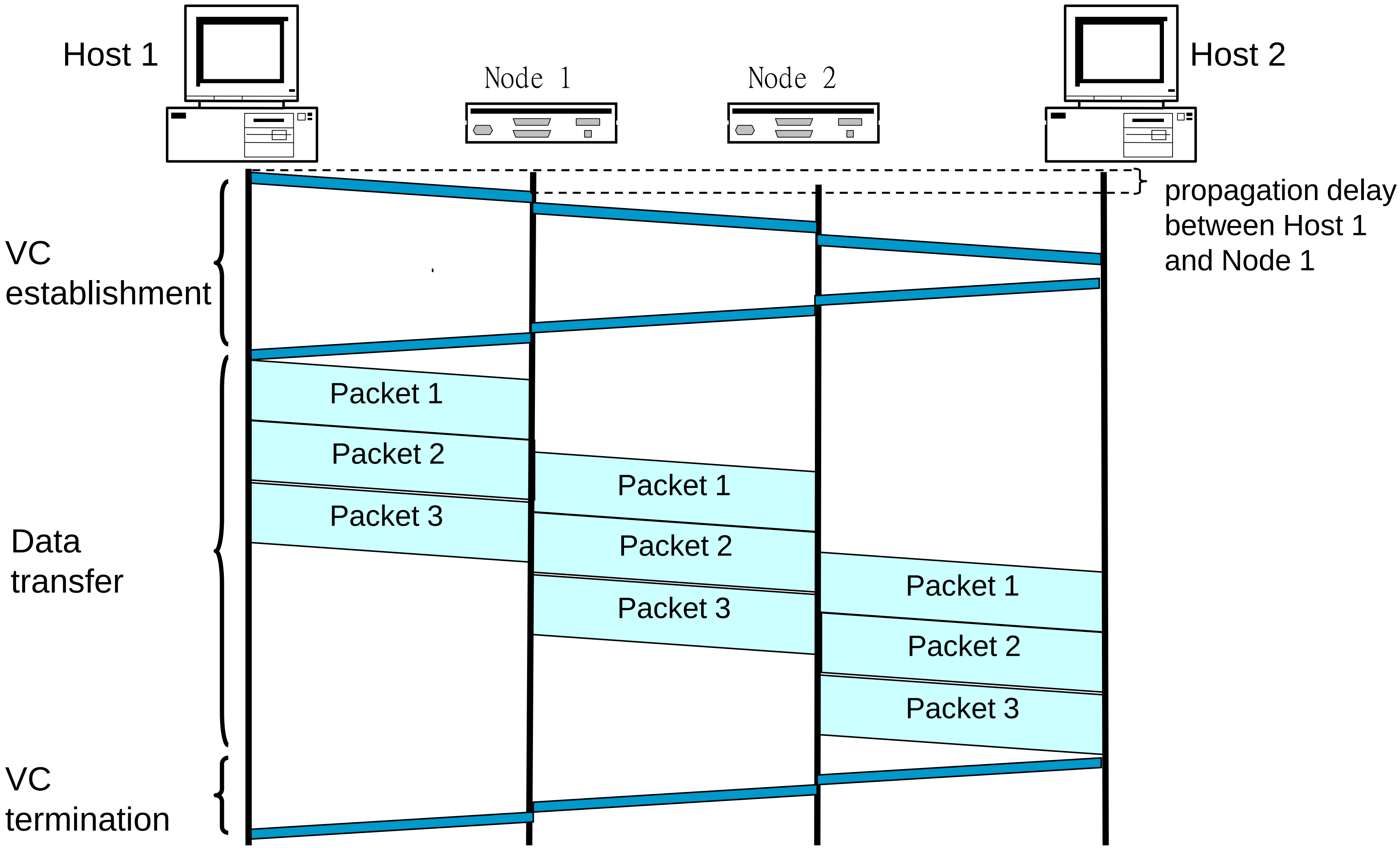
Timing of Datagram Packet Switching



Datagram Packet Switching

- Datagram Packet Switching does not require a call setup
- For short transactions, it may be faster
- Individual datagrams are routed independently
  - Increases processing overhead at the router
    - Routing table lookups

Timing of Virtual Circuit Packet Switching



Virtual Circuit Service Characteristics

- Reliable delivery of information
  - Powerful error control
  - Sequencing of packets
  - Detection and suppresion of duplicates
- Congestion control minimizes queueing delays
  - Delays, however, are more variable than they are with dedicated circuits
- Enhanced security

Datagrams vs. Virtual Circuits

- "Best Effort" vs. "Reliable" service
- Can the network be totally trusted ?
- Where should reliability belong?

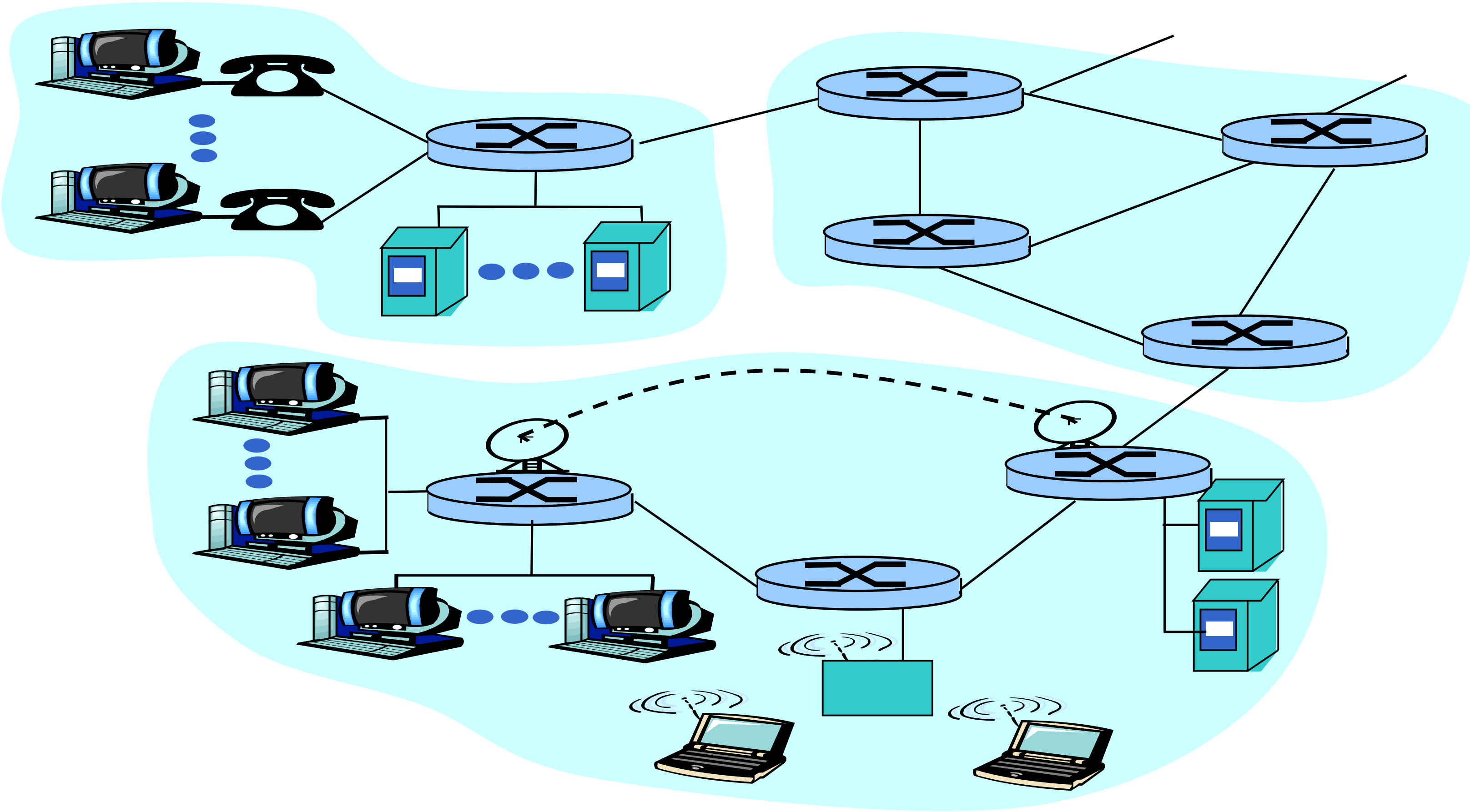
Datagram Service Characteristics

- The network makes a "best effort" attempt to deliver the packets
  - Each packet is treated as a separate entity with no prior route determination
    - Packets may follow different paths to destination
  - No guarantees for reliable delivery
    - Packets may be lost, duplicated, or may arrive out of order
- The network relies on the user application to enhance the basic datagram service

“Statefull” vs. “Stateless”

- Connection oriented networks require full state management
  - Establishment of a new connection causes a state change in the switch
  - A direct side effect, fate sharing
    - Fate of the end-to-end connection depends on the state of intermediate nodes
- Connectionless network are stateless
  - Simple and more robust
  - Only task required is to maintain routing tables

Internetworking



Internetworking

- The main goal is to provide a universal network formed out of physically different networks.
- Internetworking involves complex issues:
  - Different addressing and naming schemes
  - Different routing techniques
  - Different congestion control techniques
  - Different hardware interfaces
  - Connection oriented vs connectionless services
  - Different data unit sizes
  - Different error control techniques

Network Design Principles

- Network architectures define the standards and techniques for designing and building communication systems for computers Networks
- Two key design decisions:
  - How the network protocol stack is structured?
    - **Layering** for modularity, ease of abstraction and reuse
  - What functionality to put in each layer?
    - **End-to-End argument**

Network Design

Layering

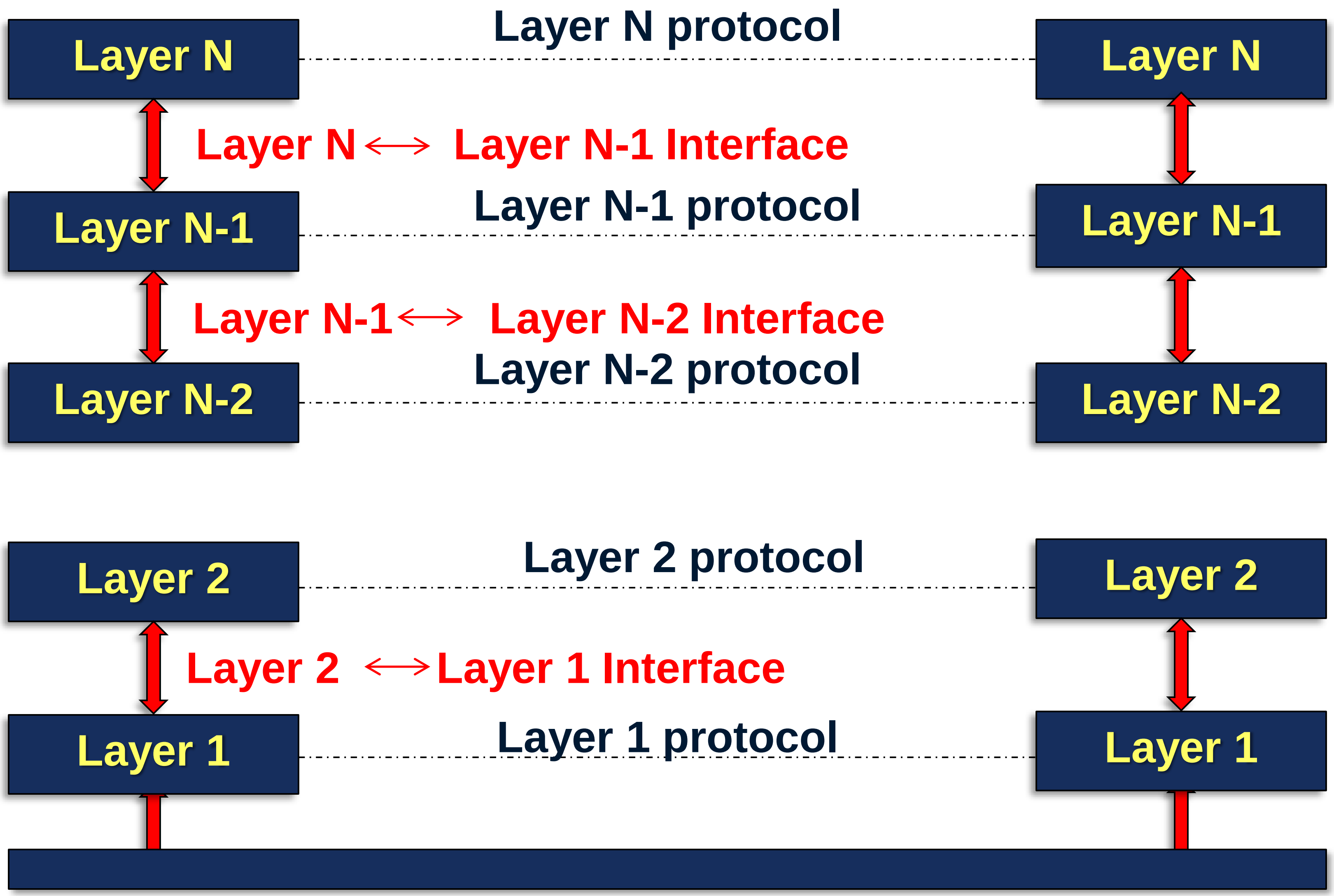
Network Design – Layering

- To reduce design complexity, most networks are organized into layers, where each layer enhances the service of the layer below
  - **Added value service**
- Two layer-n entities in different computers use a peer-to-peer layer-n protocol to communicate with each other
- Two adjacent layers within the same computer communicate through an interface
  - The interface defines the primitive operations and services lower layer provides to upper layer

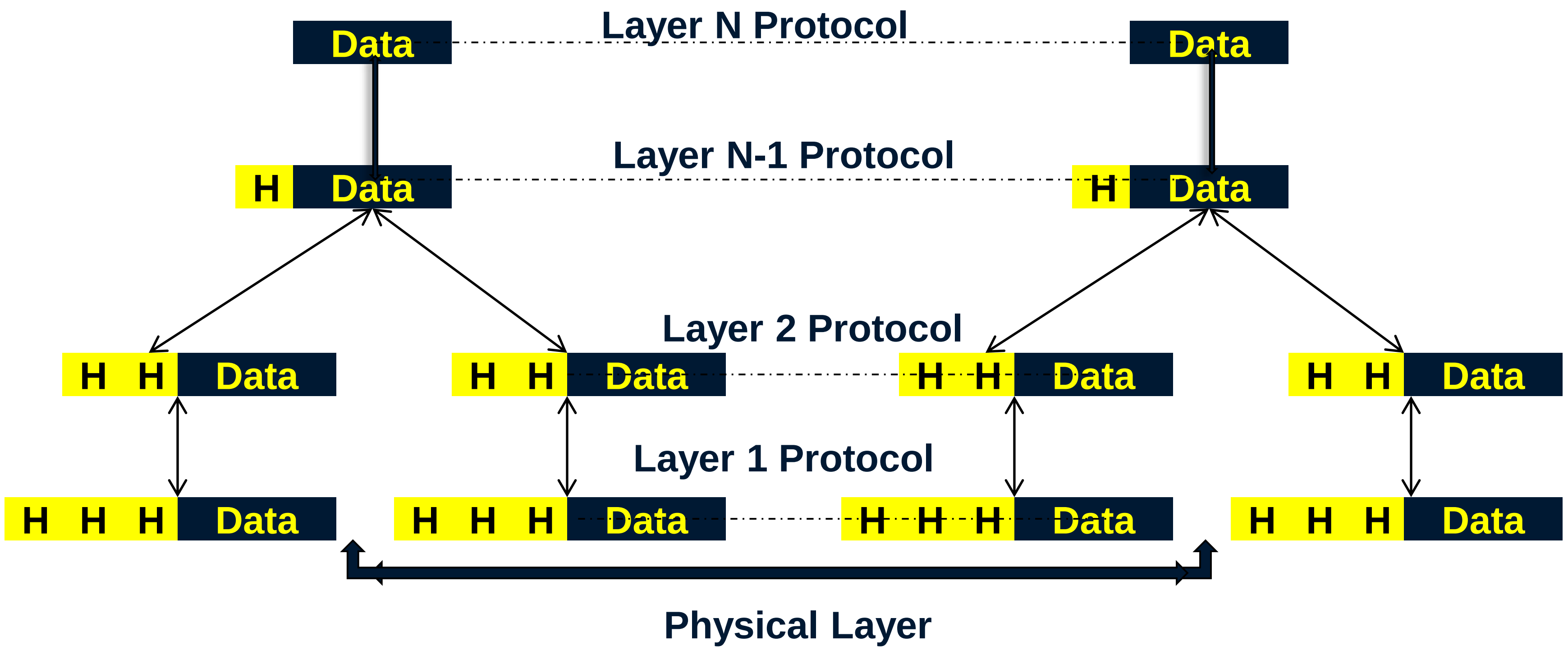
Network Architecture

- The set of layers and protocols define the protocol architecture
  - Neither the details of the implementation nor the specification are part of the protocol architecture
    - The implementation, however, must obey the appropriate protocol
  - Interfaces on all machines need not be the same, as long as the protocols are correctly implemented
    - Interfaces are not visible from the outside

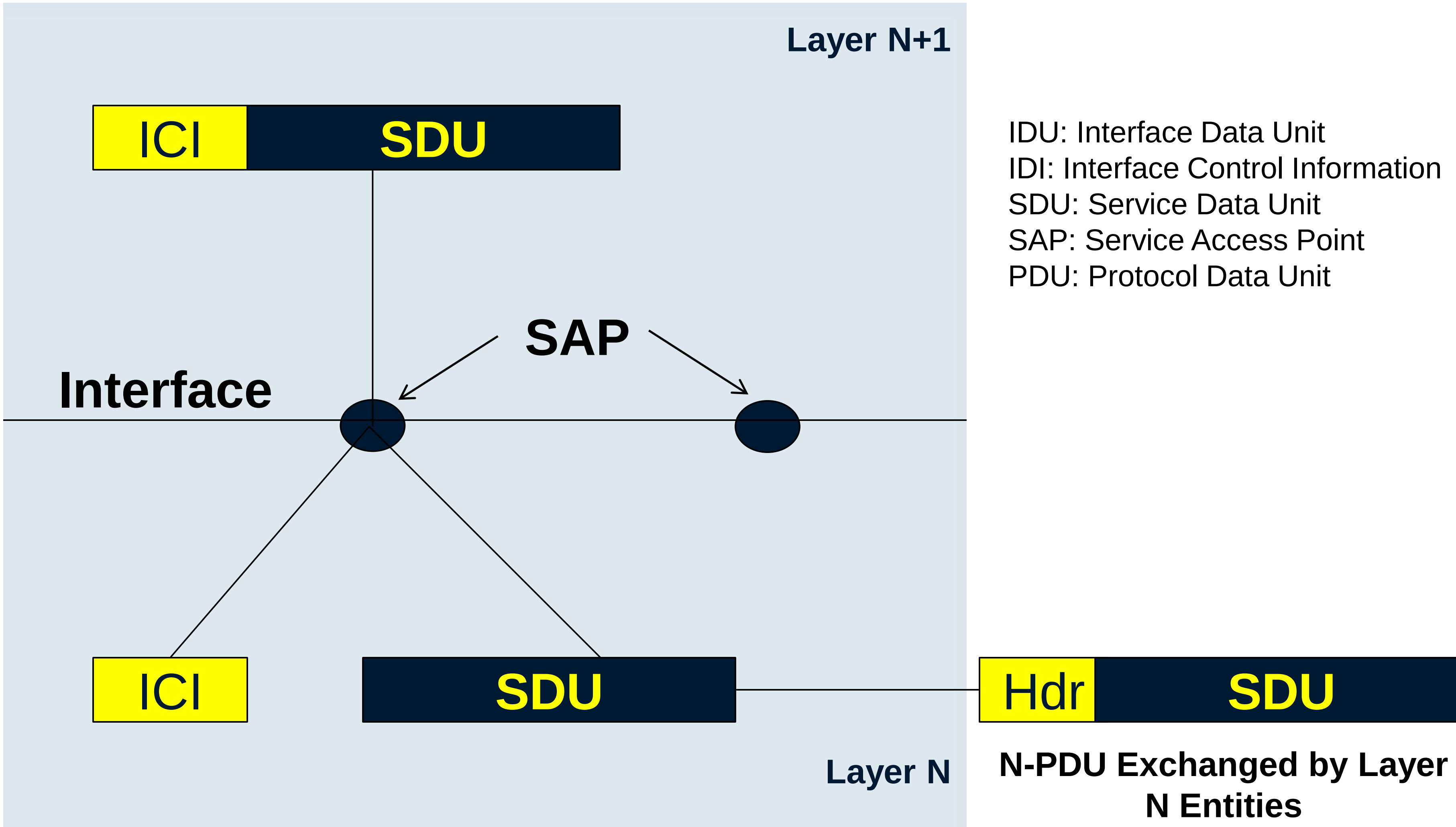
Network Architecture



Information Flow



Layers Interface Relation



IDU: Interface Data Unit
IDI: Interface Control Information
SDU: Service Data Unit
SAP: Service Access Point
PDU: Protocol Data Unit

Network Design

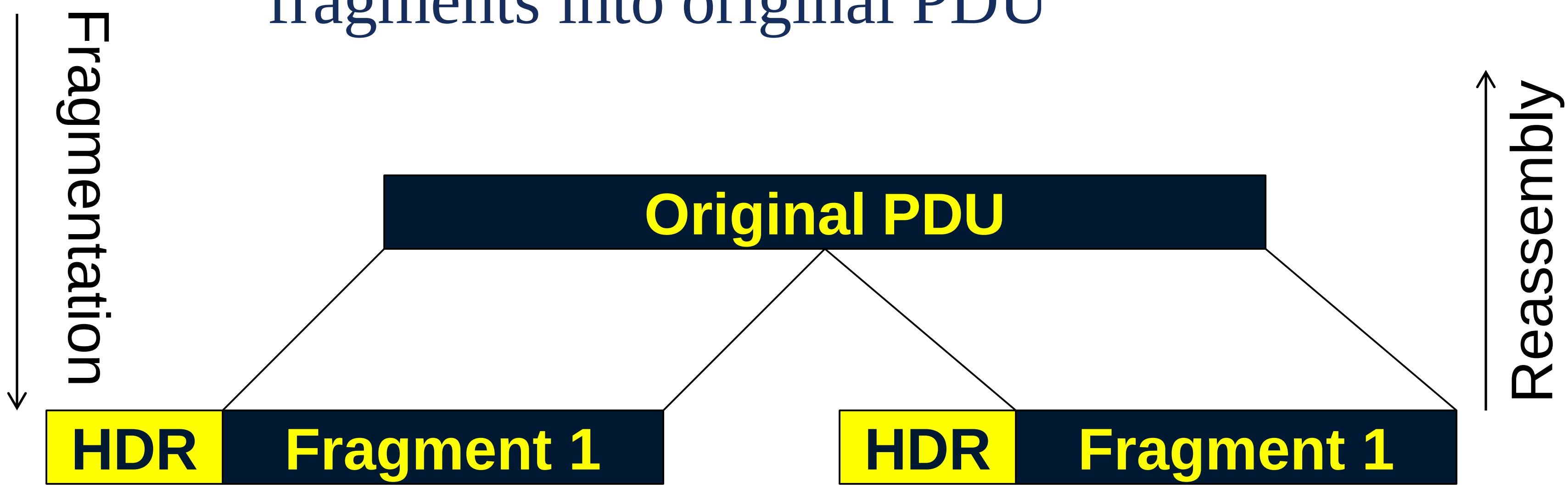
Layer Functions

Network Protocol Functions

- Connection Control
  - Connectionless vs. Connection oriented
- Segmentation and Reassembly
- Encapsulation
- Ordered Delivery
- Flow and Congestion Control
- Error Control
- Addressing
- Multiplexing
- Transmission Services

Segmentation and Reassembly

- Segmentation is the process of breaking a PDU into smaller PDUs
  - Reassembly is the process of gathering all fragments into original PDU



Reasons for Segmentation

- Network may impose limits on data size
  - X.25 packet is limited to 128 bytes
  - Ethernet packet is limited to 1526 bytes
  - ATM cell payload is limited to 48 bytes

Segmentation Benefits

- Error control may be more efficient with smaller PDUs
  - In case of errors, only a small PDU need to be retransmitted
- More equitable access to shared links
  - May result in shorter average delay
- Smaller PDUs require smaller buffers
- Checkpoints and restart-recovery operations may be required at different points in time
  - Data transfer must be synchronized to avoid losses

Segmentation Disadvantages

- Each fragment carries a fixed minimum amount of control information
  - Smaller fragments result in larger overhead
- Fragment arrivals causes interrupts that must be serviced
  - Smaller fragments result in larger number of interrupts
- Fragments require processing at different levels
  - Smaller fragments result in larger processing overhead

Encapsulation

- Each PDU contains not only data but control information
  - Some PDUs may contain only control information
- Control information contains
  - Addressing information
  - Error detection code
  - Protocol control functions

Ordered Delivery

- PDUs may not arrive in the order of they have been transmitted
- Maintaining ordered delivery of data may be achieved through PDU unique identifiers
  - IDs are assigned sequentially in increasing order modulo a “Max_Number”
    - The value of Max_Number may need to be twice the maximum number of outstanding PDUs
      - Wrap-around problem

Flow Control and Congestion Control

- Flow Control regulates the amount of data transfer between sender and receiver
  - Stop-and-Wait, simplest form of flow control
    - Data link layer protocol to control flow between adjacent nodes
  - Sliding-window, a more efficient credit based mechanism
    - Transport layer protocol
- Congestion control regulates network traffic
  - More complex task which requires a level of cooperation between hosts and routers

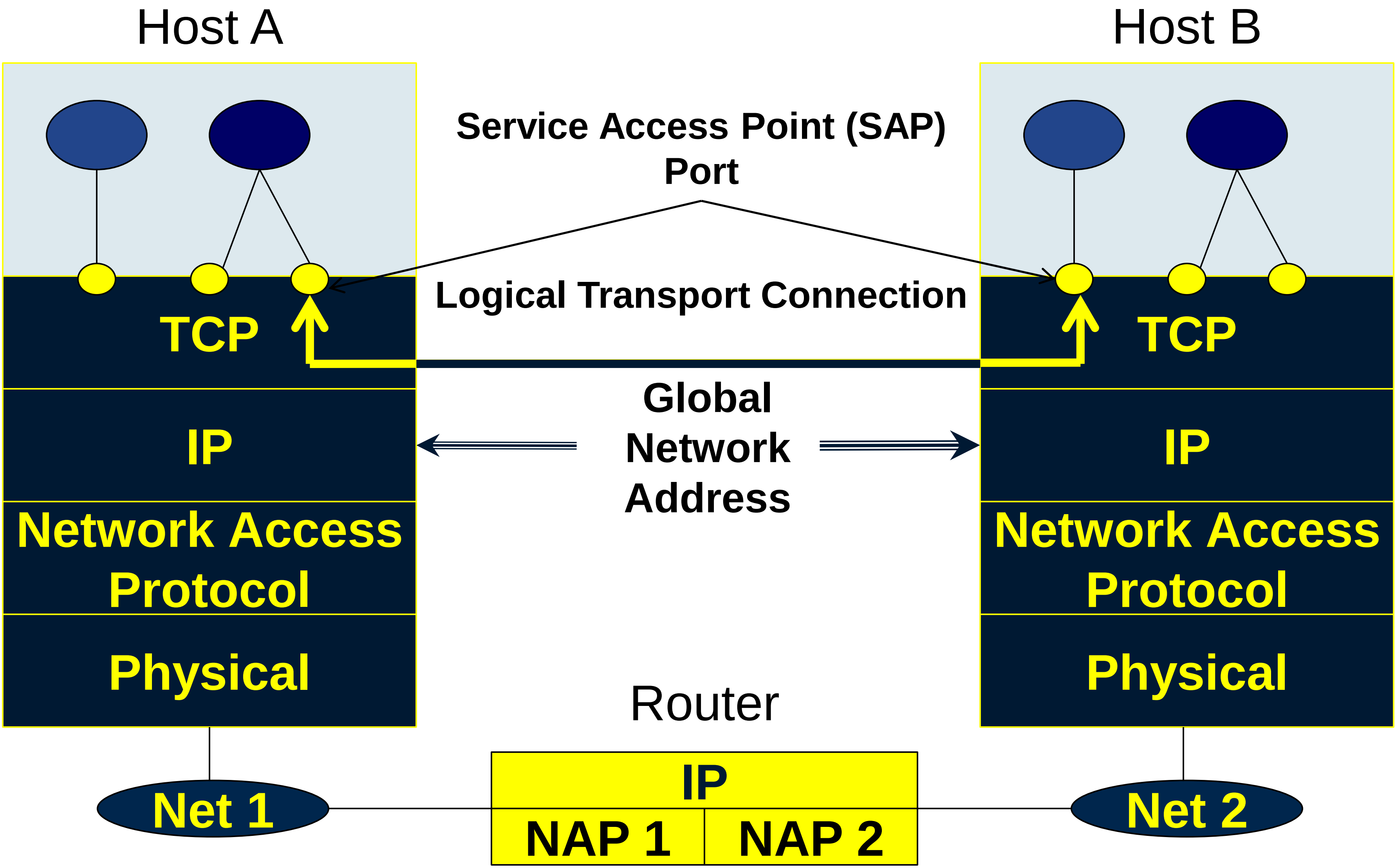
Error Control

- Error control guards against loss of damage of data and control information
  - Most techniques involve error detection and retransmission
    - Frame Check Sequence (FCS) is used to detect errors

Addressing

- Addressing is a difficult concept that covers a number of issues
  - Addressing level
  - Addressing scope
  - Connection identifiers
  - Addressing Mode

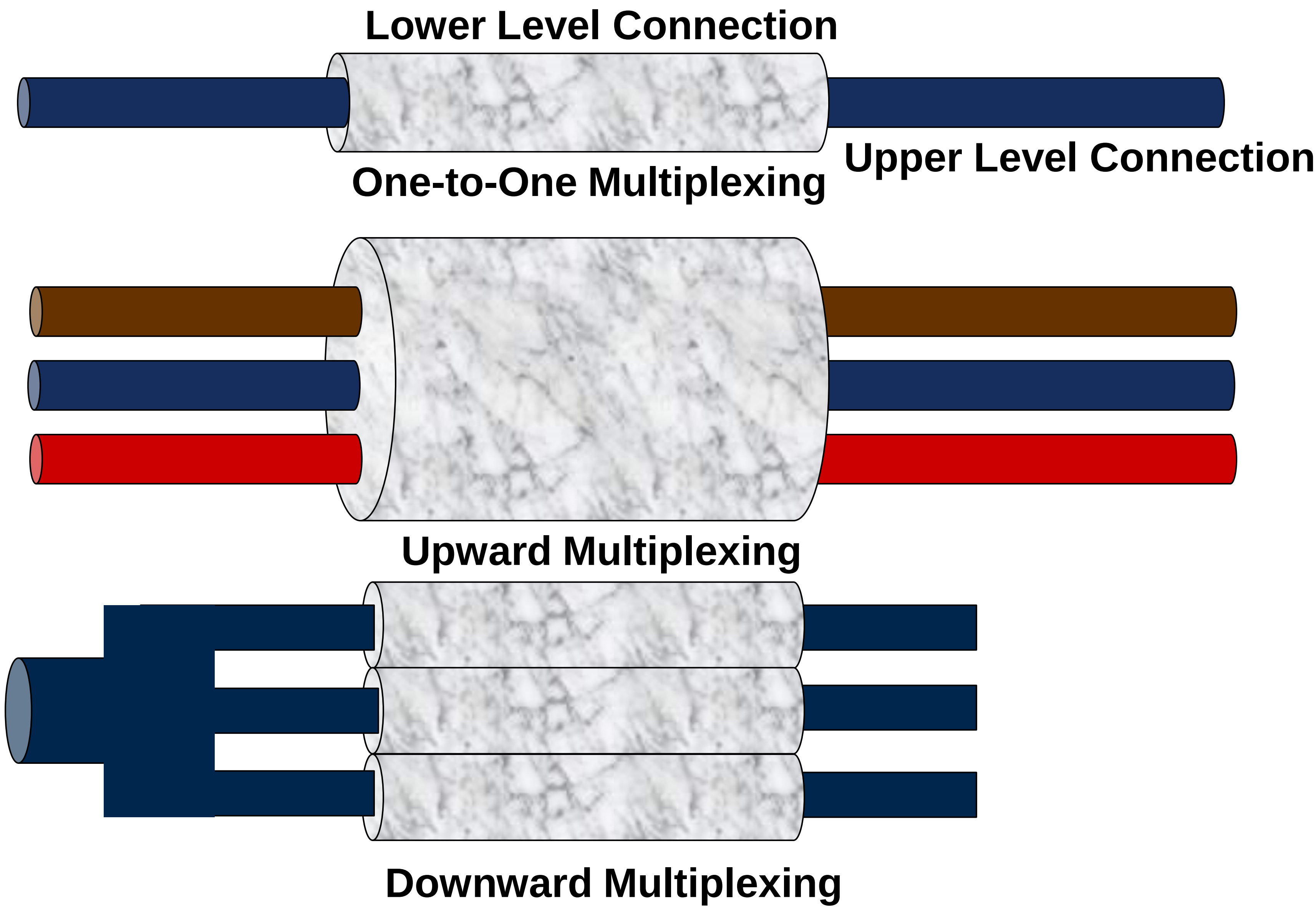
Addressing Concept



Multiplexing

- Multiplexing refers to the fundamental concept of sharing a logical or physical system resource among multiple higher level entities
  - Link, CPU, buffer, are typical shared resources

Multiplexing and Protocol Connections



Network Design

- Guide the organization and assignment of functions within the system
  - Impose a structure on the design space rather than solve a particular design problem
  - The structure provides a basis for analysis of tradeoffs and a rational for design choices

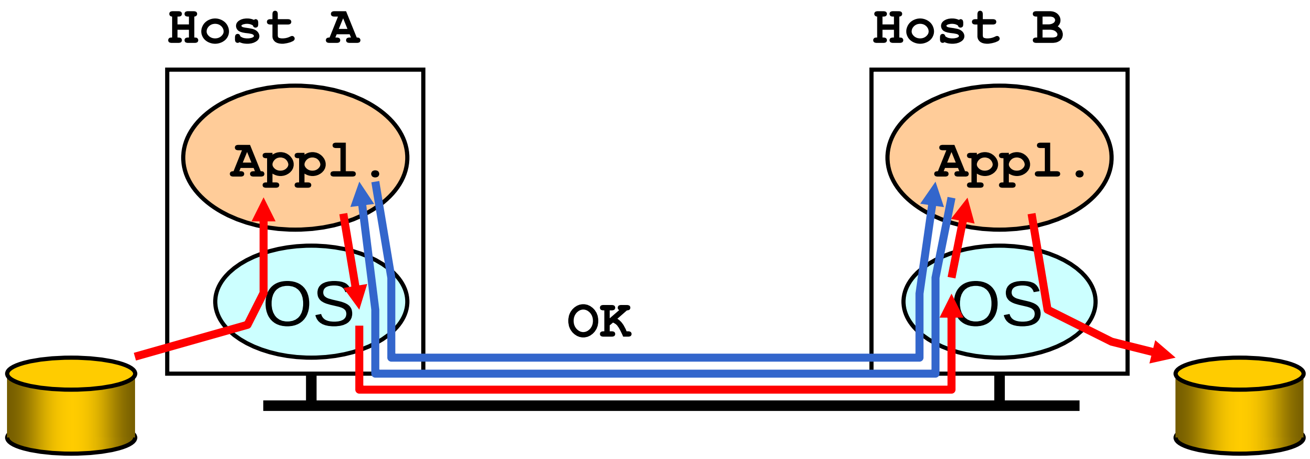
Network Design

End-to-End Argument

End-to-End Argument

- In a layered architecture, how do you divide functionality across the layers?
- In essence, the argument states that "**functions placed at low levels of a system may be redundant or of little value when the cost of providing them at the low level is factored in**"
  - A function is provided by a (sub)system only if it can be completely and correctly implemented within it
- What about performance?
  - "… sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement"

Reliable File Transfer – A Case Study



- Solution 1: make each step reliable, and then concatenate them
- Solution 2: end-to-end check and retry

Discussion

- Solution 1 may not be satisfactory
  - What happens if the sender or/and the receiver misbehave?
  - The receiver has to perform error checking anyway!
  - Should full functionality be entirely implemented at application layer?
    - In this case, lower layer do their “best” to forward packets reliably.

Discussion

- Is there any need to implement reliability at lower layers?
  - Yes, but only to improve performance
- Example: noisy network environment
  - Consider the case of a high error rate communication network
  - Reliable communication service at data link layer enhances performance
    - Errors can be detected and corrected at the frame level

End-to-End Arguments – Summary

- If a functionality can be totally supported at the application layer, don't implement it at a lower layer
  - The application knows best what it needs
- Add functionality in lower layers iff:
  - (**1**) – it is used and improves performances of a large number of applications, and
  - (**2**) – it does not impact negatively the performance of other applications
- End-to-End argument has been instrumental in the design of the Internet

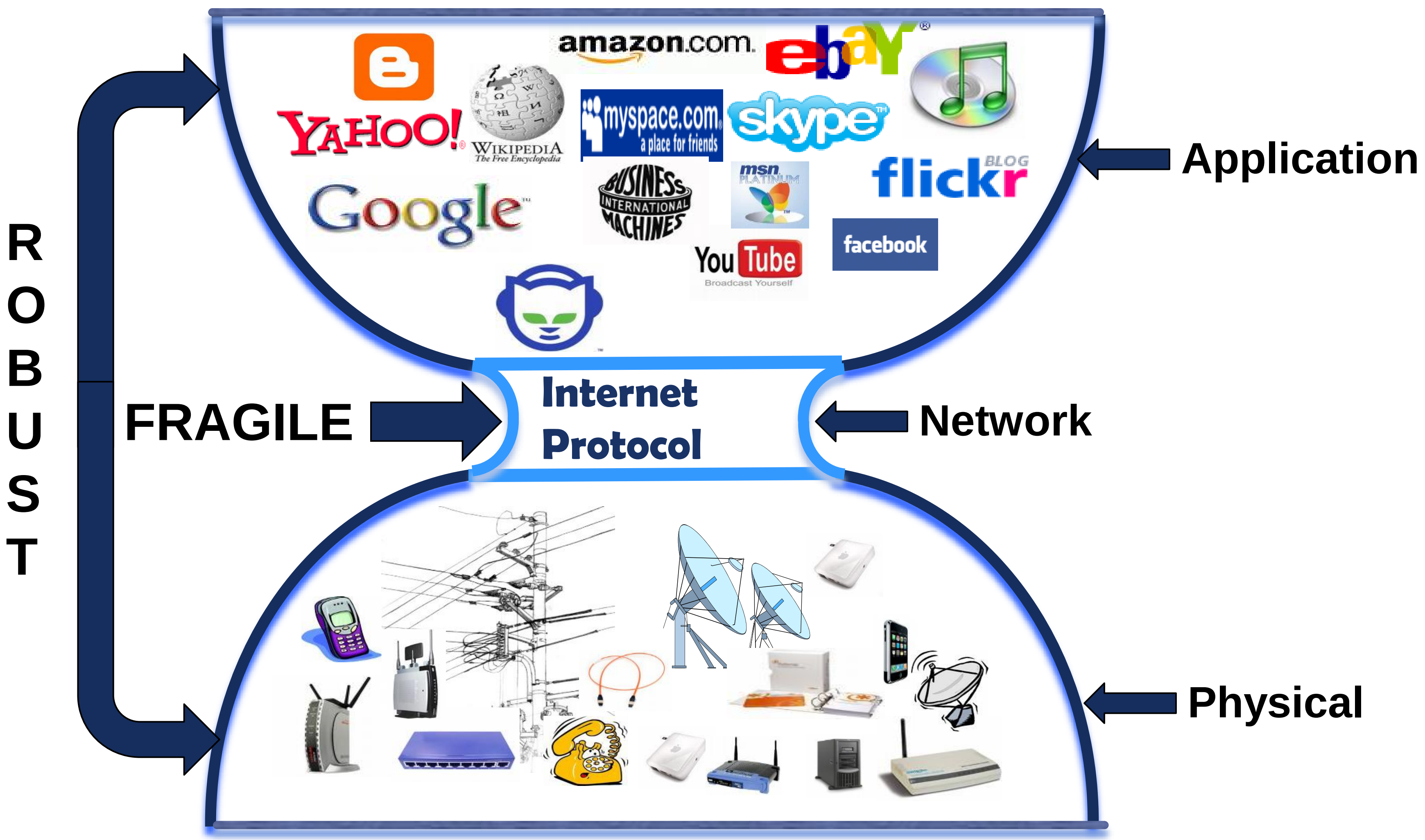
Internet Design Principles

- To achieve robustness, the emphasis on **simplicity** was a guiding principle of the Internet design.
  - The result is a “**light**” **network core** that provides **minimal packet-level forwarding service**
- This minimalistic approach to network design was later justified and generalized into the so called end-to-end argument.

End-to-End Argument and Internet Design

- The IP network layer provides a simple, best effort, datagram service
  - May not be reliable
- TCP supports reliable data delivery
  - Performance enhancement for a variety of applications: Telnet, FTP, HTTP, etc
  - Decision does not impact other applications, if UDP can be used
- Everything else is implemented at application level

Internet Architecture



Multi-Service Networks

QoS Requirements

Mutli-Service Networks

- Technological advances in computing and communications networks and the recognition the vast commercial potential of advanced information infrastructure created opportunities for new forms of data communication media
  - Multimedia Content
    - Text, Voice and Video

Enabling Technology

- Advances in computing and communication technologies are paving the way to multimedia processing and communication
  - Video compression, optical fiber networks, digital recording and digital high definition television
- Efficient processing and communication of multimedia data may require coordination and harmonization between different components of the system

Typical Applications

- Entertainment
  - Broadcasting, Video On Demand, Near real-time Video On Demand (VOD), Interactive Video On Demand (IVOD), Games,
  - Distributed access and exchange of different types of media
  - Real-Time Audio/Video capture and playback
- Information Browsing
  - Resource discovery and access, information search
  - Off-line browsing, playback, and editing of stored multimedia data
- Teleconferencing, Distance Learning, e-Commerce
  - Access to digital libraries and video archival, tutoring
  - Retail goods, reservations, real estate, job interviews

Applications Main Characteristics

- Video Conferencing
  - Real-time capture and processing
  - Latency is critical
  - Unpredictable future
- Video Playback
  - Off-line creation and processing
  - Latency less critical (read ahead)
  - Predictable future
- Video Editing
  - Off-line capture
  - Real-time processing
  - Predictable future ?

Bandwidth Requirements

- In the past, estimates of the bandwidths required for multimedia applications were daunting

Media Description	Uncompressed Speed Range
Studio Quality NTSC Video	120 Mbps
Broadcast Quality NTSC Video	216 Mbps
Broadcast Quality High Definition Video	1500 Mbps
Visualization Imagery Full motion (60 fps)	1500–2500 Mbps
Monochrome binary still image (600x600 spots/in) (135 pp/min)	120 Mbps
Full Color Still image (400x400 pixels/in) (60 pp/min)	500 Mbps

[Lyles and Swinehart]

Bandwidth Requirements Compression to the Rescue

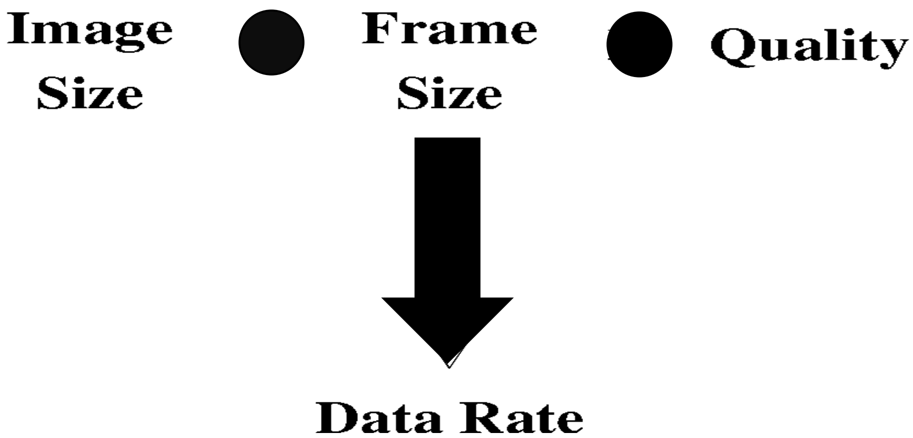
- High compression rates greatly reduce the application bandwidth requirements
  - True lossless compression is limited by the image
    - Generally, lossless compression results in a compression factor of no more than two to one
  - Much higher compression ratios can be achieved if information is lost
    - Visually lossless compression can be achieved by taking advantage of the characteristics of the human visual system

Compression Standards

- Several schemes have been proposed
  - Joint Photographic Experts Group (JPEG)
  - CCITT Recommendation H.261 (p x 64)
  - Moving Picture Experts Group (MPEG)

Compression Schemes

- Fundamental tradeoff



- Real-time compression is more expensive than off-line compression

Bandwidth Requirements

Media Description	Uncompressed Speed Range	Compressed Speed Range	Compression Method
Studio Quality NTSC Video	120 Mbps	3–6 Mbps	MPEG
Broadcast Quality NTSC Video	216 Mbps	10–30 Mbps	MPEG, JPEG
Broadcast Quality High Definition Video	1500 Mbps	20–30 Mbps	Proprietary
Visualization Imagery Full motion (60 fps)	1500–2500 Mbps	50–200 Mbps	Quality requirement not settled yet
Monochrome binary still image (600x600 spots/in) (135 pp/min)	120 Mbps	5–50 Mbps	CITT Std. Proprietary (Lossless)
Full Color Still image (400x400 pixels/in) (60 pp/min)	500 Mbps	45 Mbps	JPEG

[Lyles and Swinehart]

Bandwidth Requirements

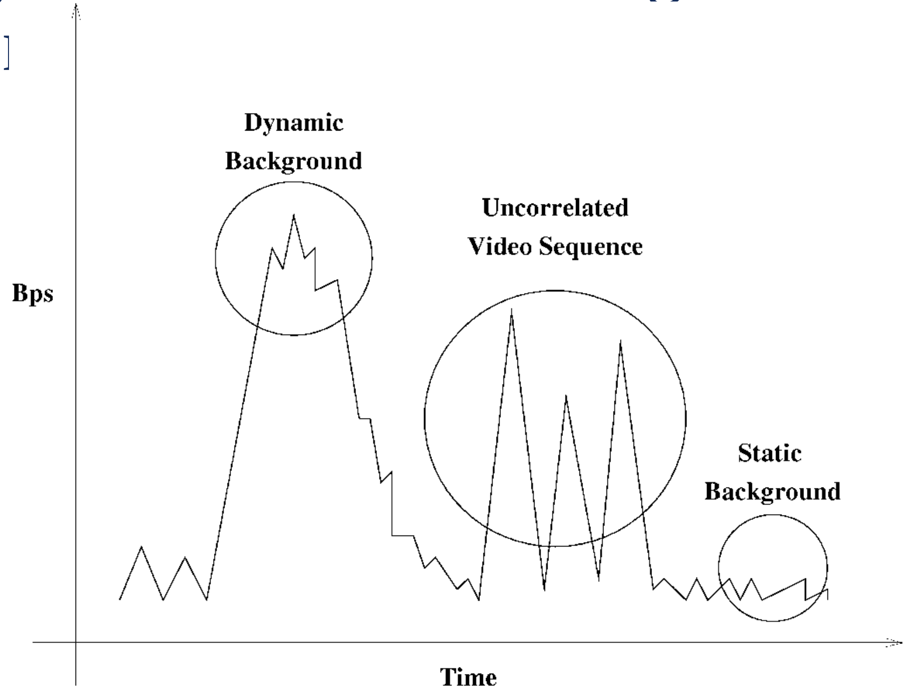
- If studio-quality video can be provided to a desktop at 10 to 15 Mbps, do we need even 150 Mbps links to that desktop?
  - The answer is "Yes"
  - Individuals may be using several cameras and monitors and multiple applications simultaneously
  - Group settings may require multiple channels

Communication Systems Requirements

- Multimedia brings about several requirements for networking systems
  - To a large extent, these requirements are application dependent
  - Reflect a set of bounds and guarantees on specific parameters
    - bandwidth, delay, jitter, …

System Requirements

- Ability to handle a wide range of traffic rate, efficie



System Requirements

- End-to-end jitter (delay variation) need to be bounded
  - Large jitter values cause degradation of interactive applications
  - Large jitter values result in large buffer requirements and high end-to-end delay
- All guarantees necessary for achieving data transfer within the time bound must be met

System Requirements

- Support for synchronization among multiple streams
  - Inter-stream synchronization
  - Intra-stream synchronization

Multi Service Networks

Reliability, and Latency-Bandwidth Tradeoffs

Link Reliability

- Bit Error Rate (BER)

  Prob[bit not in error] = (1-BER)

- Over one link :

  Prob[packet not in error] $= (1\text{-}BER)^L$

- Over *N* links :

  Prob[packet not in error] $= ((1\text{-}BER)^L)^N$

- *L*, packet length and *N* number of hops

Slow Packet Networks

- BER=10^{-3}, worst case while meeting Telco specifications

Over Errored Voice Grade Lines

User Data (Bits)	***80***	***800***	***8000***
Overhead (Bits)	***50***	***50***	***50***
Packet Length	***130***	***850***	***8050***

Over 1 Link	***93.7%***	***65.4%***	***1.8%***
Over 3 Links	***82.3%***	***30.0%***	***Nil***
Over 5 Links	***72.2%***	***12.0%***	***Nil***

Fast Packet Networks

- BER=10^{-9}, reported for average Fiber Optic Links

Over Poor Optical Fiber Links

User Data (Bits)	***384***	***8,000***	***80,000***
Overhead (Bits)	***40***	***256***	***2,000***
Packet Length	***424***	***8,256***	***82,000***
Over 1 Link	***99.99%***	***99.99%***	***99.91%***
Over 5 Links	***99.99%***	***99.95%***	***99.59%***
Over 15 Links	***99.99%***	***99.87%***	***98.77%***

Fundamental Observation

- Without error detection and correction, “Slow Packet Networks” would be useless over multiple hops or for long packets
  - Error correction on a Hop-by-Hop basis minimizes delays and improves overall performance
- Fast Packet Networks are designed for high quality digital transmission
  - Performance improvement can be achieved by confining error detection and correction to end-to-end systems, if necessary

Latency vs. Bandwidth

- Parameters characterizing a communication network
  - C : Capacity of the network (Mbps)
  - P : Packet length (bits)
  - L : Length of the network (miles)
- For a given network, these parameters can be combined into a single critical parameter, denoted as ***a***

  $\boldsymbol{a} = \mathbf{5}\ L \cdot C / P$, *where* 5 represents the approximate number of µsec it takes light to move one mile

  - ***a*** represent the ratio of the latency of the channel – the time it takes energy to move from one end of the link to the other – to the time it takes to pump one packet into the link
    - It measures how many packets can be injected in the link before the first bit appears at the other end

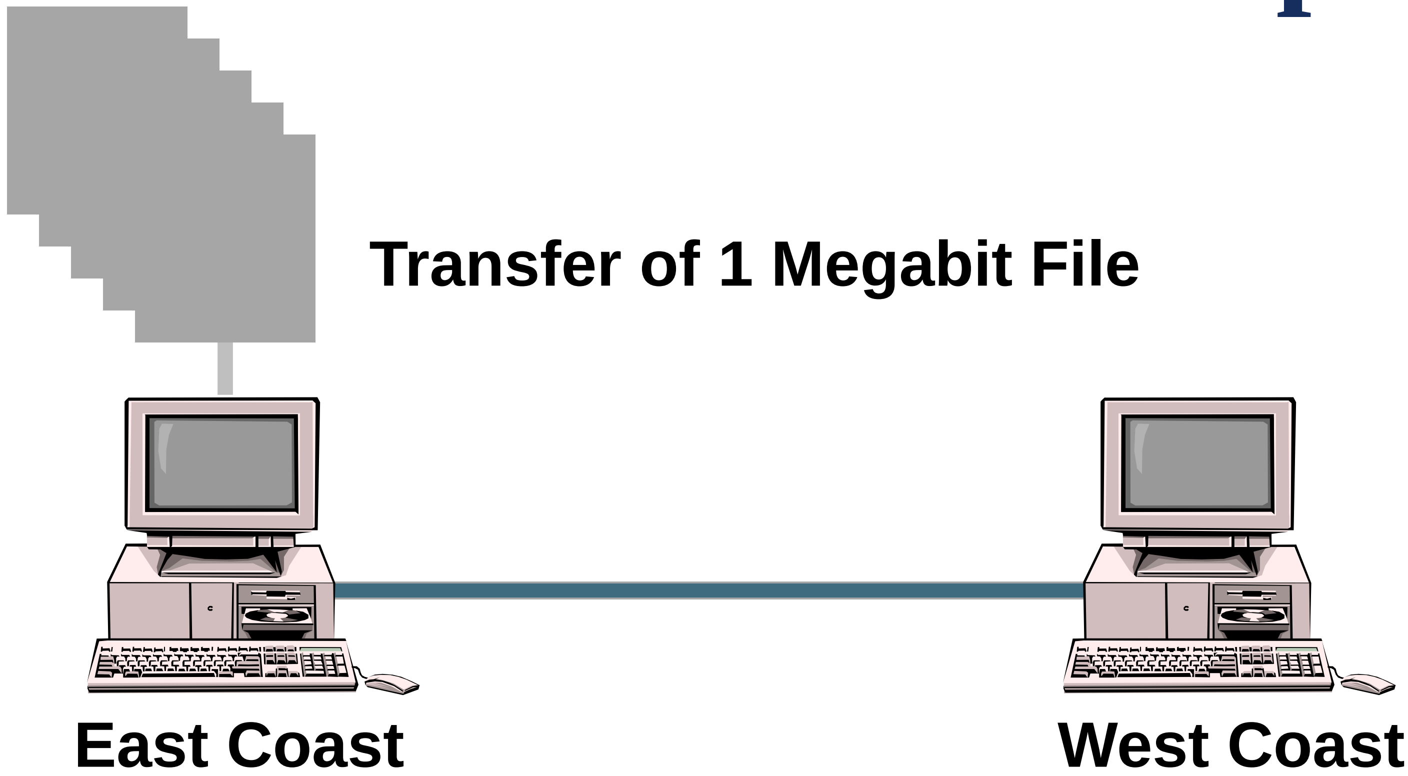
Effect of Parameter *a*

	Bandwidth (Mbps)	Pkt Length (bits)	Prop Delay (μ - sec)	Ratio *a*
Local Area Networks	10.00	1000	5	0.05
Wide Area Networks	0.05	1000	20,000	1.0
Satellite	0.05	1000	250,000	12.50
Fiber Link	1,000.0	1000	15,000	15,000.00

Effect of Parameter *a*

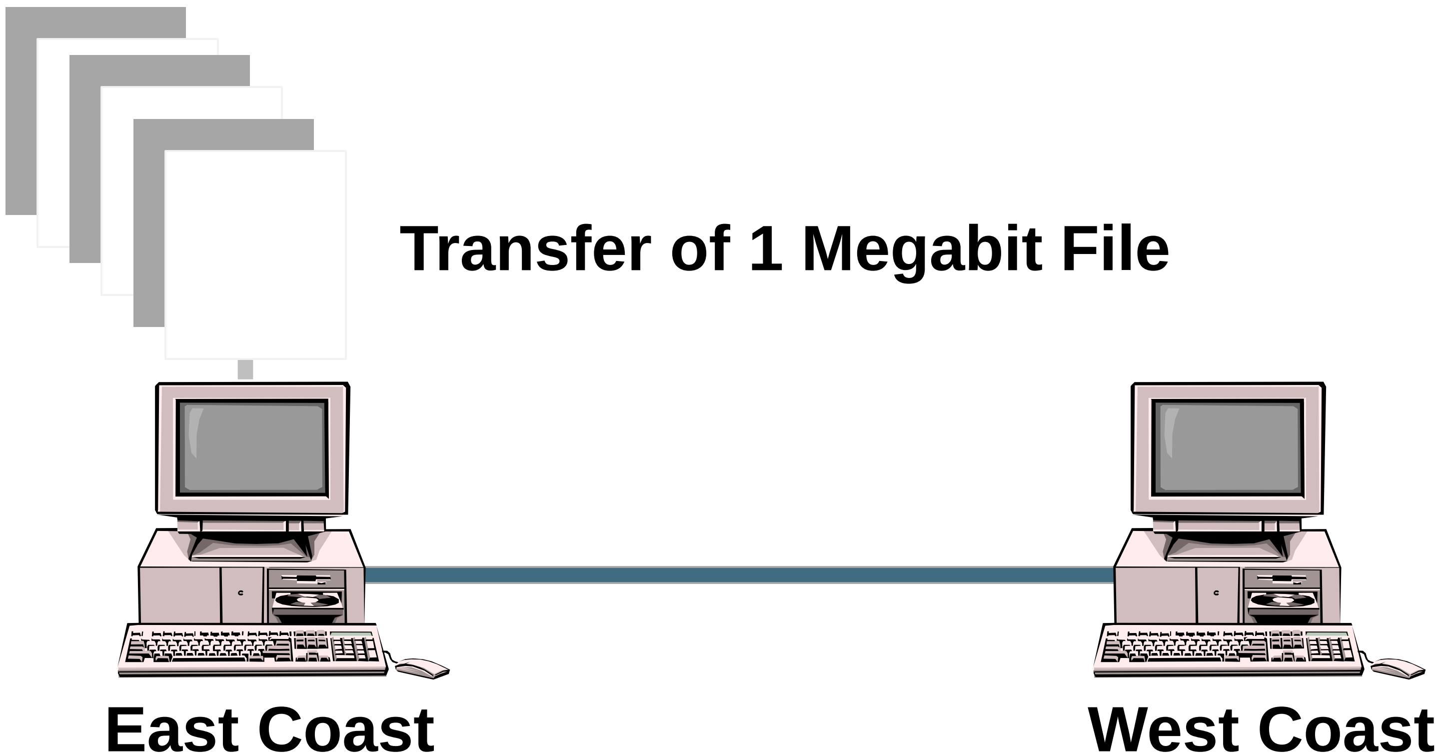
- Given that ***a*** grows dramatically with high speed networks
  - Are high speed networks fundamentally different from low speed networks?
  - What is the impact of ***a*** over network design parameters

Effect of Parameter *a*
File Transfer Over 64 Kbps



- First bit arrives after 1000 bits have been pumped into channel
- Terminal buffers1000 times as much data as that in the channel

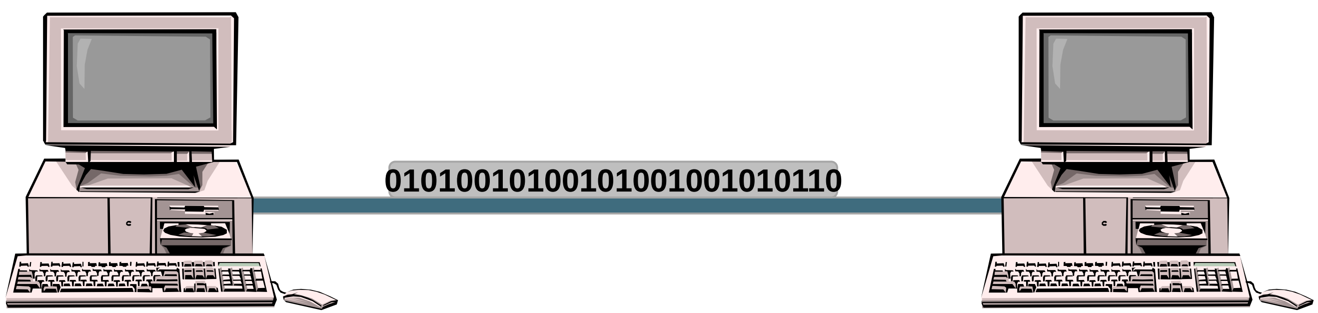
Effect of Parameter *a*
File Transfer Over T1 (1.544 Mbps)



- Terminal buffers only 40 times as much data as that in the channel

Effect of Parameter *a*
File Transfer Over 1.2 Gbps

Transfer of 1 Megabit File



- ❑ The entire file is injected into the channel before the first bit reaches destination

Effect of Parameter *a*
File Transfer Over 1.2 Gbps

- A fundamental change comes about with the introduction of high speed networks
  - Networks moved from “Capacity Limited” to “Latency Limited”
- Network fundamental schemes must be revisited

What Breaks When We Go Faster?

- It is tempting to think of “high-speed” as meaning that latency improves at the same rate as bandwidth
  - We cannot forget about the limitation imposed by the speed of light
    - Round Trip Time for 1 Gpbs link is exactly the same as the Round Trip Time of 1 Mbps link

What Breaks When We Go Faster?

- The bandwidth available on the network starts to rival with the bandwidth inside the computers that are connected to the network
  - End host design issues become important
    - Both host's architecture and the software running on the host must be designed carefully if we were to deliver the Gbps bandwidth end to end

What Breaks When We Go Faster?

- Availability of high bandwidth create new opportunities for new classes of application
  - These applications require performance guarantees, not just raw performance
  - These applications are likely to stress the network's underlying service model
    - Current service model is developed for applications that run at low speed, mostly

Architectural Design Issues For Next Generation Networks

The World is Changing!

- We built a complex network SIMPLY
  - The far-reaching original design choice of the Internet architecture traded security and rich behavior for simplicity
- Our network infrastructure is evolving into a "**Socio-Technical Ecosystem of Things**"
  - Deal with scale, ease of management, mobility, heterogeneity,
  - Needed is trustworthy, socially aware, economically viable Internet

Rethinking the Architecture

Need to meet a set of requirements that are contextual and future-looking

Future Internet Architectures

- New “architectural thinking” to overcome fundamental limitations of current frameworks and design principles
  - Is the concept of layering fundamental, and if so what is the optimal set of layers appropriate for complex socio-technical networks?
  - How should functionalities be assigned to different layers
  - Is it time for cross-layer optimization?

Future Internet Architectures

- New "architectural thinking" to exploit new and yet to be discovered physical substrates
  - Leverage optical and harness advances in wireless communication technology for high speed communication links
    - Develop design and architectural principles for seamless mobility support
  - Develop architectures for self-evolving, robust, manageable future networks
    - Embed "intelligence" for better control and management
    - Network virtualization, Software Defined Network and Open Flow protocols

Conclusion

- Introduction to Computer Networks
  - Classes of Networks
  - Switching Techniques
- Network Protocol Architectures
- Future Networks
  - Design requirements
  - Need for new architectural thinking