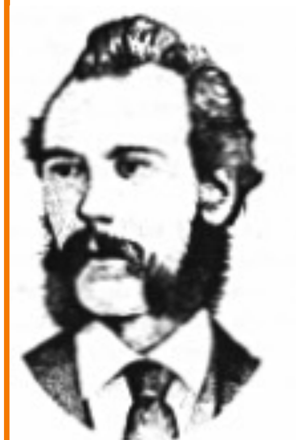
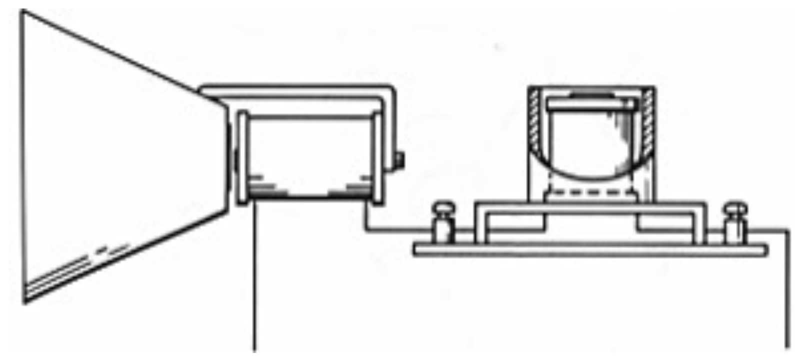
Outline for today's class

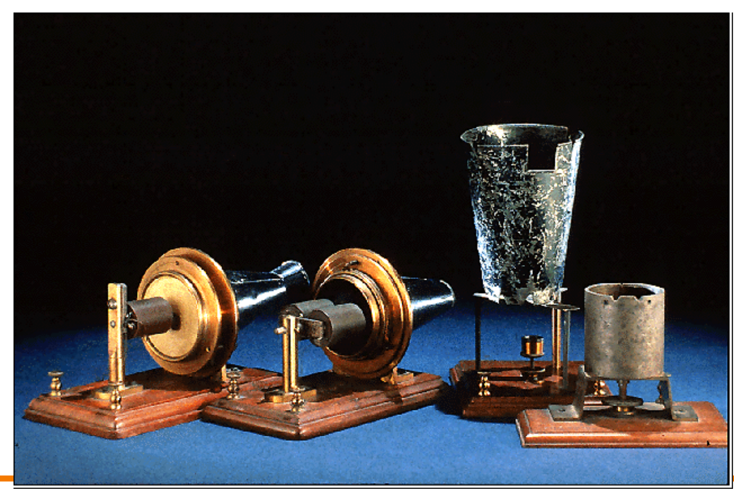
- The telephone network
- Taxonomy of networks
- Some basics of packet switching
- Statistical multiplexing
  - This is something you should know deep in your soul…

Telephones



- Alexander Graham Bell
  - 1876: Demonstrates the telephone at US Centenary Exhibition in Philadelphia





Telephone was an app, not a network!

- The big technological breakthrough was to turn voice into electrical signals and vice versa.
  - Great achievement
  - One of the nastiest patent battles in history

- The demonstration of this new device involved two phones connected by a single dedicated wire.

Todays Applications

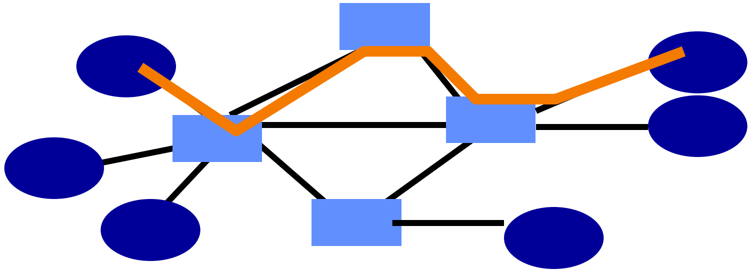
- Most people know about the Internet (a computer network) through applications
  - World Wide Web
  - Email
  - Online Social Network
  - Streaming Audio Video
  - File Sharing
  - Instant Messaging
  - …

What about the phone “network”?

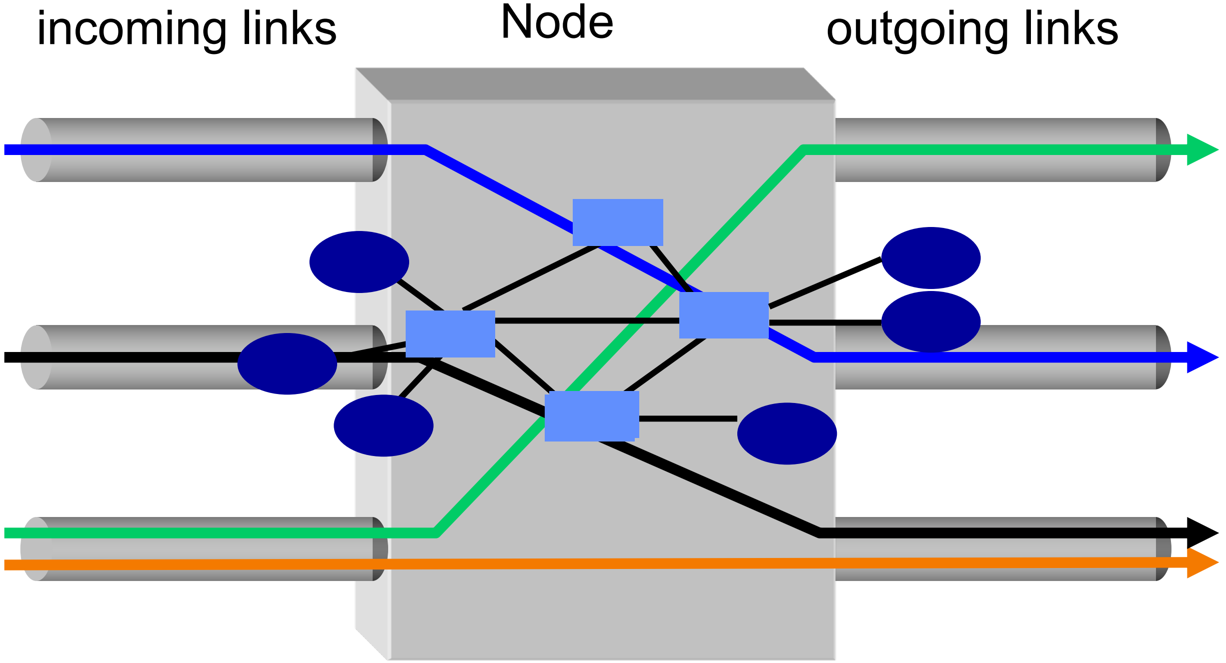
- You can’t have a dedicated wire between every two telephones
  - Doesn’t scale
  - Most wires will go unused….
- You need a “shared network” of wires
  - Much like the highway is shared by cars going to different destinations
- The telephone network grew into the first large-scale electronic network

Telephone network uses circuit switching

- Establish: source creates circuit to destination
  - Nodes along the path store connection info
  - And reserve resources for the connection
  - If circuit not available: “Busy signal”
- Transfer: source sends data over the circuit
  - No destination address in msg, since nodes know path
  - Continual stream of data
- Teardown: source tears down circuit when done

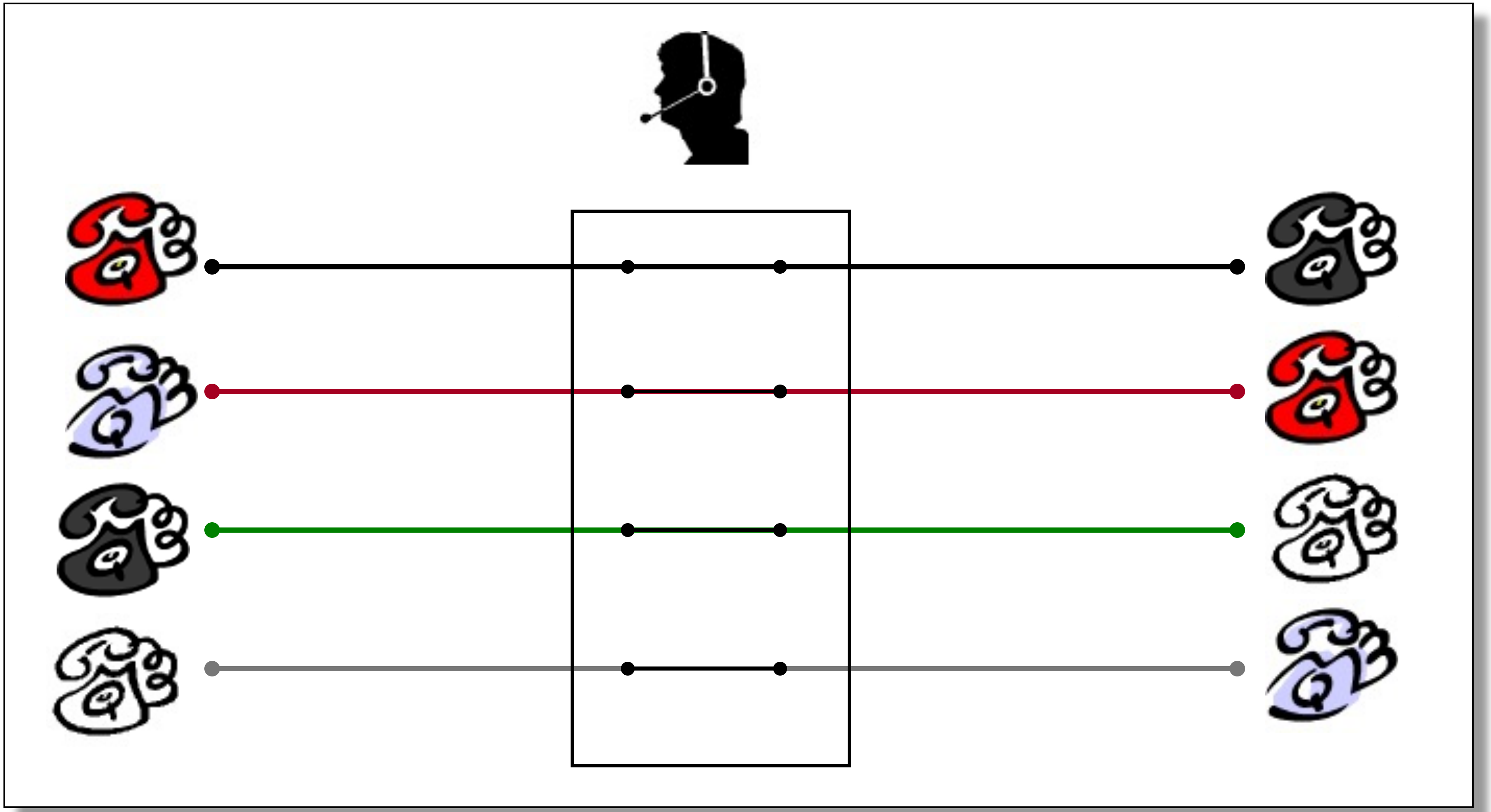


The switch in “circuit switching”



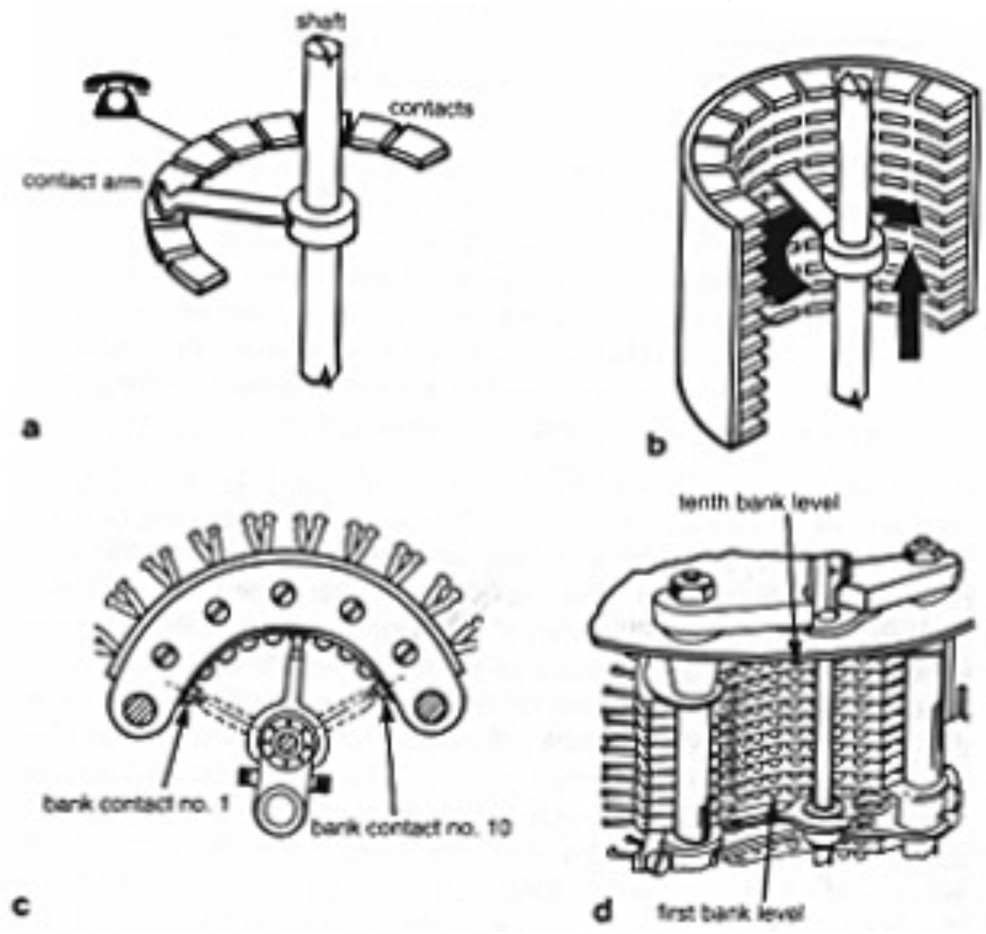
How does the node connect the incoming link to the outgoing link?

Circuit Switching With Human Operator

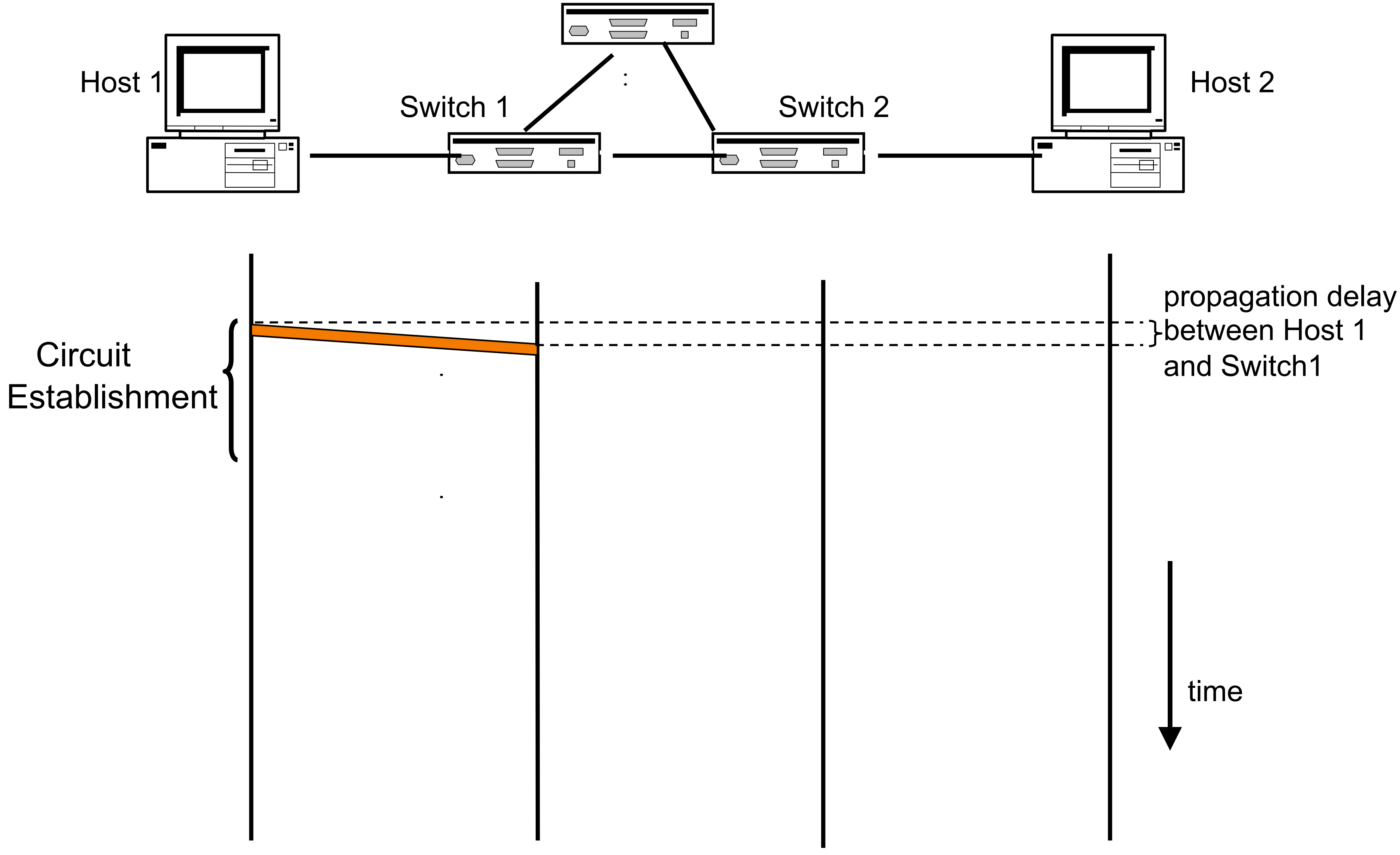


“Modern” switches

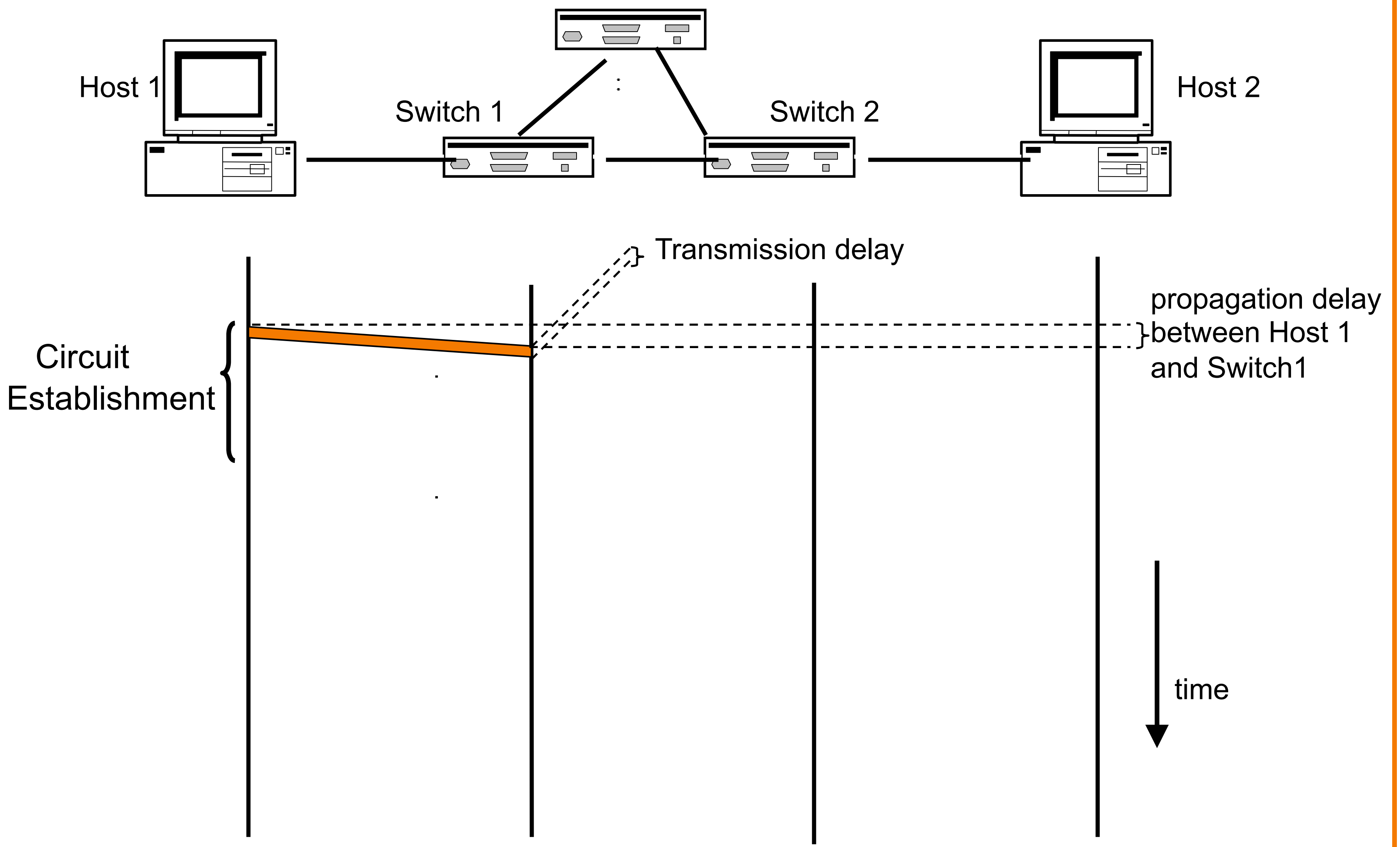
- Almon Brown Strowger (1839 - 1902)
  - 1889: Invents the “girl-less, cuss-less” telephone system -- the *mechanical switching system*



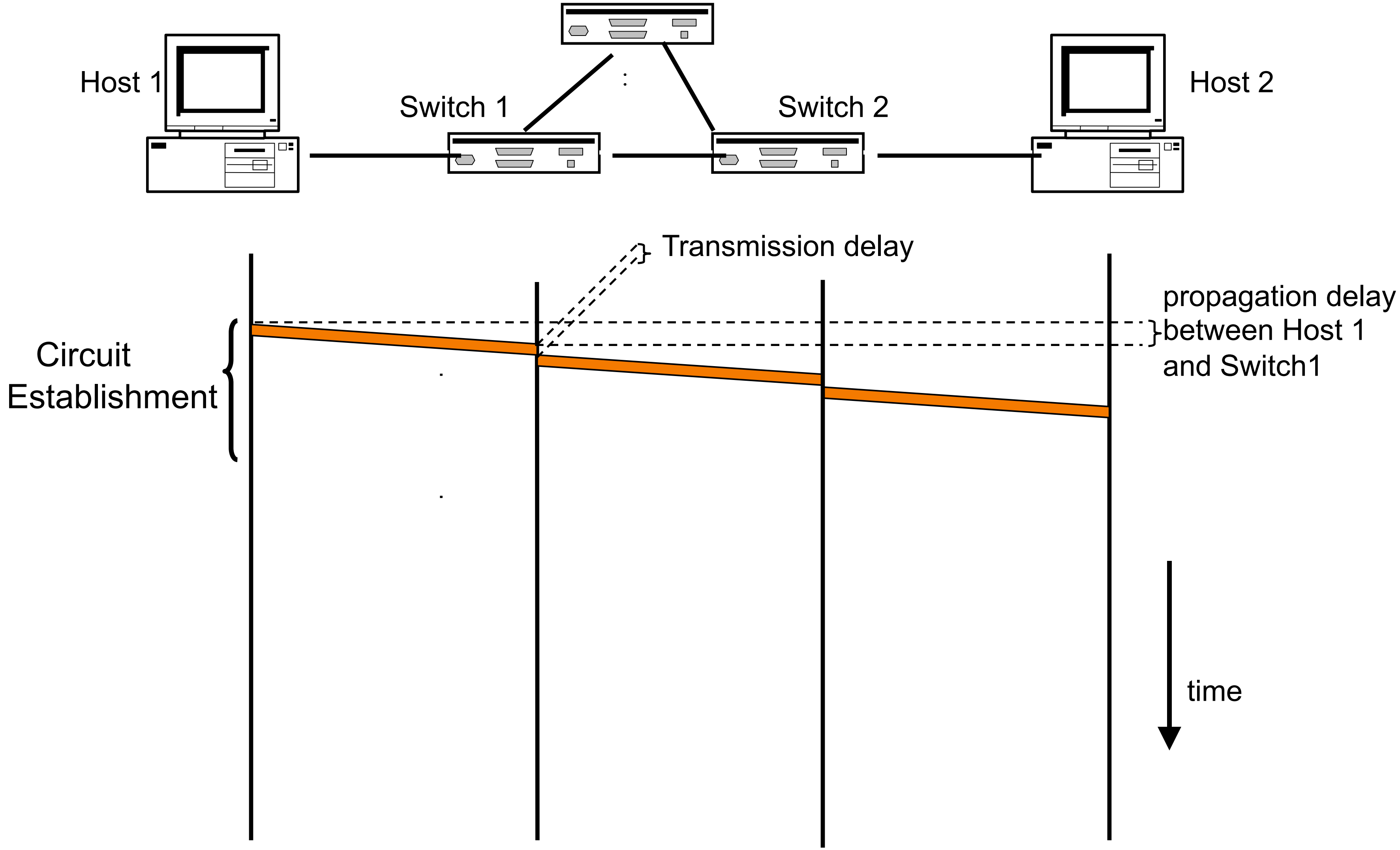
Timing in Circuit Switching



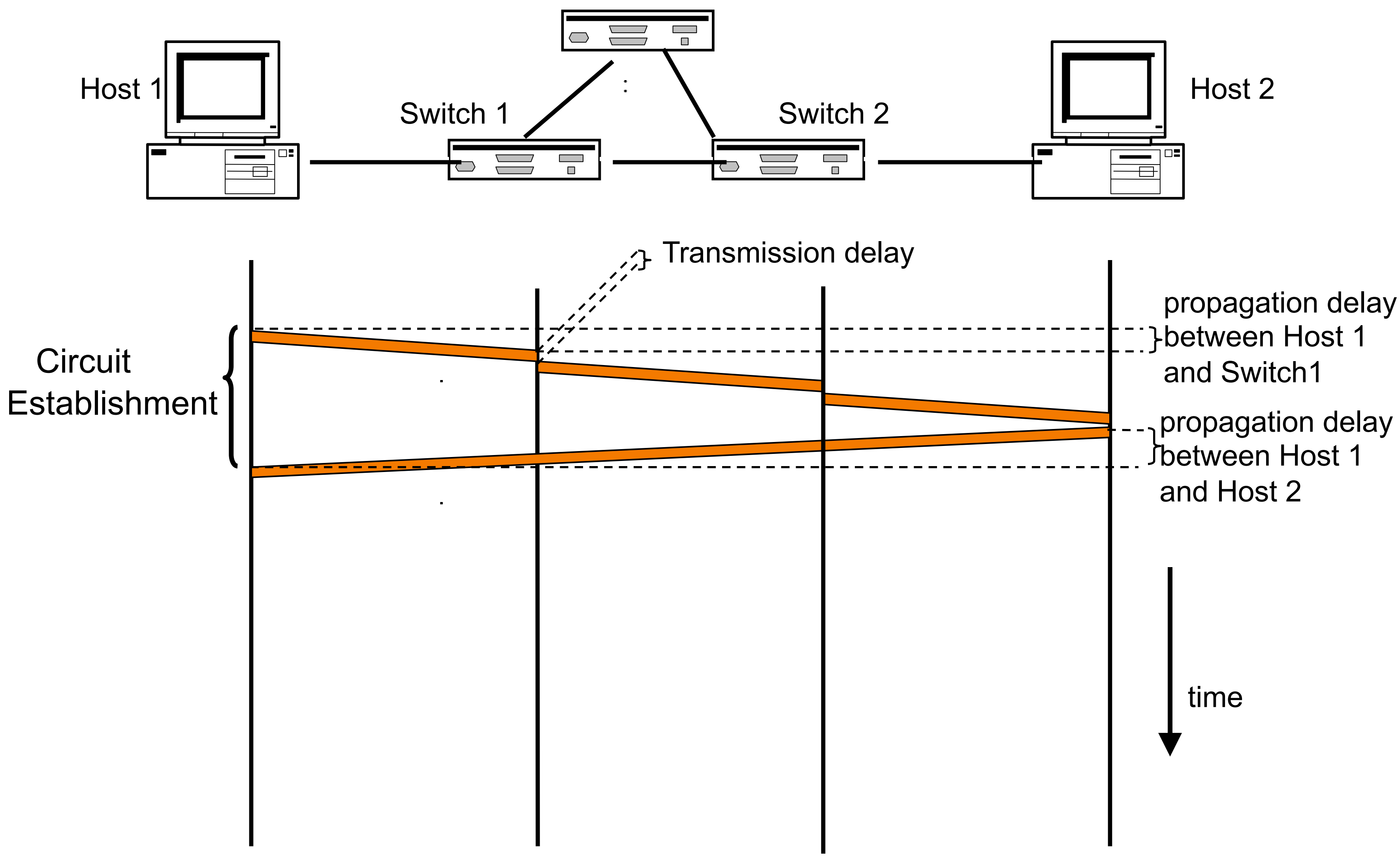
Timing in Circuit Switching



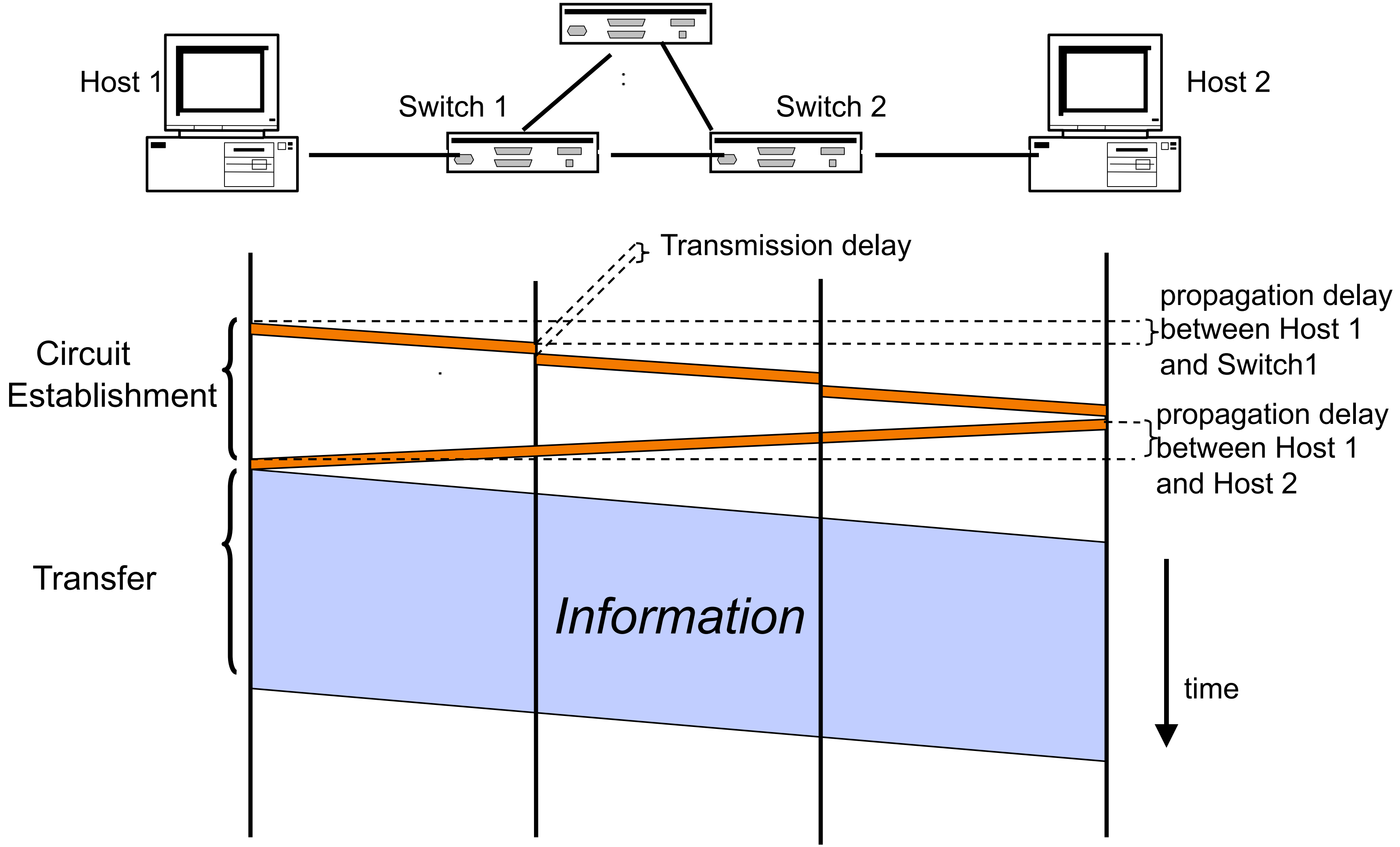
Timing in Circuit Switching



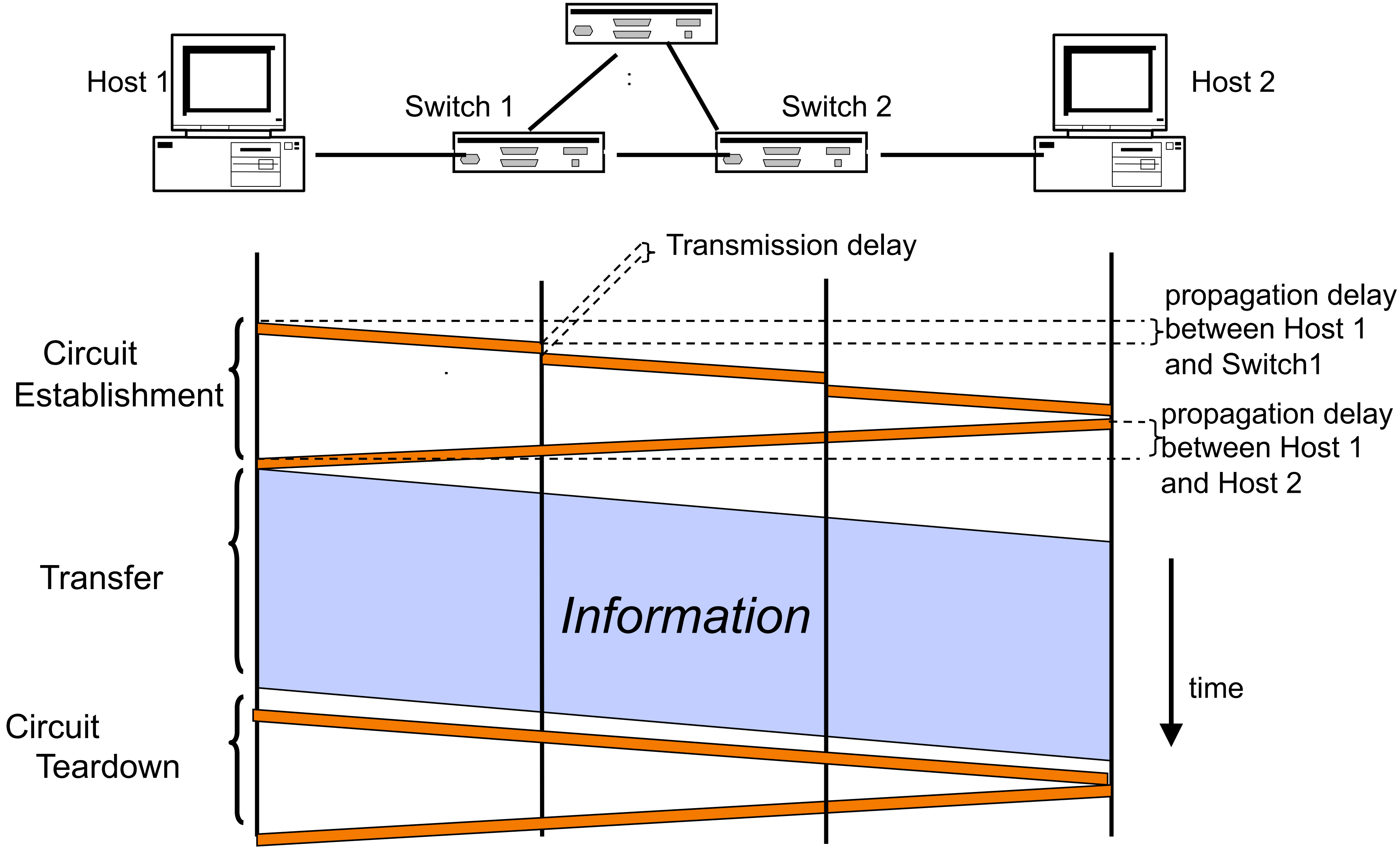
Timing in Circuit Switching



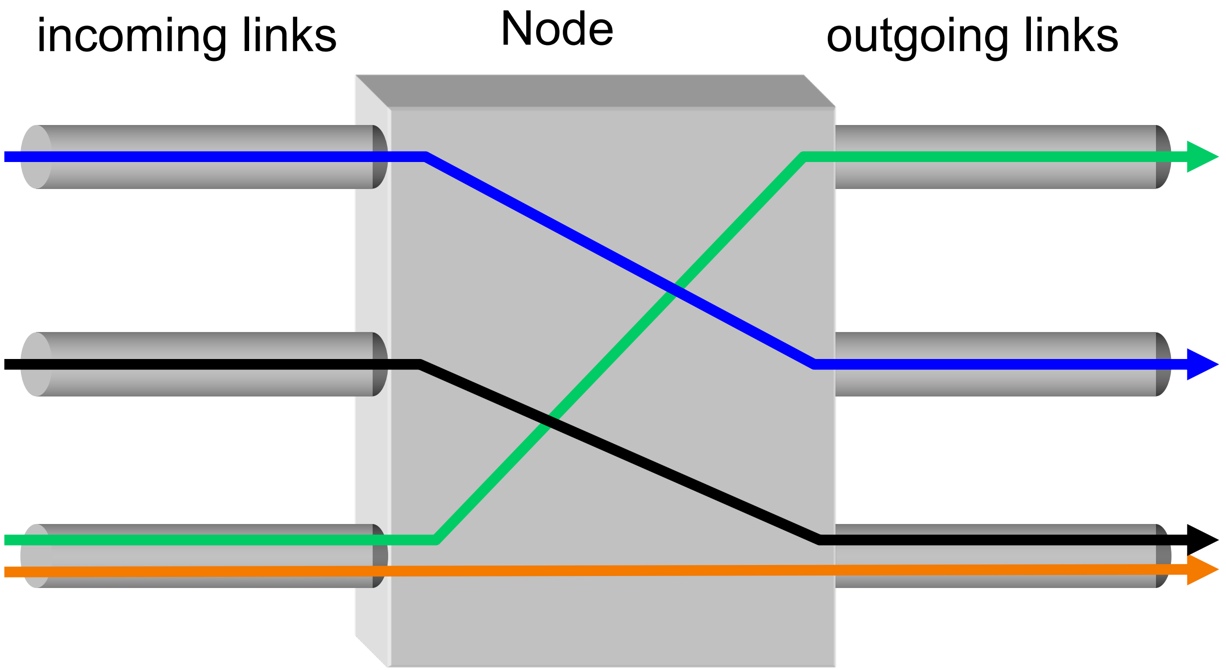
Timing in Circuit Switching



Timing in Circuit Switching



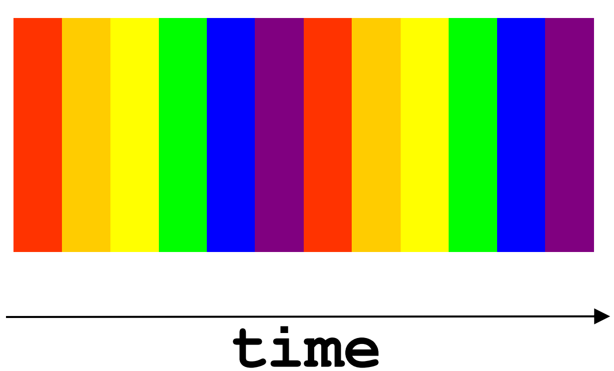
Sharing a link



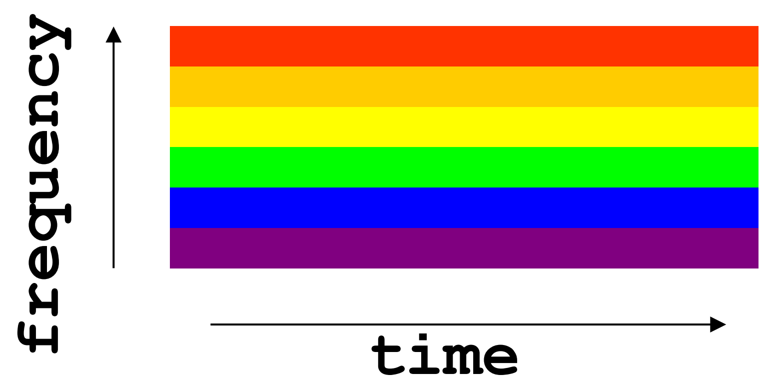
How do the black and orange circuits share the outgoing link?

Circuit Switching: *Multiplexing* a Link

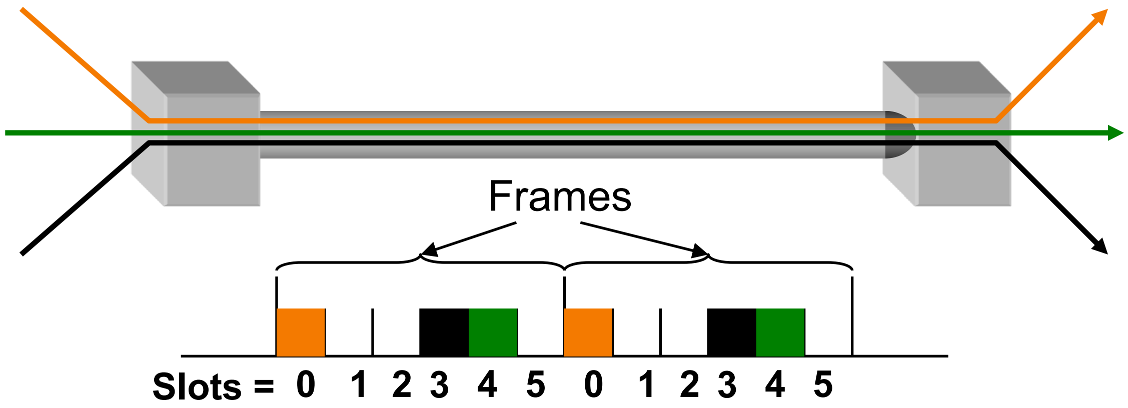
- Time-division
  - Each circuit allocated certain time slots
  - Time Division Multiple Access (TDMA)



- Frequency-division
  - Each circuit allocated certain frequencies
  - Frequency Division multiple access (FDMA)



Time-Division Multiplexing/Demultiplexing



- Time divided into frames; frames into slots
- Relative slot position inside a frame determines to which conversation data belongs
  - E.g., slot 0 belongs to orange conversation
- Requires synchronization between sender and receiver
- Need to dynamically bind a slot to a conversation
- If a conversation does not use its circuit **capacity is lost!**

Strengths of phone system

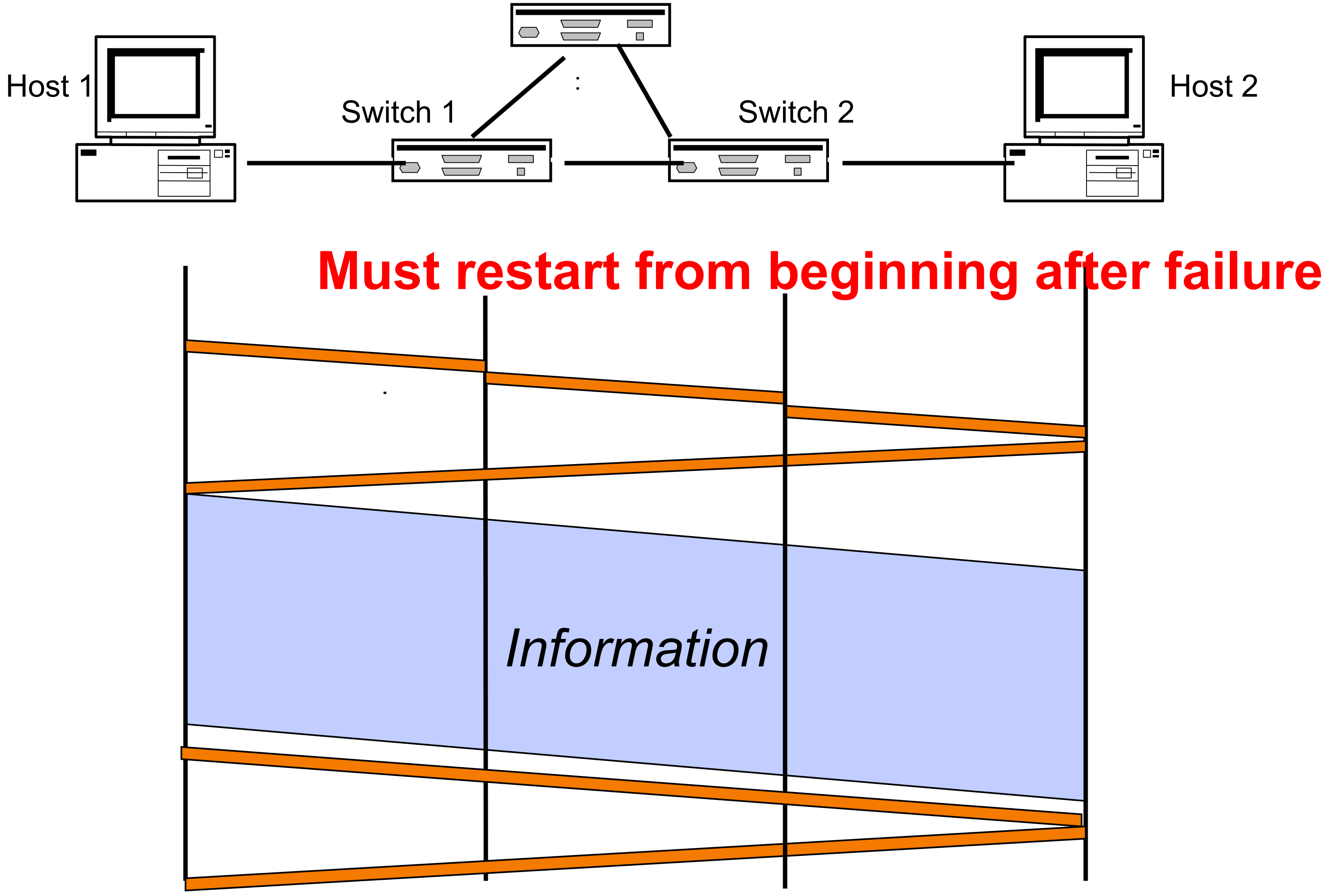
- Predictable performance
  - Known delays
  - No drops
- Easy to control
  - Centralized management of how calls are routed
- Easy to reason about
- Supports a crucial service

What about weaknesses?

Weakness #1: Not resilient to failure

- Any failure along the path prevents transmission
- Entire transmission has to be restarted
  - “All or nothing” delivery model

All-or-Nothing Delivery



Weakness #2: Wastes bandwidth

- Consider a network application with:
  - Peak bandwidth P
  - Average bandwidth A
- How much does the network have to reserve for the application to work?
  - The peak bandwidth
- What is the resulting level of utilization?
  - Ratio of A/P

Smooth vs Bursty Applications

- Some applications have relatively small P/A ratios
  - Voice might have a ratio of 3:1 or so
- Data applications tend to be rather bursty
  - Ratios of 100 or greater are common
- Circuit switching too inefficient for bursty apps
- Generally:
  - Don’t care about factors of two in performance
  - But when it gets to several orders of magnitude….

Statistical Multiplexing

- Will delve into this in more detail later
- But this is what drives the use of a shared network
- And it is how we could avoid wasting bandwidth

Weakness #3: Designed Tied to App

- Design revolves around the requirements of voice
- Not general feature of circuit switching
  - But definitely part of the telephone network design
  - Switches are where functionality was implemented

Weakness #4: Setup Time

- Every connection requires round-trip time to set up
  - Slows down short transfers
- In actuality, may not be a big issue
  - TCP requires round-trip time for handshake
  - No one seems to mind….
- This was a big issue in the ATM vs IP battle
  - But I think it is overemphasized as a key factor

How to overcome these weaknesses?

- There were two independent threads that led to a different networking paradigm….

What if we wanted a resilient network?

- How would we design it?
- This is the question **Paul Baran** asked….

Paul Baran

- Baran investigated survivable networks for USAF
  - Network should withstand almost any degree of destruction to individual components without loss of end-to-end communications.
- "On Distributed Communications" (1964)
  - Distributed control
  - Message blocks (packets)
  - Store-and-forward delivery

What about a less wasteful network?

- How would we design it?

- This is the question **Len Kleinrock** asked…..
  - Analyzed packet switching and statistical multiplexing

Returning to title of lecture

- If the Internet is the answer, then what was the question?
- There were two questions:
  - How can we build a more reliable network?
  - How can we build a more efficient network?
- Before considering nature of Internet, let's consider the broader design space for networks
  - Term "network" already implies we are sharing a communications infrastructure (i.e. not dedicated links)

Taxonomy of Networks

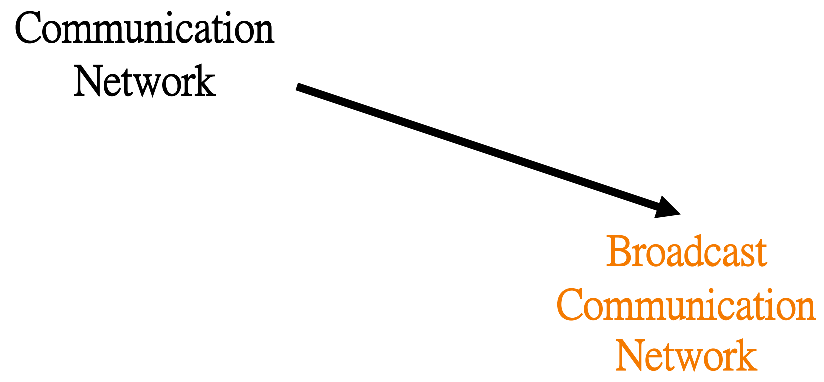
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the ***nodes*** exchange information:

Communication
Network

Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the ***nodes*** exchange information:

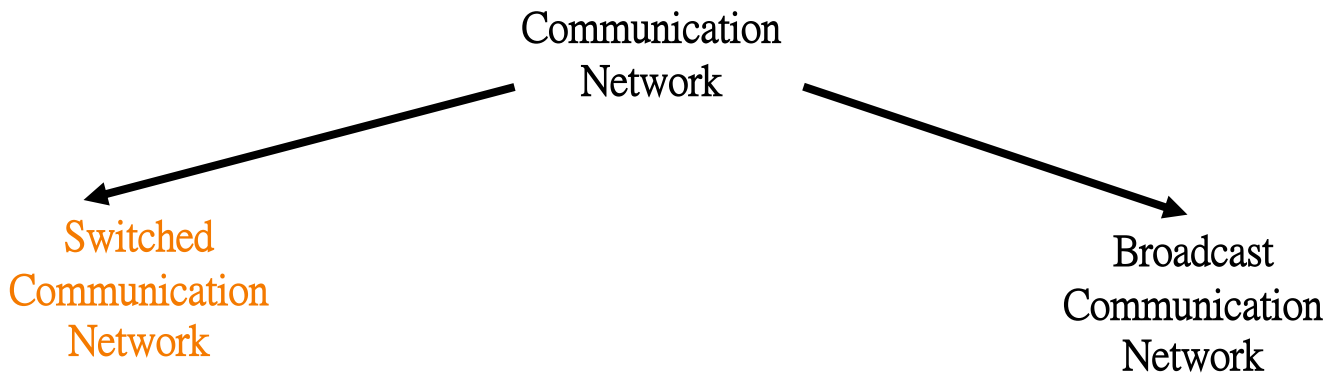


Broadcast Communication Networks

- Information transmitted by any node is received by every other node in the network
  - Usually only in LANs (*Local Area* Networks)
    - E.g., WiFi, Ethernet (classical, but not current)
    - E.g., lecture!
- What problems does this raise?
- Problem #1: limited range
- Problem #2: coordinating access to the shared communication medium
  - *Multiple Access Problem*
- Problem #3: privacy of communication

Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:

Communication Network

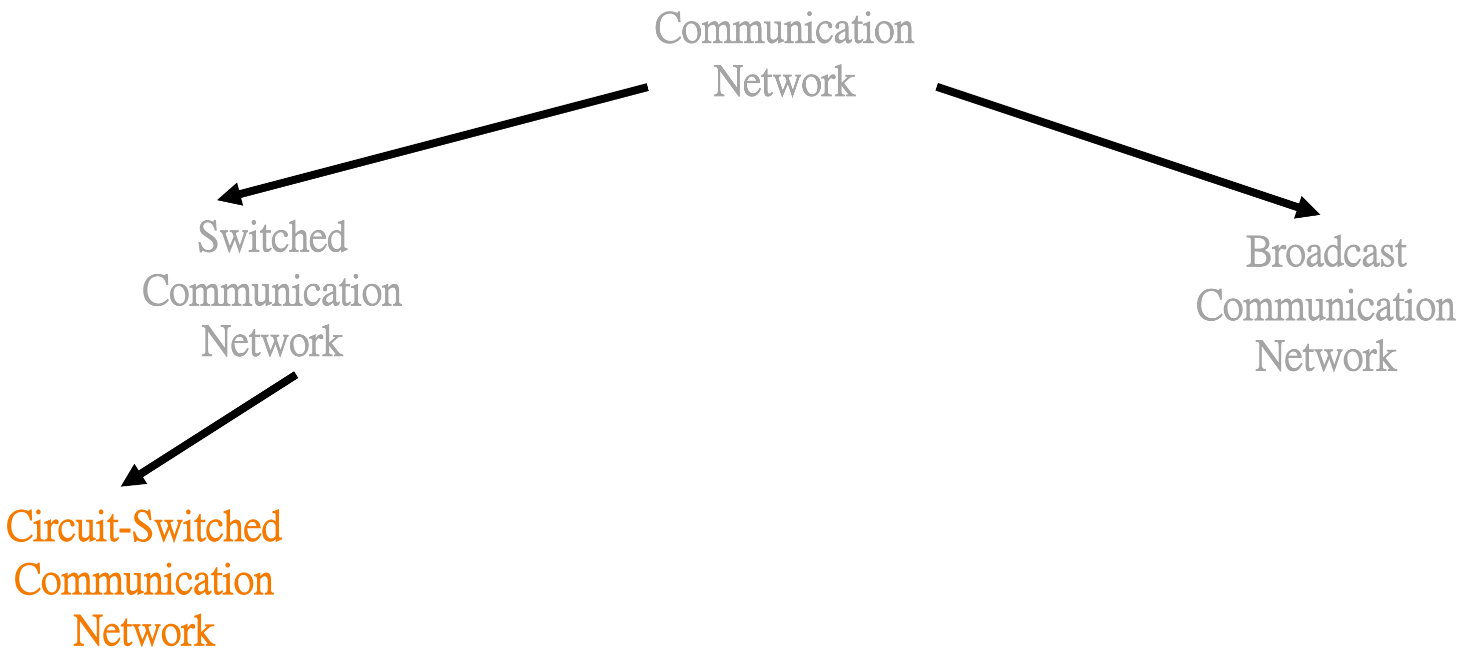
- Switched Communication Network
- Broadcast Communication Network

The term "switched" means that communication is directed to specific destinations

The question is how that "switching" is done

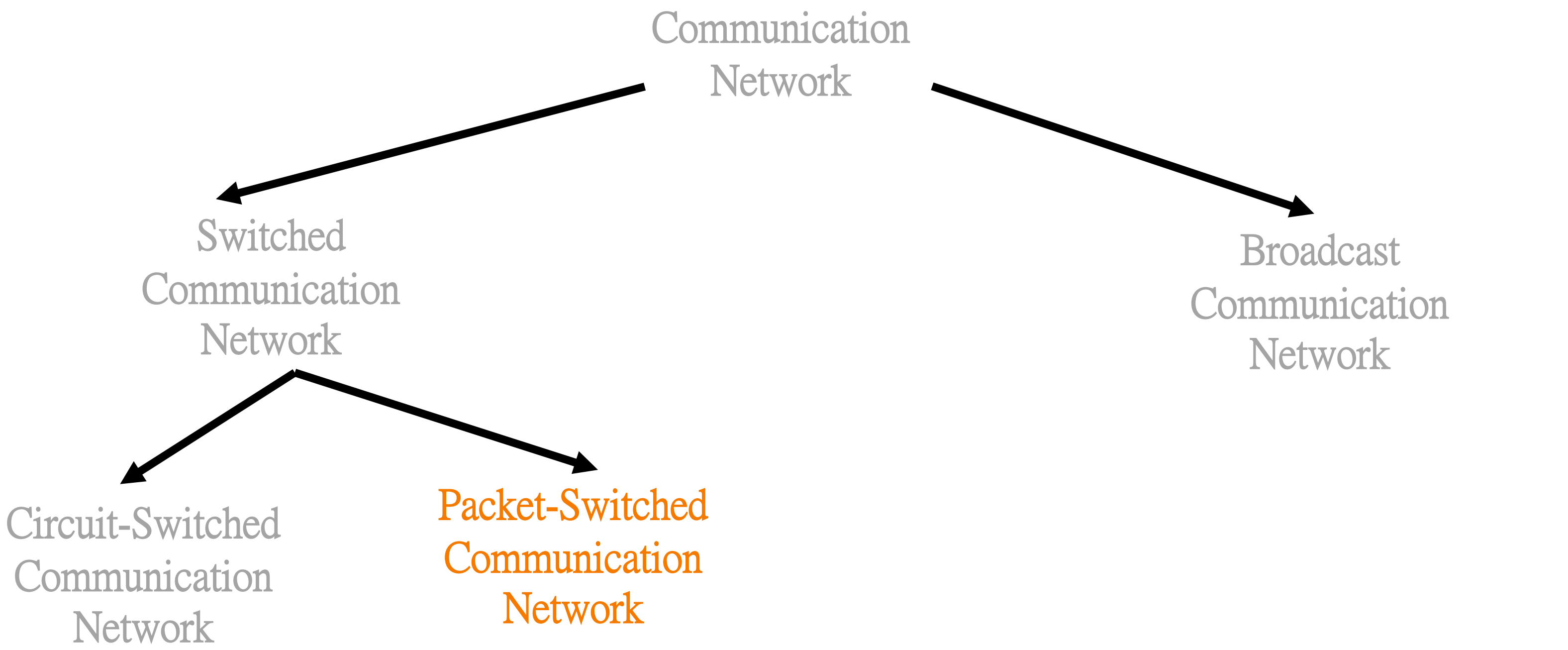
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



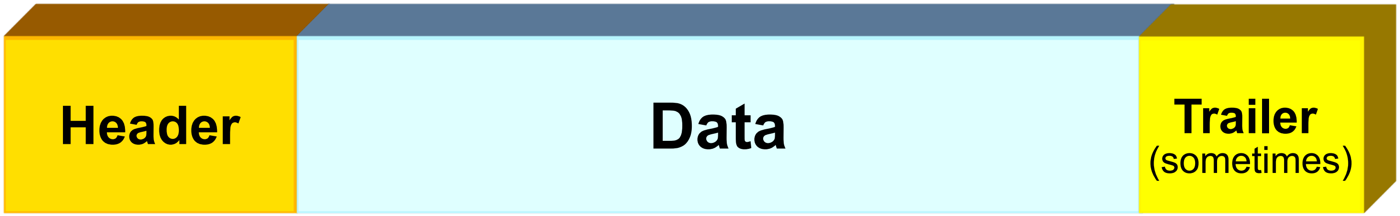
Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



Packet Switching

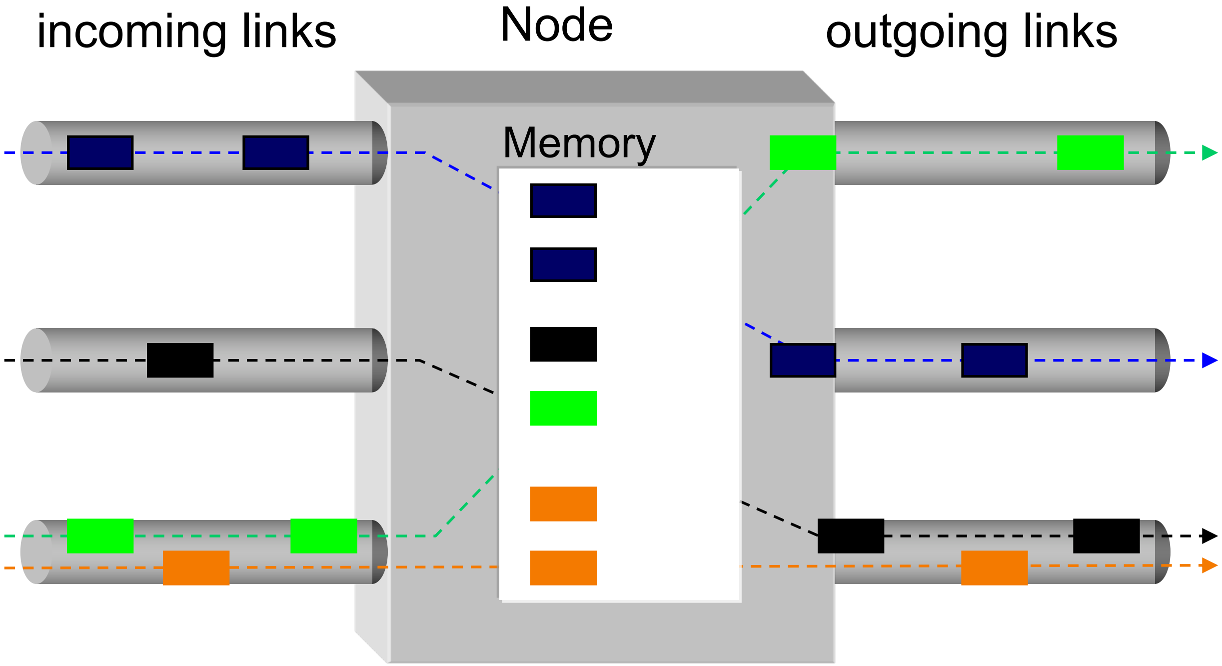
- Data sent as chunks of formatted bit-sequences (**Packets**)
- Packets have following structure:



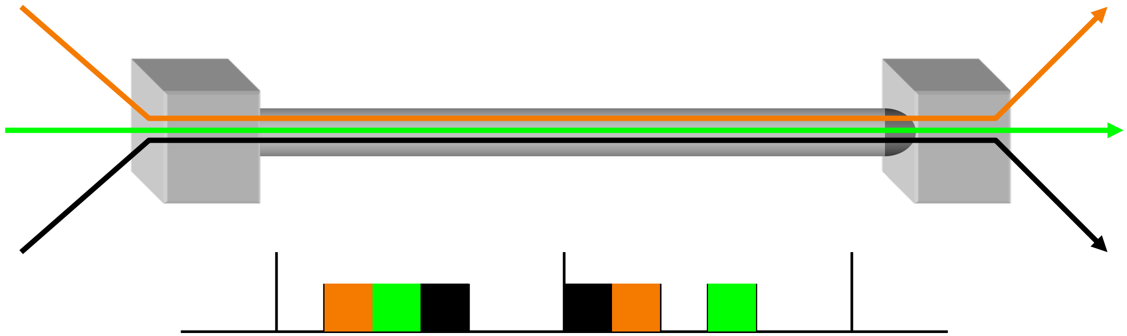
  - Header and Trailer carry control information (e.g., destination address, checksum)
- Each packet traverses the network from node to node along some path (**Routing**) based on header info.
- Usually, once a node receives the entire packet, it stores it (hopefully briefly) and then forwards it to the next node (**Store-and-Forward Networks**)

Packet Switching

- Node in a packet switching network



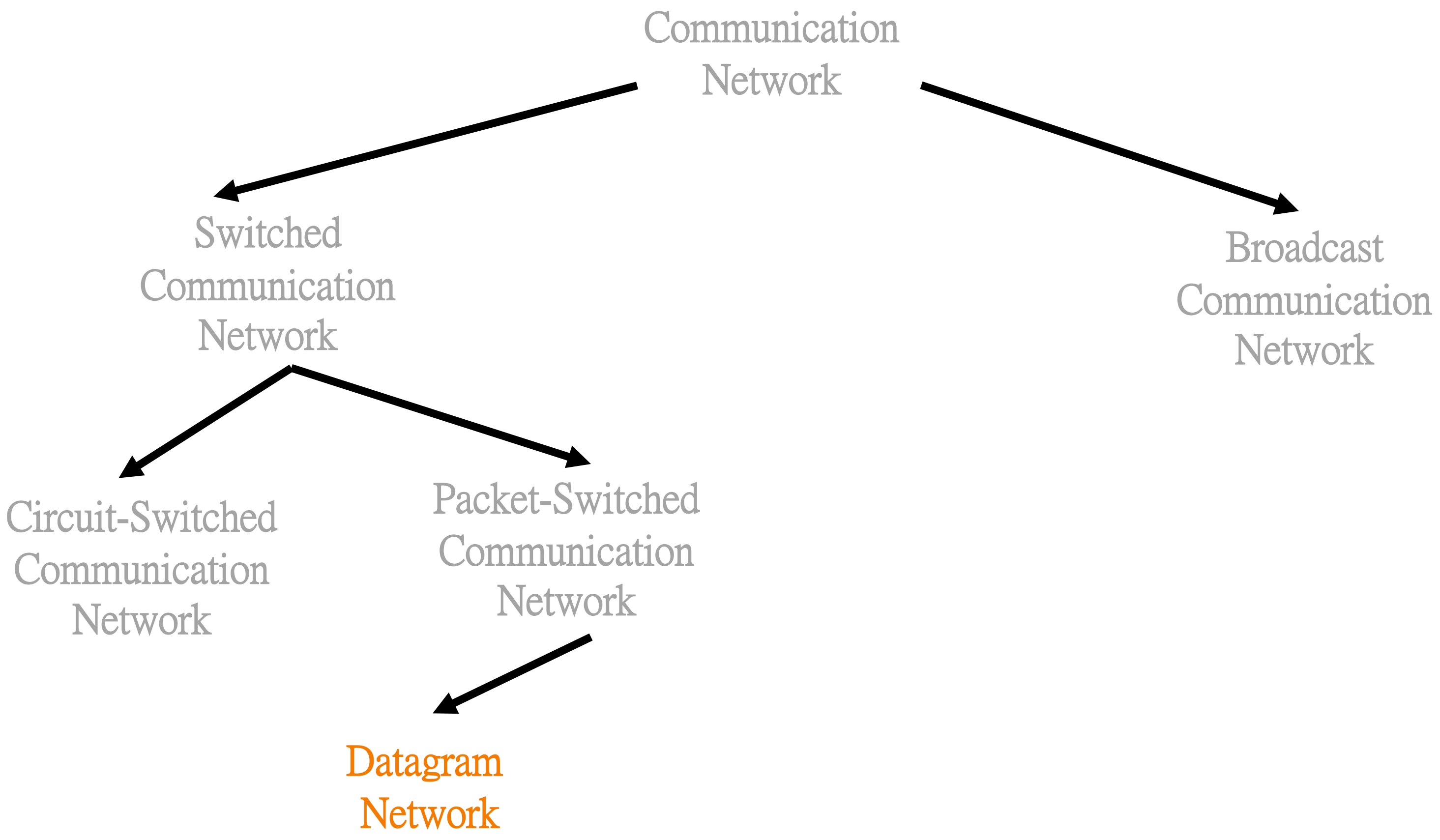
Packet Switching: Multiplexing/Demultiplexing



- How to tell packets apart?
  - Use meta-data (header) to describe data
- No reserved resources; dynamic sharing
  - Single flow can use *the entire link capacity* if it is alone
  - This leads to increased efficiency

Taxonomy of Communication Networks

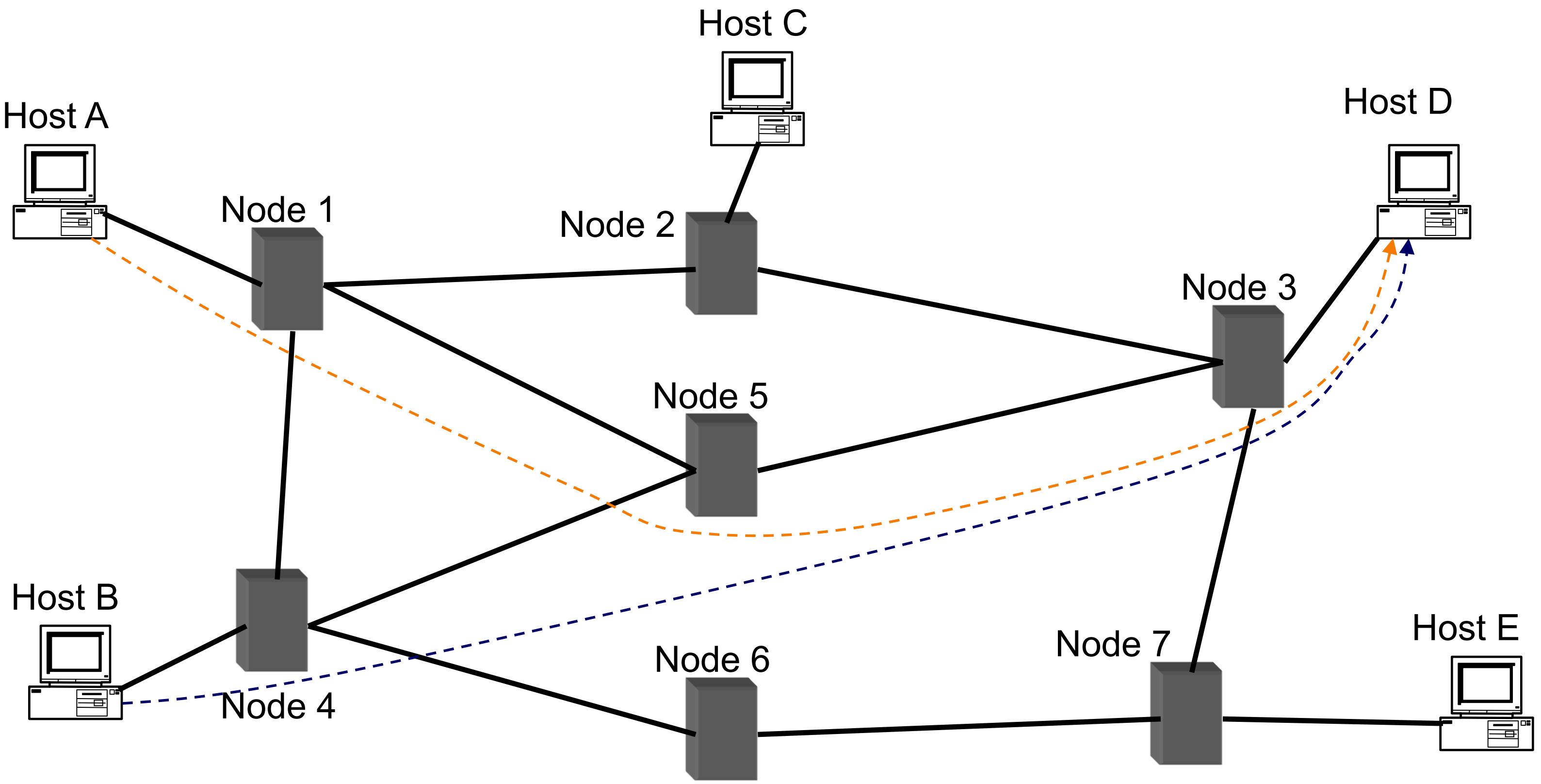
- Communication networks can be classified based on the way in which the nodes exchange information:



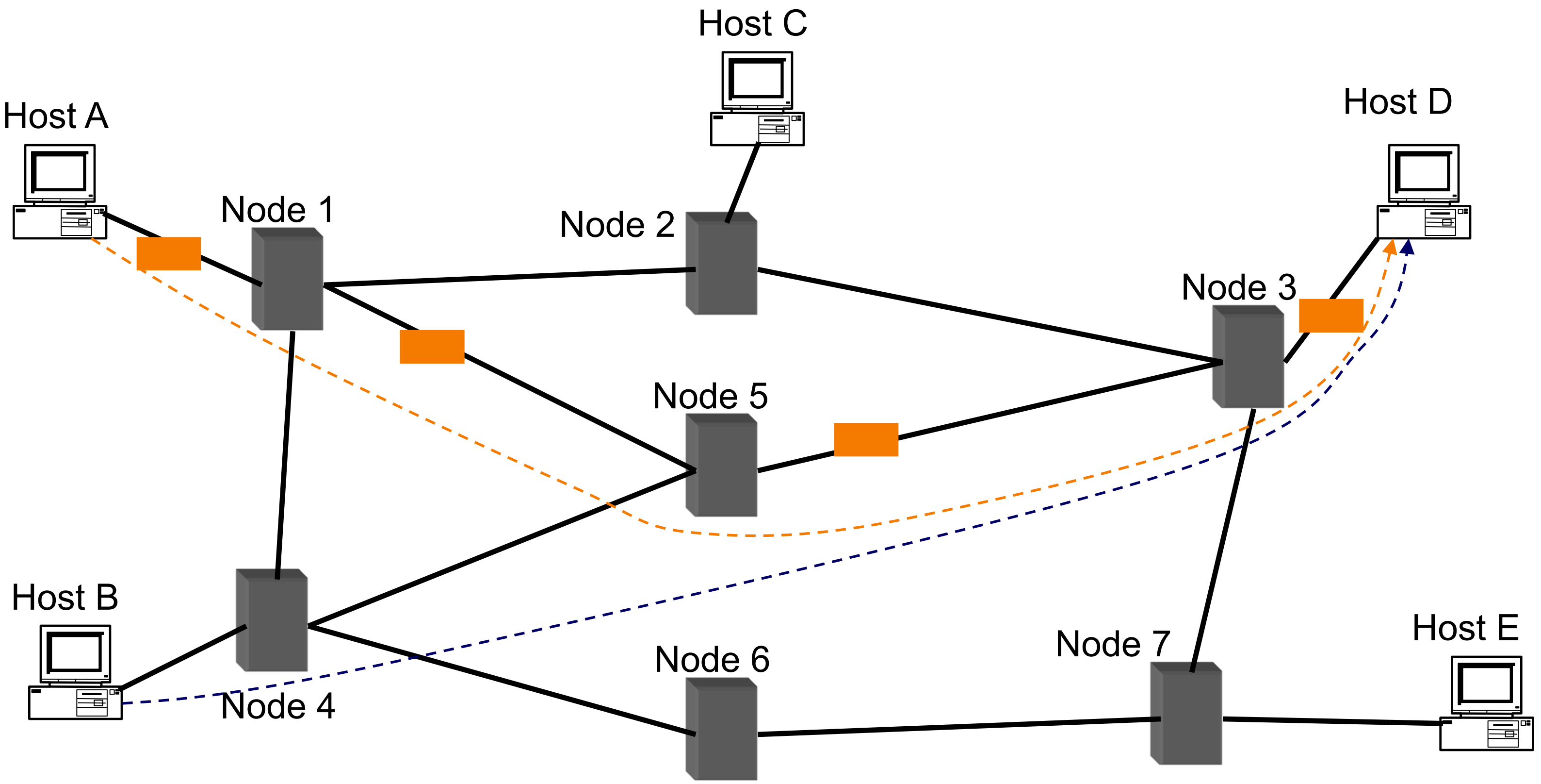
Datagram Packet Switching

- Each packet is independently switched
  - Each packet header contains full destination address
  - Routers/switches make independent routing decisions
- No resources are pre-allocated (reserved) in advance
- Leverages "statistical multiplexing"
  - Gambling that packets from different conversations won't all arrive at the same time, so we don't need enough capacity for all of them at their peak transmission rate
  - *Assuming independence of traffic sources*, can compute probability that there is enough capacity

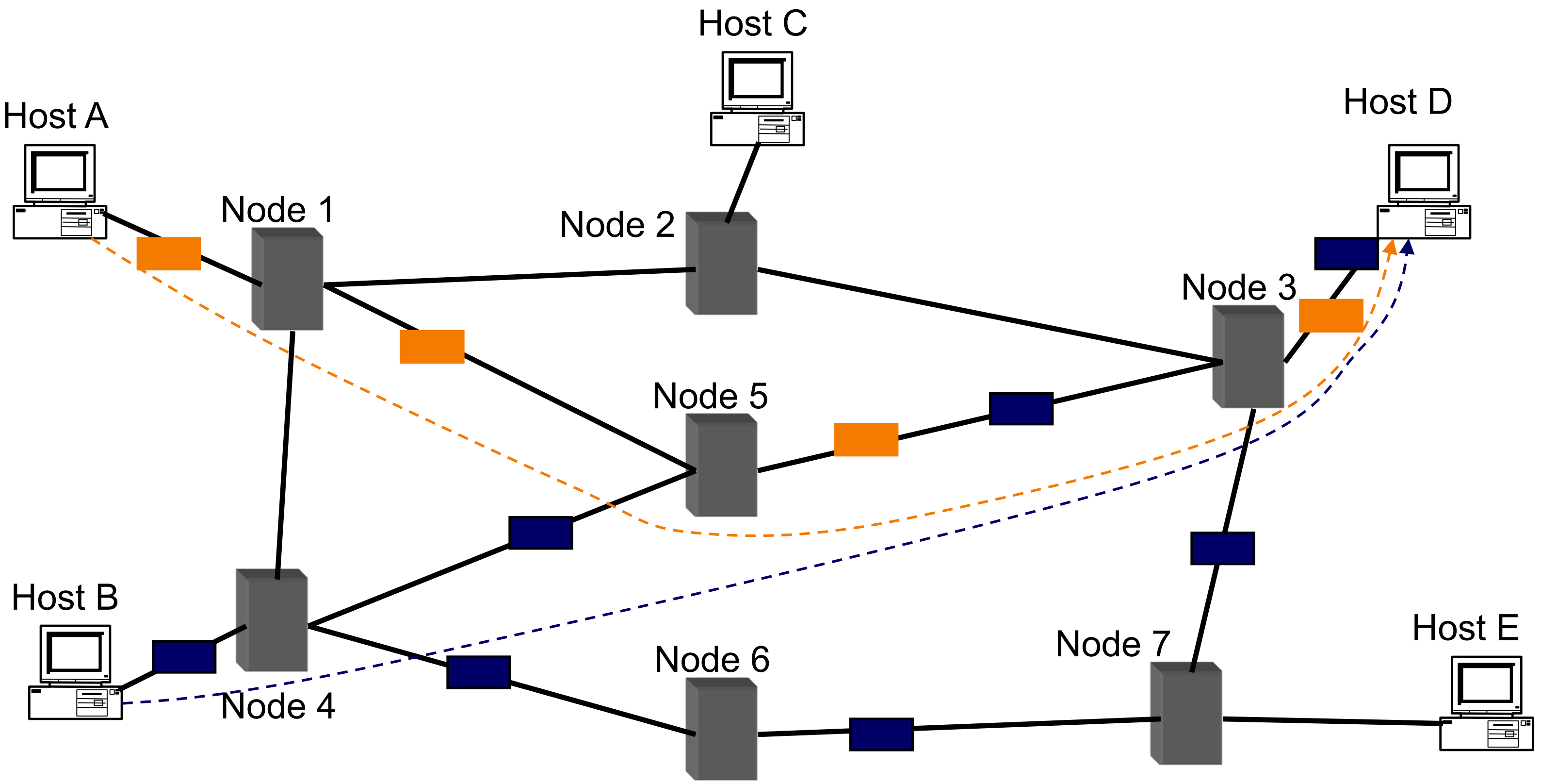
Datagram Packet Switching



Datagram Packet Switching

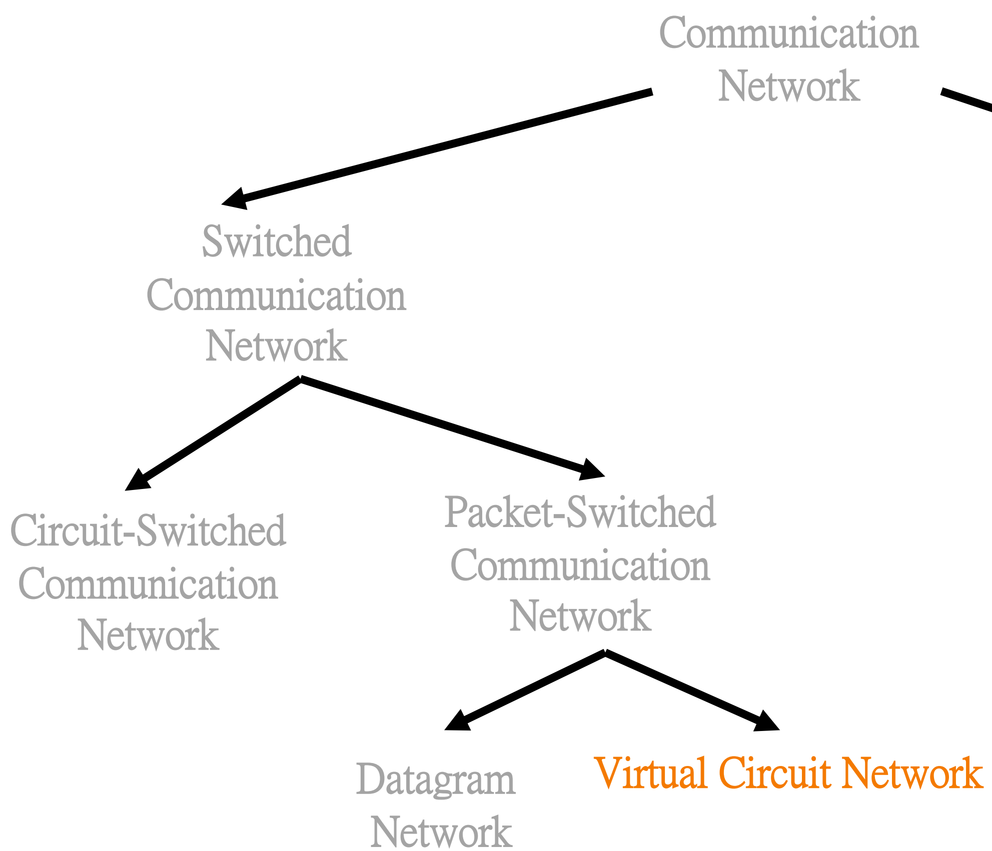


Datagram Packet Switching



Taxonomy of Communication Networks

- Communication networks can be classified based on the way in which the nodes exchange information:



A hybrid of circuits and packets; headers include a “circuit identifier” established during a setup phase

Basics of Datagram Networks

Application Protocol

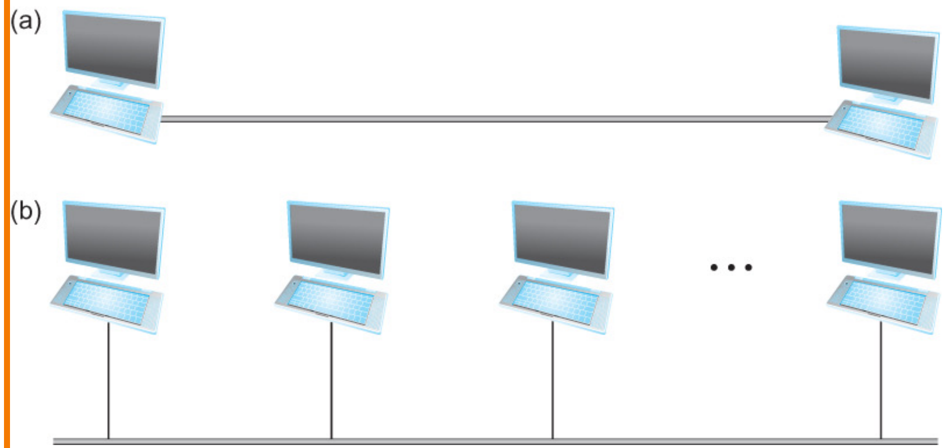
- URL
  - Uniform resource locater
  - http://www.cs.princeton.edu/~llp/index.html
- HTTP
  - Hyper Text Transfer Protocol
- TCP
  - Transmission Control Protocol
- 17 messages for one URL request
  - 6 to find the IP (Internet Protocol) address
  - 3 for connection establishment of TCP
  - 4 for HTTP request and acknowledgement
    - Request: I got your request and I will send the data
    - Reply: Here is the data you requested; I got the data
  - 4 messages for tearing down TCP connection

Requirements

- Application Programmer
  - List the services that his application needs: delay bounded delivery of data
- Network Designer
  - Design a cost-effective network with sharable resources
- Network Provider
  - List the characteristics of a system that is easy to manage

Connectivity

- Need to understand the following terminologies
  - Scale
  - Link
  - Nodes
  - Point-to-point
  - Multiple access
  - Switched Network
    - Circuit Switched
    - Packet Switched
  - Packet, message
  - Store-and-forward



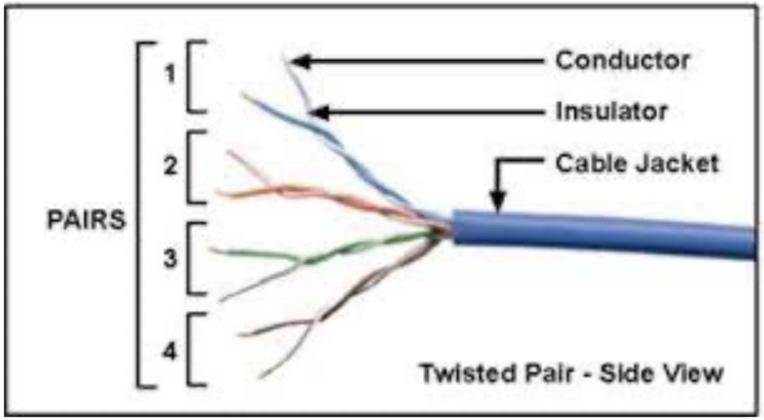
(a) **Point-to-point**

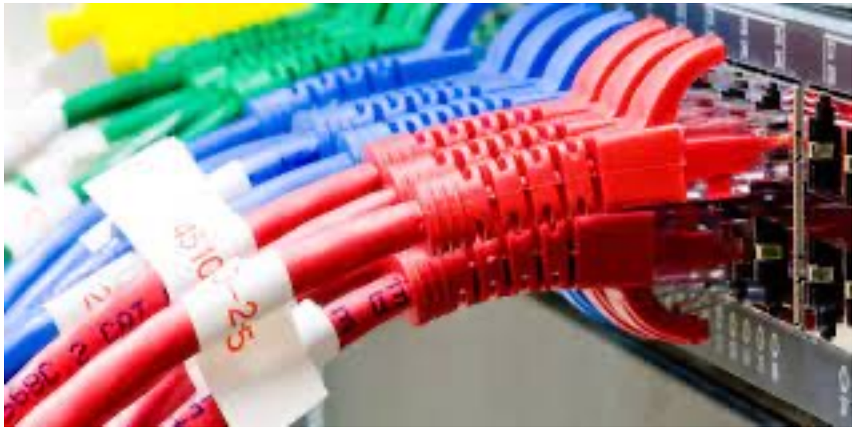
(b) **Multiple access**

Nodes and Links

- Link: transmission technology
  - Twisted pair, optical, radio, whatever
- Node: computational devices on end of links
  - Host: general-purpose computer
  - Network node: switch or router

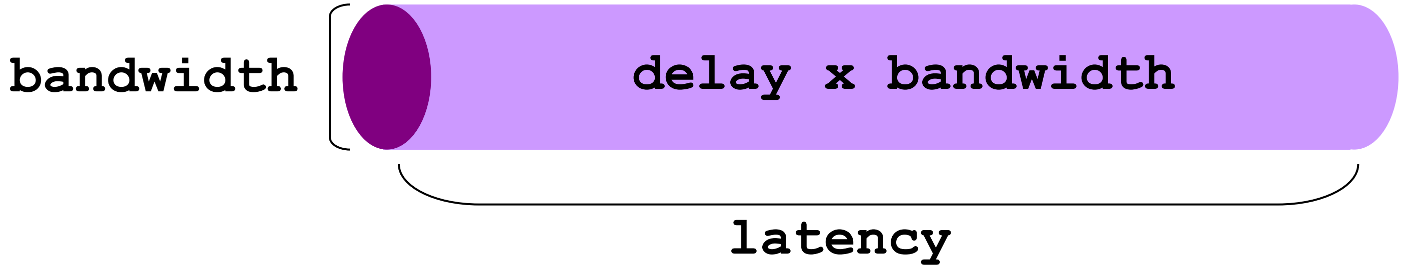
Copper Cable (Twisted Pair vs Co-ax)





Properties of Links

- Latency (delay)
  - Propagation time for data sent along the link
  - Corresponds to the “length” of the link
- Bandwidth (capacity)
  - Amount of data sent (or received) per unit time
  - Corresponds to the “width” of the link
- Bandwidth-delay product: (BDP)
  - Amount of data that can be “in flight” at any time
  - Propagation delay × bits/time = total bits in link



Examples of Bandwidth-Delay

- Same city over slow link:
  - B~100mbps
  - L~.1msec
  - BDP ~ 10000bits ~ 1.25MBytes

- Cross-country over fast link:
  - B~10Gbps
  - L~10msec
  - BDP ~ 10^8bits ~ 12.5GBytes

Examples of Transmission Times

- 1500 byte packet over 14.4k modem: ~1 sec
- 1500 byte packet over 10Gbps link: ~10^{-6}sec

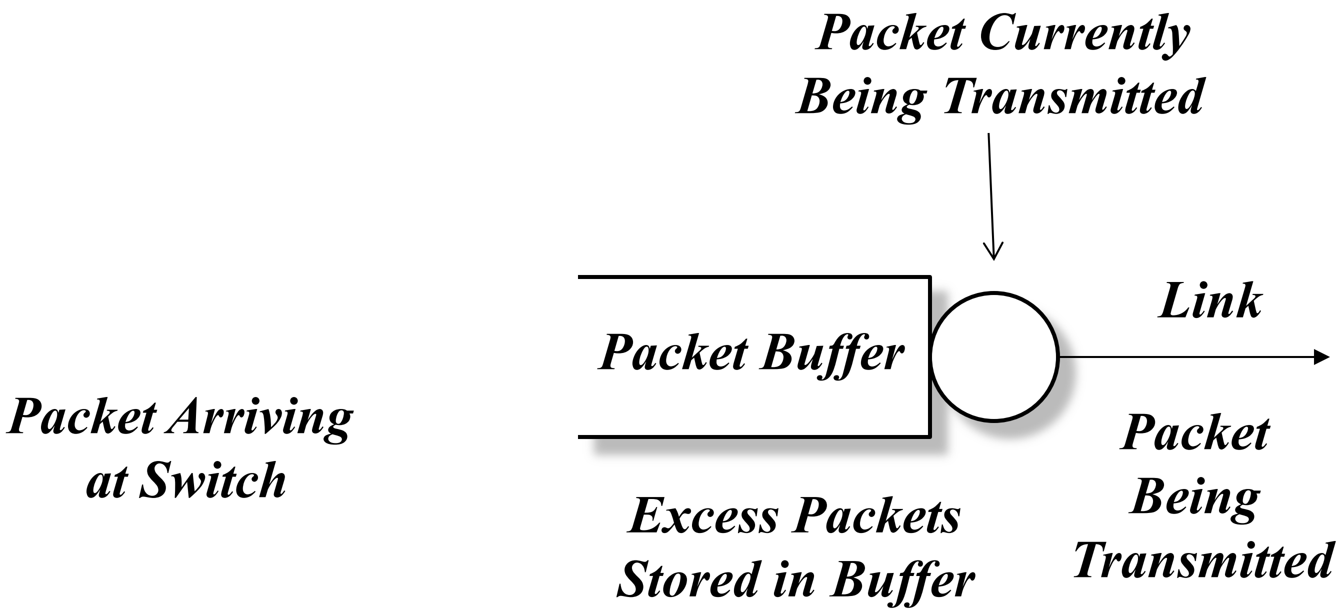
Utilization

- Fraction of time link is busy transmitting
  - Often denoted by ρ
- Ratio of arrival rate to bandwidth
  - Arrival: A bits/sec on average
  - Utilization = A/B = Arrival/Bandwidth

Packets

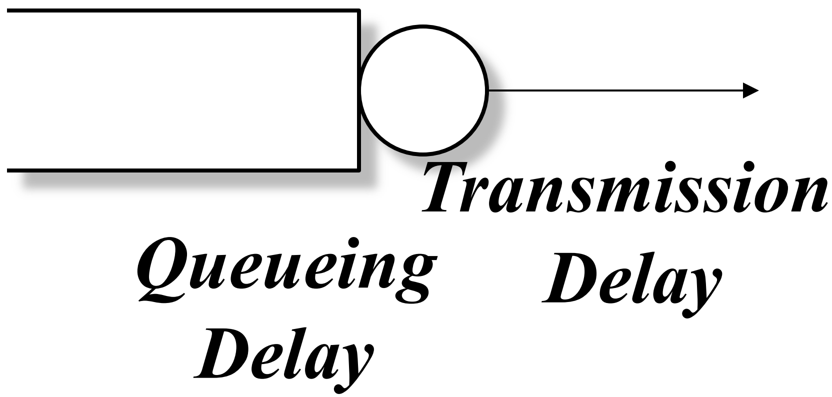
- Payload (Body)
  - Data being transferred
- Header
  - Instructions to the network for how to handle packet
  - Think of the header as an interface!

The Lifecycle of Packets



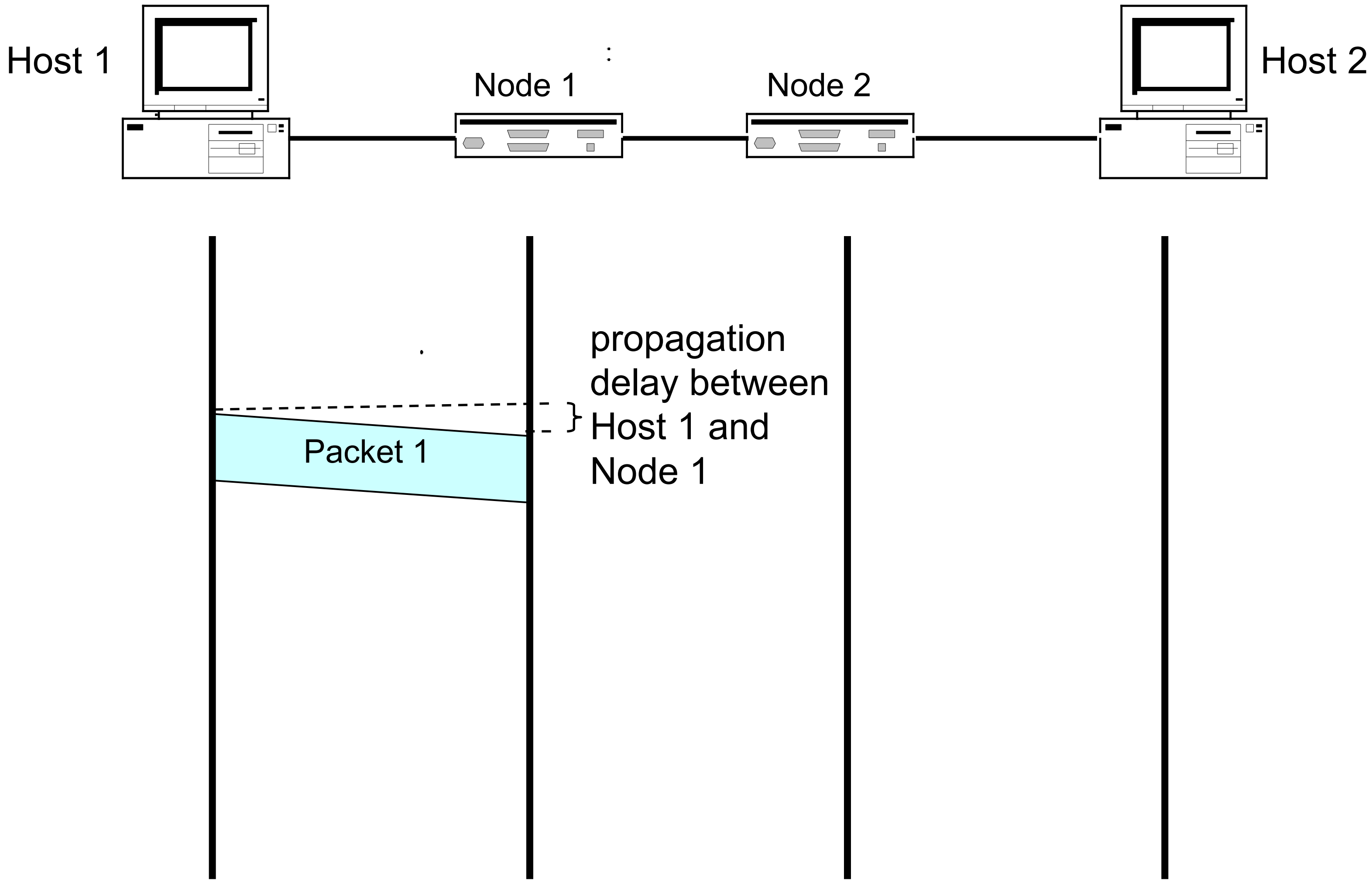
The Delays of Their Lives

Round-Trip Time (RTT) is the time it takes a packet to reach the destination and the response to return to the sender

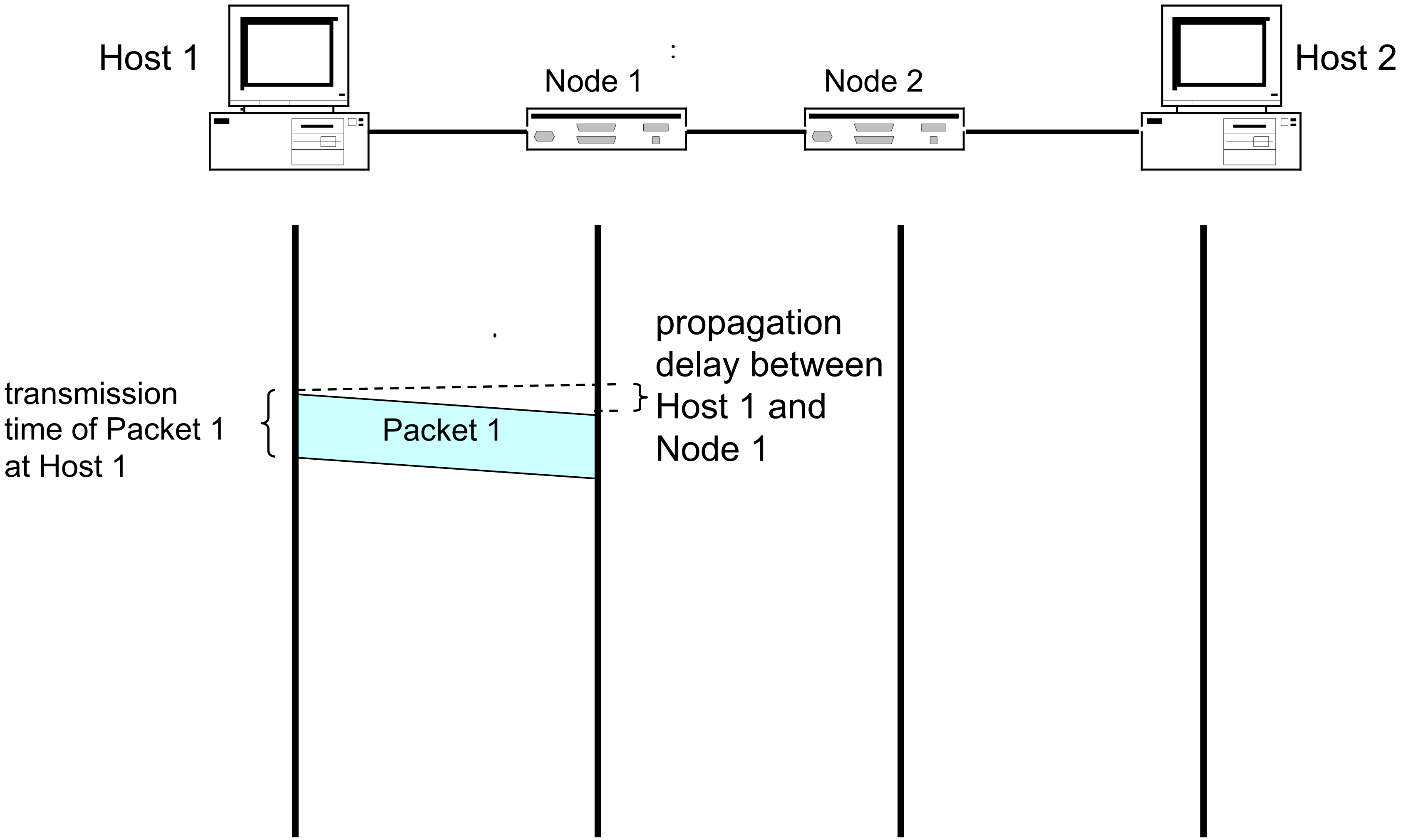


Propagation Delay is how long it takes to reach the next switch after transmission

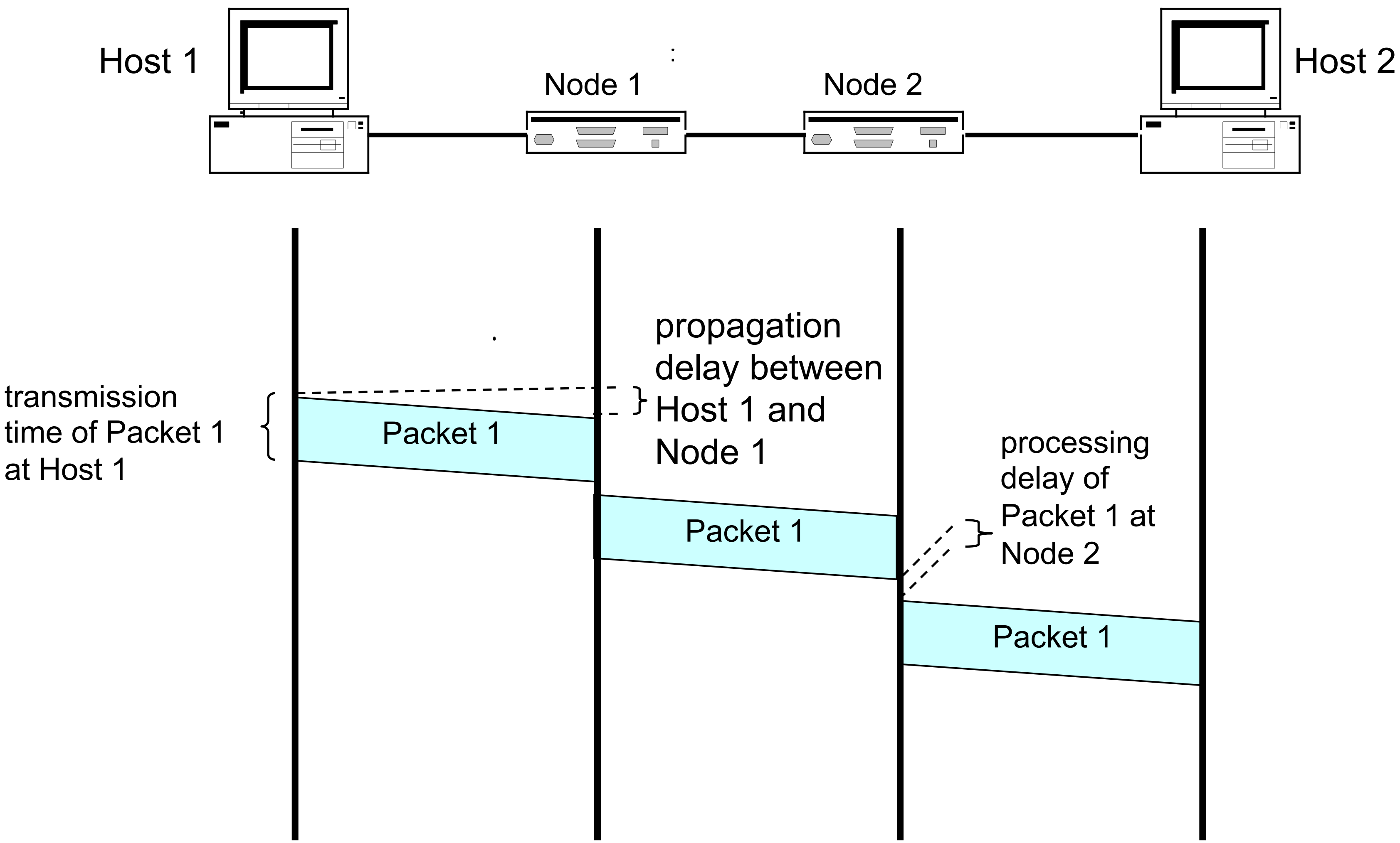
Timing of Datagram Packet Switching



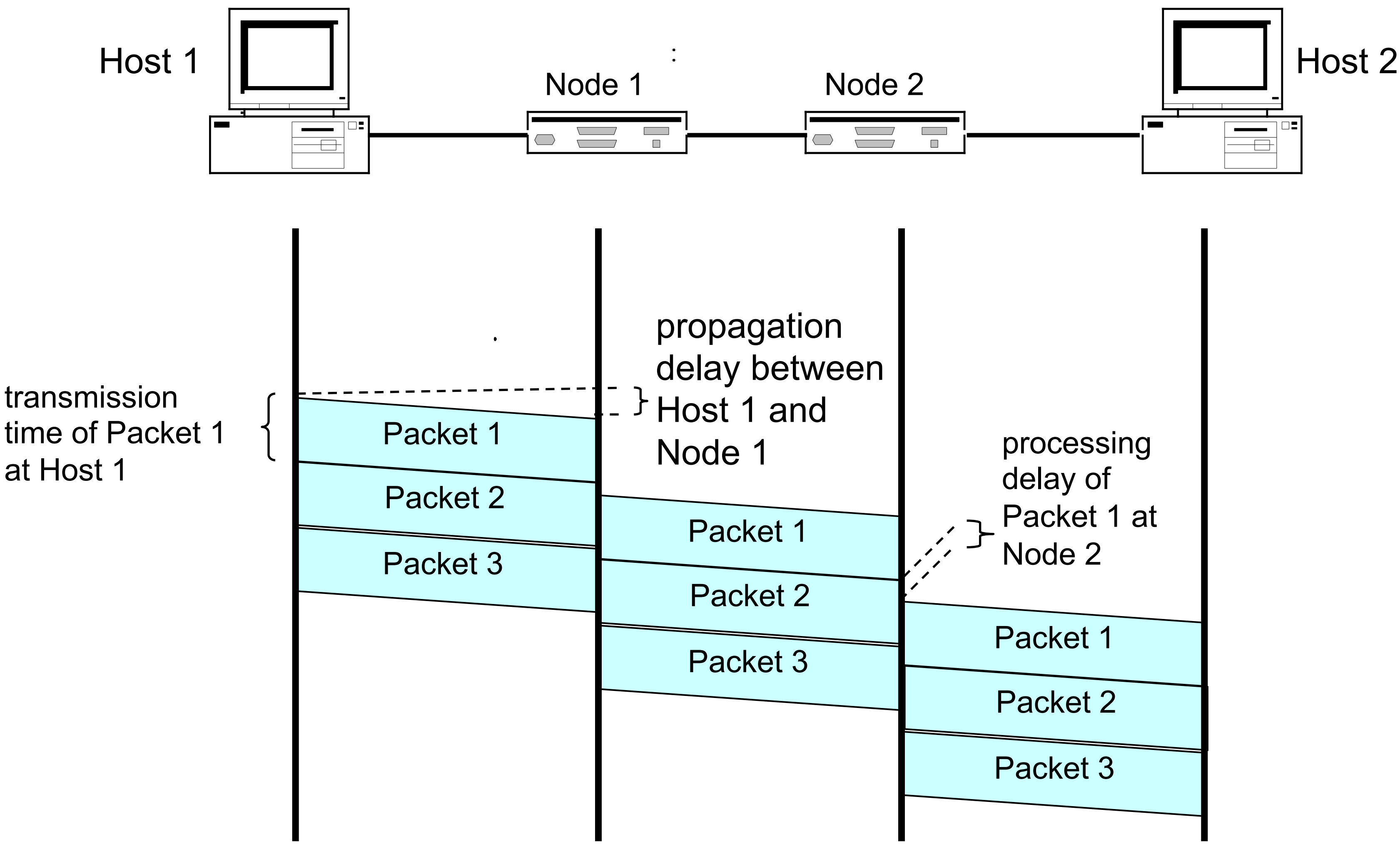
Timing of Datagram Packet Switching



Timing of Datagram Packet Switching



Timing of Datagram Packet Switching



Review of Networking Delays

- Propagation delay: latency
  - Time spent in traversing the link
  - “speed of propagation” delay
- Transmission delay:
  - Time spent being transmitted
  - Ratio of packet size to bandwidth
- Queueing delay:
  - Time spent waiting in queue
  - Ratio of total packet bits ahead in queue to bandwidth
- Roundtrip delay (RTT)
  - Total time for a packet to reach destination and a response to return to the sender

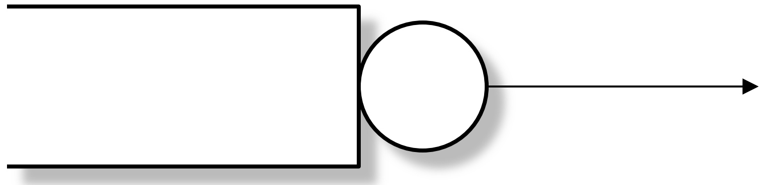
Trends

- Propagation delay?
  - No change
- Transmission delay?
  - Getting smaller!
- Queueing delay?
  - Usually smaller
- How does this affect applications?
  - CDNs work very hard to move data near clients
  - Reduces backbone bandwidth requirements
  - But also decreases latency
  - Google: time is money!

Queueing Delay

- Does not happen if packets are evenly spaced
  - And arrival rate is less than service rate

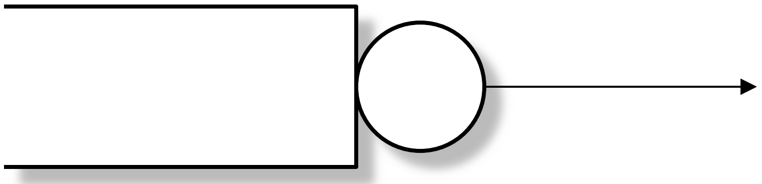
Smooth Arrivals = No Queueing Delays



Queueing Delay

- Does not happen if packets are evenly spaced
  - And arrival rate is less than service rate
- Queueing delay caused by “packet interference”
  - Burstiness of arrival schedule
  - Variations in packet lengths

Bursty Arrivals = Queueing Delays



There is substantial queueing delay even though link is underutilized

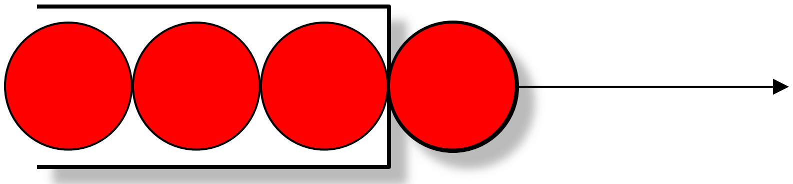
Queueing Delay Review

- Does not happen if packets are evenly spaced
  - And arrival rate is less than service rate
- Queueing delay caused by “packet interference”
  - Burstiness of arrival schedule
  - Variations in packet lengths
- Made worse at high load
  - Less “idle time” to absorb bursts
  - Think about traffic jams in rush hour….

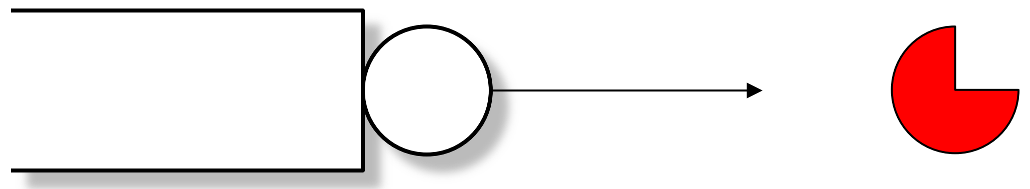
Jitter

- Difference between minimum and maximal delay
- Latency plays no role in jitter
  - Nor does transmission delay for same sized packets
- Jitter typically just differences in queueing delay
- Why might an application care about jitter?

Packet Losses: Buffers Full



Packet Losses: Corruption



Basic Queueing Theory Terminology

- Arrival process: how packets arrive
  - Average rate A
  - Peak rate P
- Service process: transmission times
  - Average transmission time
  - For networks, function of packet size
- W: average time packets wait in the queue
  - W for "waiting time"
- L: average number of packets waiting in the queue
  - L for "length of queue"
- Two different quantities

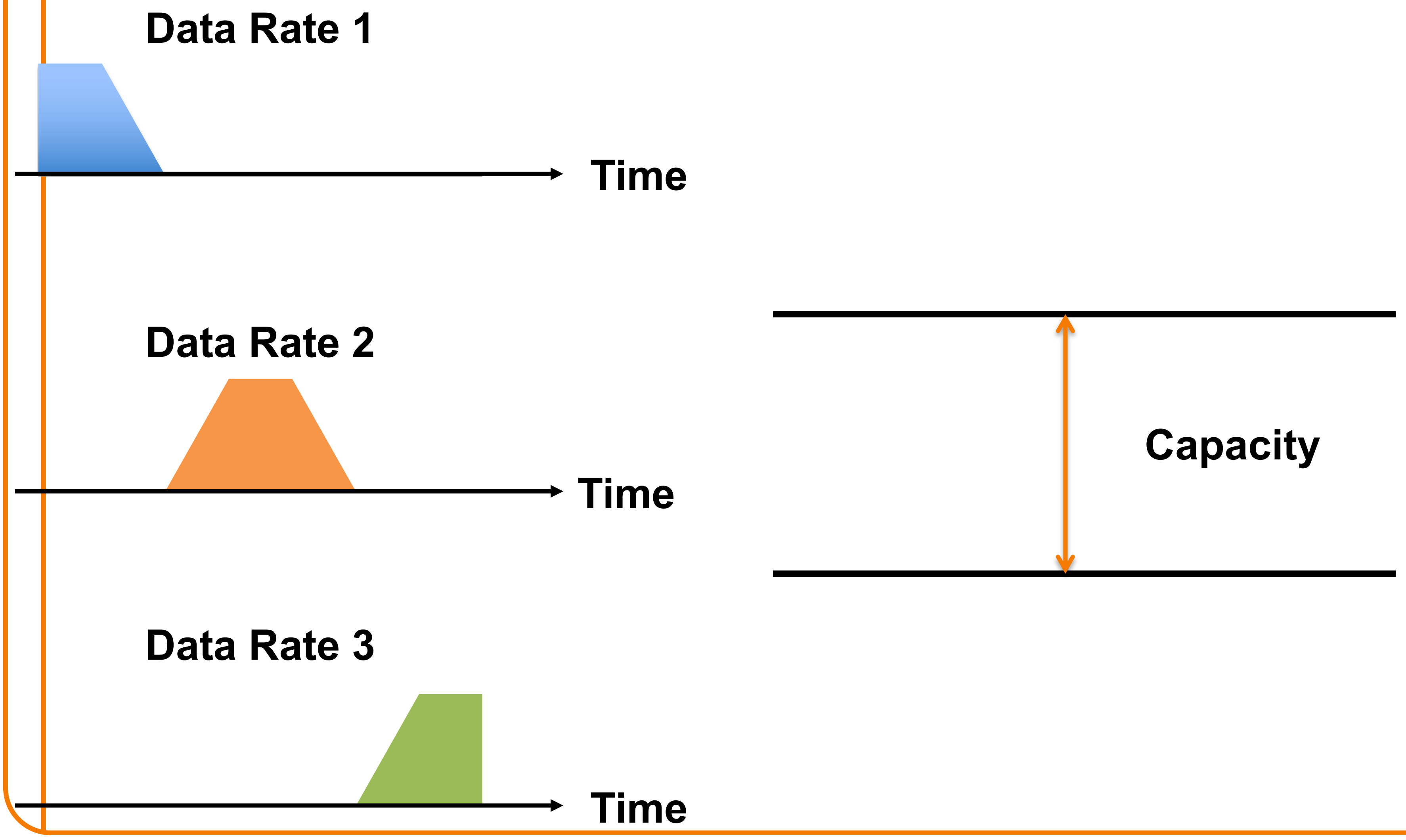
Little's Law (1961)

$$L = A \times W$$

- Compute L: count packets in queue every second
  - How often does a single packet get counted? W times
- Could compute L differently
  - On average, every packet will be counted W times
  - The average arrival rate determines how frequently this total queue occupancy should be added to the total
- Why do you care?
  - Easy to compute L, harder to compute W

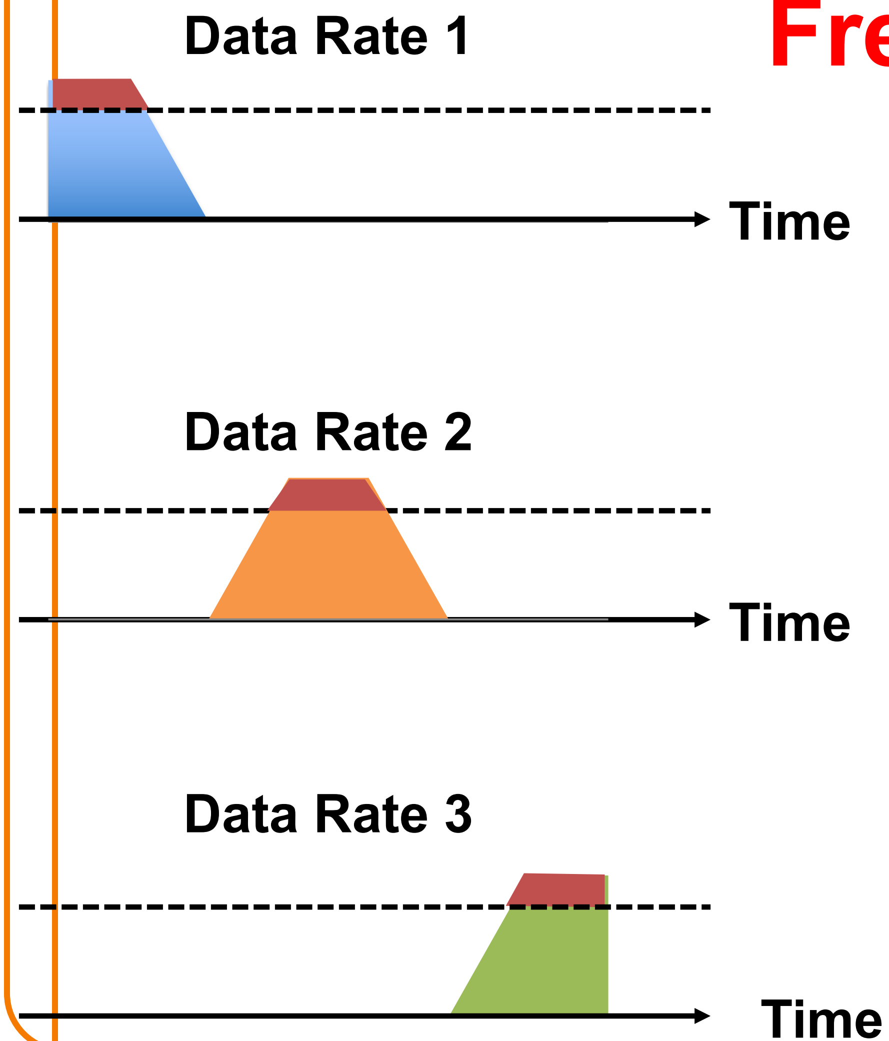
Statistical Multiplexing

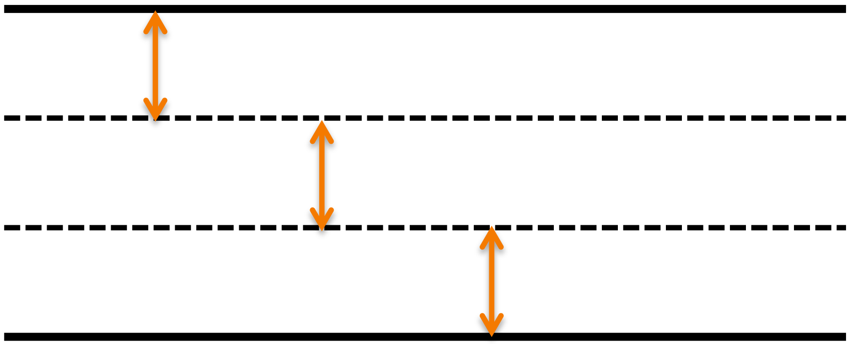
Three Flows with Bursty Arrivals



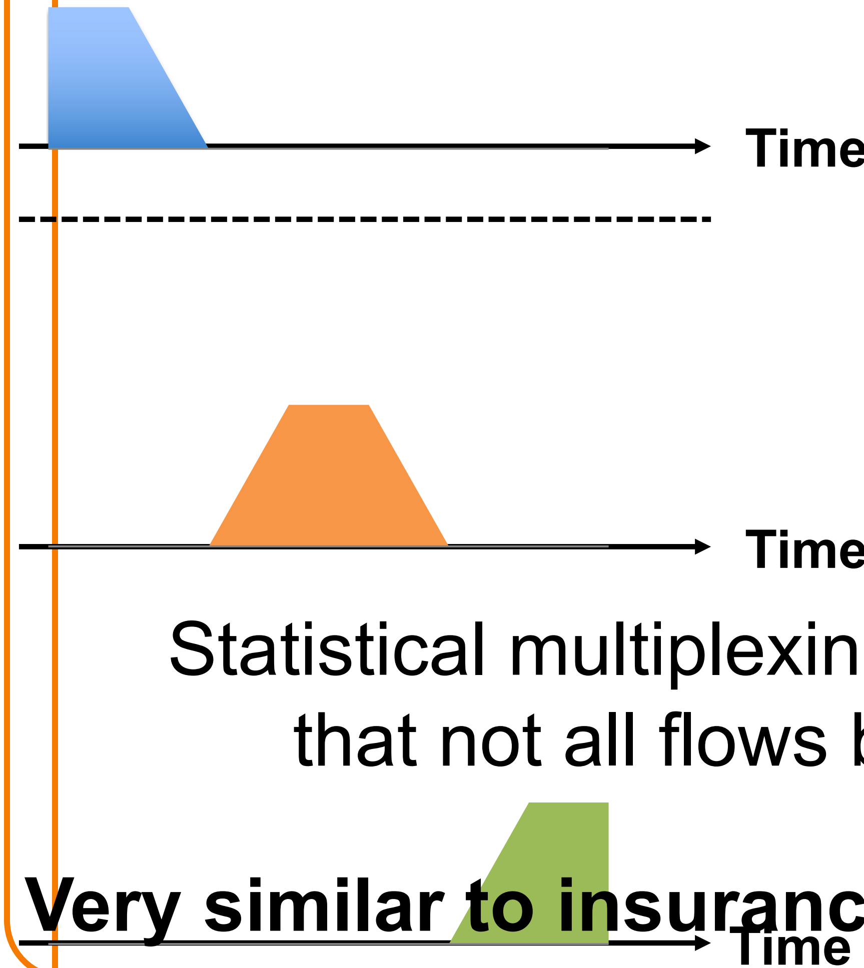
When Each Flow Gets 1/3rd of Capacity

Frequent Overloading





When Flows Share Total Capacity



No Overloading

Statistical multiplexing relies on the assumption that not all flows burst at the same time.

Very similar to insurance, and has same failure case

Recurrent theme in computer science

- Greater efficiency through “sharing”
  - Statistical multiplexing
- Phone network rather than dedicated lines
  - Ancient history
- Packet switching rather than circuits
  - Today’s lecture
- Cloud computing
  - Shared datacenters, rather than single PCs

Internet History

Timeline

1961 Baran and Kleinrock advocate packet switching

1962 Licklider's vision of Galactic Network

1965 Roberts connects two computers via phone

1967 Roberts publishes vision of ARPANET

1969 BBN installs first IMP at UCLA

IMP: Interface Message Processor

1971 Network Control Program (protocol)

1972 Public demonstration of ARPANET

The beginning of the Internet revolution

- Kleinrock's group at UCLA tried to log on to Stanford computer: His recollection of the event…
- We typed the L…
  - "Do you see the L?"
  - "Yes, we see the L."
- We typed the O…
  - "Do you see the O?"
  - "Yes, we see the O."
- Then we typed the G…
  - …and the system crashed!

Timeline continued…

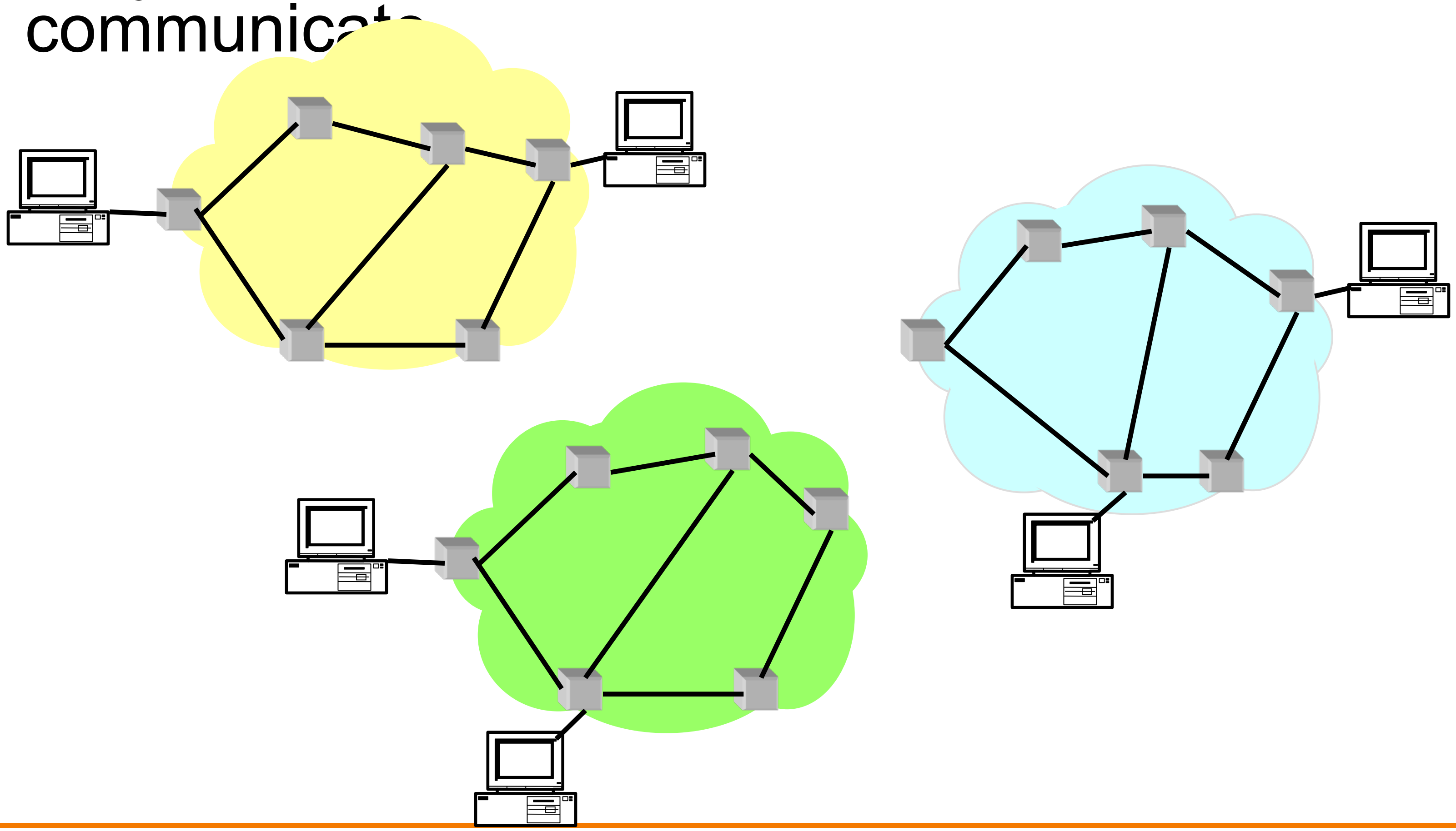
1972 Email invented

1972 Telnet introduced

1972 Kahn advocates Open Architecture networking

The Problem

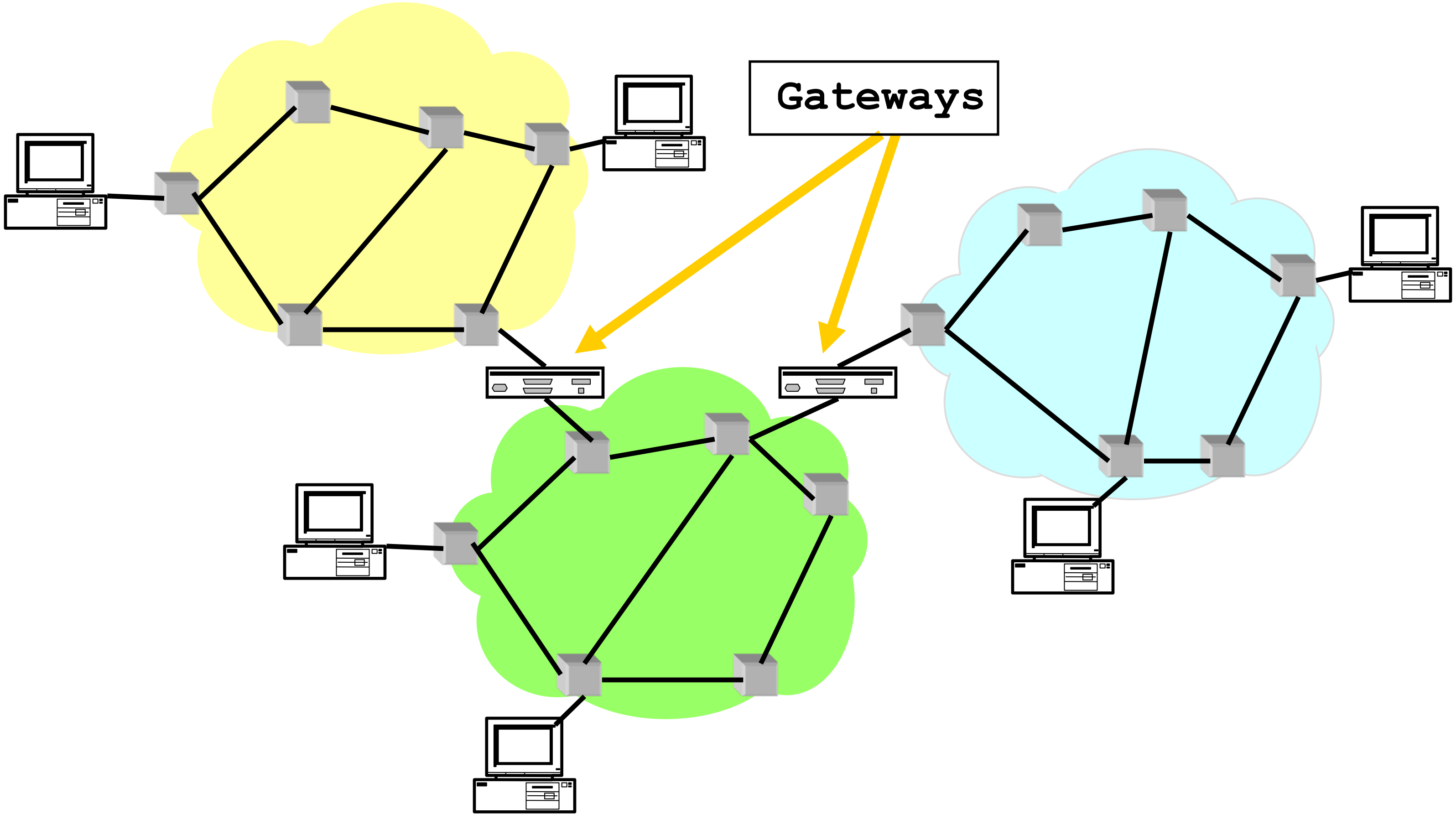
- Many different packet-switching networks
- Only nodes on the same network could communicate



Kahn's Rules for Interconnection

- Each network is independent and must not be required to change (why?)
- Best-effort communication (why?)
- Boxes (routers) connect networks
- No global control at operations level (why?)

Solution



Kahn’s vision

- Kahn imagined there would be only a few networks (~20) and thus only a few routers
- He was wrong
  - Why?
- Imagined gateways would “translate” between networks
  - We think of it as all routers supporting IP

Timeline continued….

1973 FTP introduced

1974 Cerf and Kahn paper on TCP/IP

1980 TCP/IP adopted as defense standard

1983 Global NCP to TCP/IP flag day

198x XNS, DECbit, and other protocols

1984 Janet (British research network)

1985 NSFnet (picks TCP/IP)

198x Internet meltdowns due to congestion

1986 Van Jacobson saves the Internet (BSD TCP)

Unsung hero of Internet: David D. Clark

- Chief Architect 1981-1988
- Great consistency of vision
- Kept the Internet true to its basic design principles
- Authored what became known as the End-to-end principle (next lecture)
- Conceives <u>and</u> articulates architectural concepts
  - Read his “Active Networking and End-To-End Arguments”
- Perhaps the only “irreplaceable” Internet pioneer

Timeline continued…

1988 Deering and Cheriton propose multicast

1989 Birth of the web….Tim Berners-Lee

Why did it take physicist to invent web?

- Physicists are the smartest people in the world?
- Computer scientists were trying to invent nirvana
  - Well, actually Xanadu (Ted Nelson)
  - More generally, CS researchers focused on hyptertext
- Again, users didn't need what we wanted to invent
  - Think about it: a paper on the web design would have been rejected by every CS conference and journal

Timeline continued…..

1993 Search engines invented (Excite)

199x ATM rises and falls (as internetworking layer)

199x QoS rises and falls

1994 Internet goes commercial

1998 IPv6 specification

1998 Google reinvents search

200x The Internet boom and bust

2012 **EE122 enrollment suggests boom is back!**

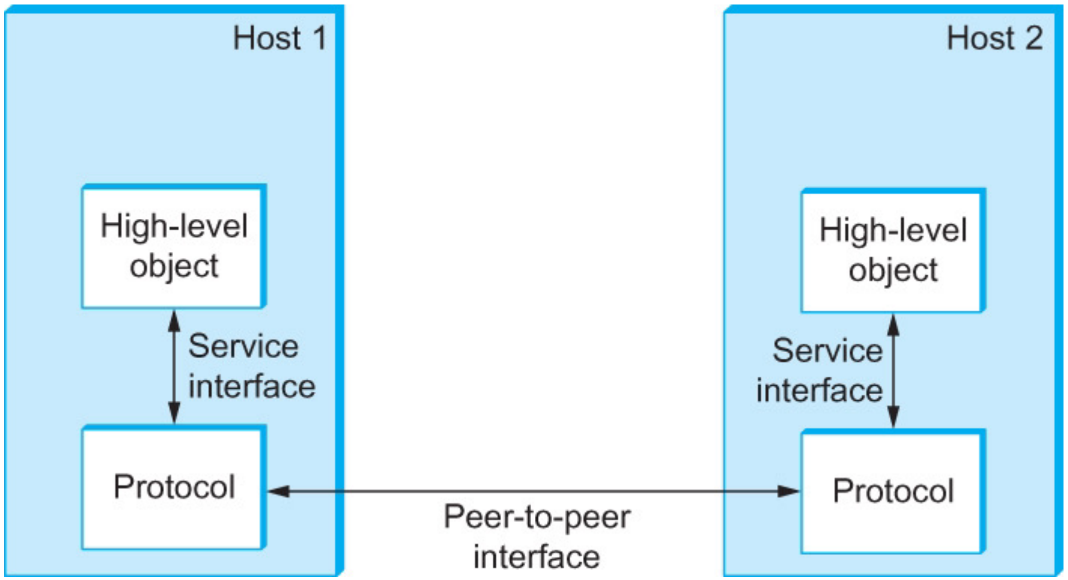
~80 in 2010 to ~200 in 2011 to ~340 in 2012

Protocols & API

Protocols

- Protocol defines the interfaces between the layers in the same system and with the layers of peer system
- Building blocks of a network architecture
- Each protocol object has two different interfaces
  - service interface: operations on this protocol
  - peer-to-peer interface: messages exchanged with peer
- Term “protocol” is overloaded
  - specification of peer-to-peer interface
  - module that implements this interface

Interfaces

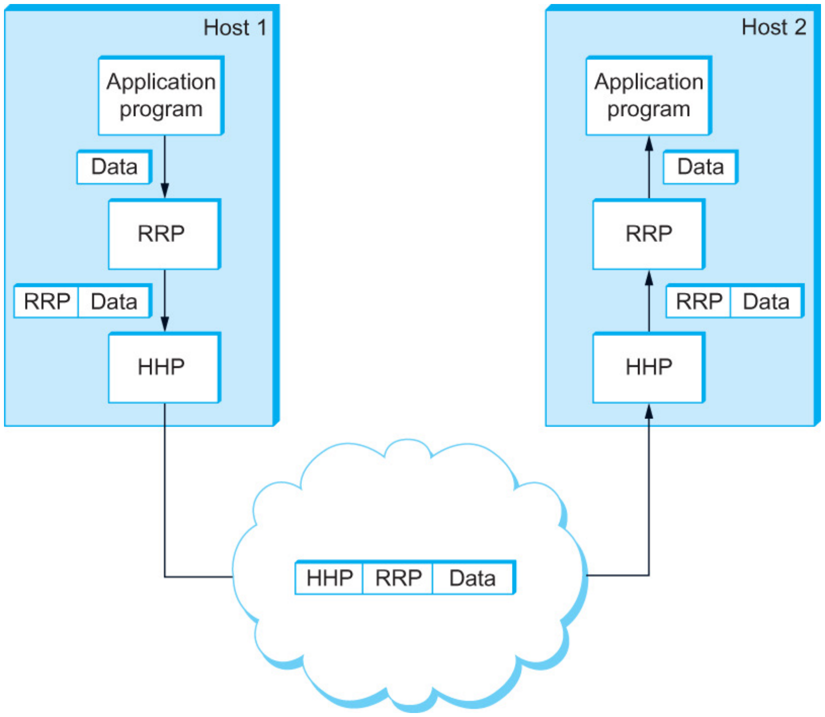


Service and Peer Interfaces

Protocols

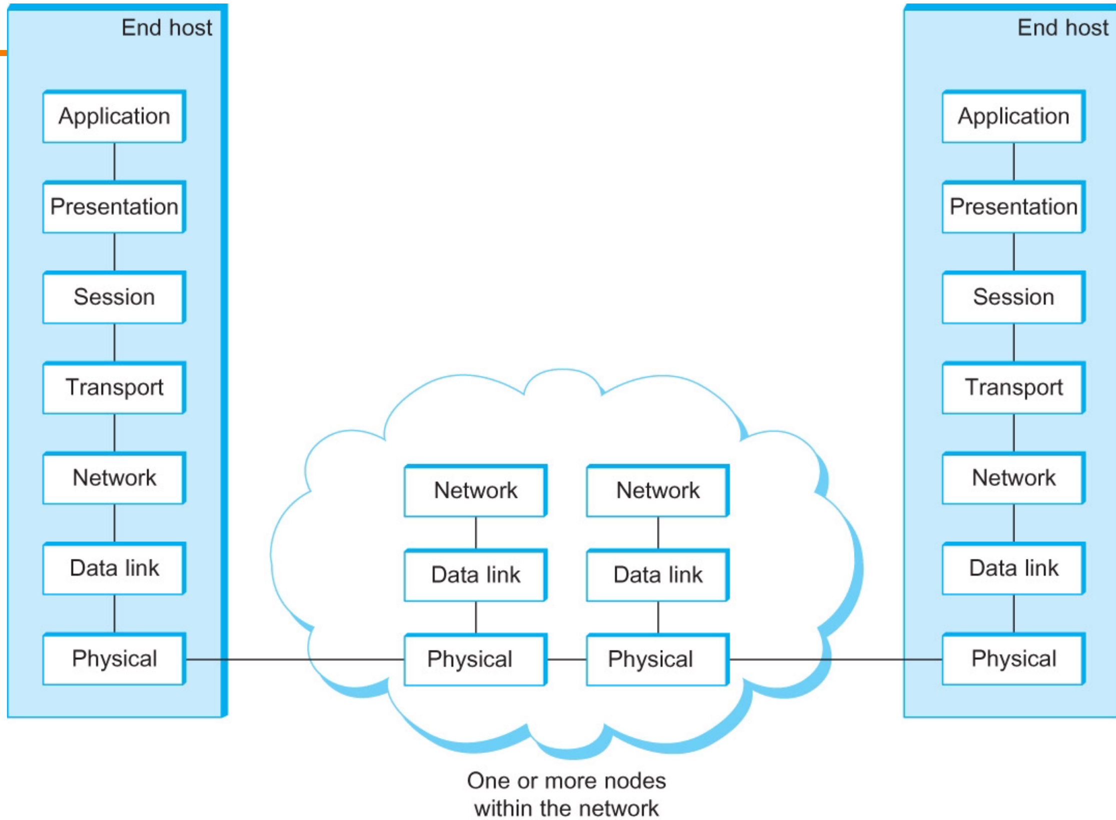
- Protocol Specification: prose, pseudo-code, state transition diagram
- Interoperable: when two or more protocols that implement the specification accurately
- IETF: Internet Engineering Task Force

Encapsulation



High-level messages are encapsulated inside of low-level messages

OSI Architecture



The OSI 7-layer Model

OSI – Open Systems Interconnection

Description of Layers

- Physical Layer
  - Handles the transmission of raw bits over a communication link
- Data Link Layer
  - Collects a stream of bits into a larger aggregate called a *frame*
  - Network adaptor along with device driver in OS implement the protocol in this layer
  - Frames are actually delivered to hosts
- Network Layer
  - Handles routing among nodes within a packet-switched network
  - Unit of data exchanged between nodes in this layer is called a *packet*

The lower three layers are implemented on all network nodes

Description of Layers

- Transport Layer
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a *message*
- Session Layer
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
- Presentation Layer
  - Concerned about the format of data exchanged between peers
- Application Layer
  - Standardize common type of exchanges

The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers

Internet Architecture

- Defined by IETF
- Three main features
  - Does not imply strict layering. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks
  - An hour-glass shape – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture
  - In order for a new protocol to be officially included in the architecture, there needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification

Application Programming Interface

- Interface exported by the network
- Since most network protocols are implemented (those in the high protocol stack) in software and nearly all computer systems implement their network protocols as part of the operating system, when we refer to the interface "*exported by the network*", we are generally referring to the interface that the OS provides to its networking subsystem
- The interface is called the network Application Programming Interface (API)

Application Programming Interface (Sockets)

- Socket Interface was originally provided by the Berkeley distribution of Unix
  - Now supported in virtually all operating systems
- Each protocol provides a certain set of *services*, and the API provides a syntax by which those services can be invoked in this particular OS

Socket

- What is a socket?
  - The point where a local application process attaches to the network
  - An interface between an application and the network
  - An application creates the socket
- The interface defines operations for
  - Creating a socket
  - Attaching a socket to the network
  - Sending and receiving messages through the socket
  - Closing the socket