Week 1: Introduction to Computer Communications

EE3017/IM2003 Computer Communications

School of Electrical and Electronic Engineering

Dr. SHAO Xuguang, Michelle

Email: XGSHAO@ntu.edu.sg

Room: S1-B1a-10

Phone: 6513-7648

Part I Topic Outline

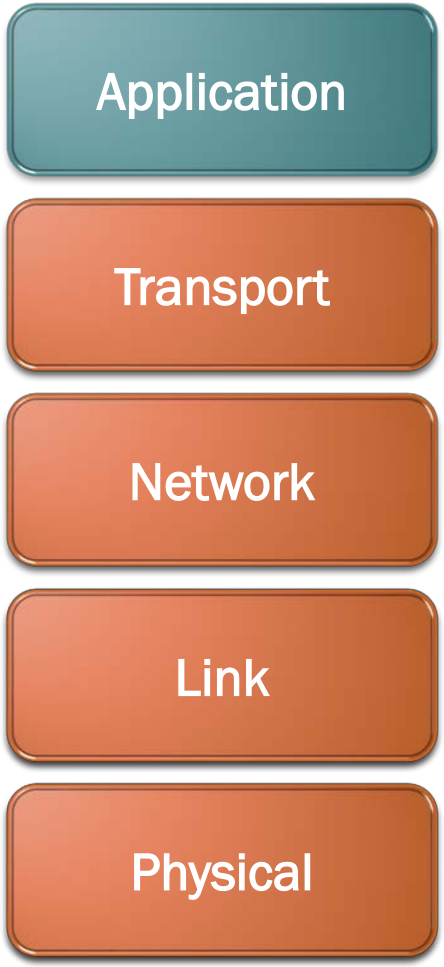
Part I Topic Outline

Introduction to Computer Communications 01

Network Architectures and Services, Switched Network, Protocol Layers

Data Communications Fundamentals 02

Data Link Layer 03



Introduction to Computer Communications

Topic I Outline

- Introduction 01
- Network Architecture 02
- Circuit Switching vs. Packet Switching 03
- Pipelining in Packet Switching Network 04
- Layered Protocol Architecture 05

Learning Objectives

By the end of this topic, you should be able to:

- Explain what a computer network is.
- Explain the network architecture.
- Describe the two major switching techniques.
- Explain the use of pipelining in packet switching networks.
- List the four types of delay in packet switching networks.
- Conduct the delay analyses in packet switching networks.
- Explain the importance of protocol layers and encapsulation.

Introduction

Introduction

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

- The Mathematical Theory of Communication

Claude E. Shannon

1916 - 2001

The History of Communication

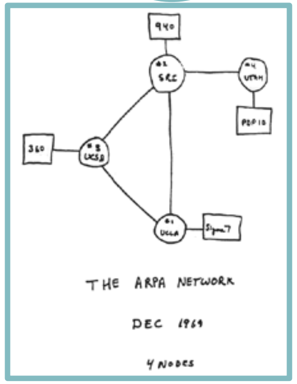
Technology	Speed	Distance
Natural	Low	Short
Papermaking and Printing		
Printing	Low	Long
Industrial Revolution		
Analog	Medium	Long
Digital Revolution		
Digital	High	Global

Historical Internet Traffic Growth

1969

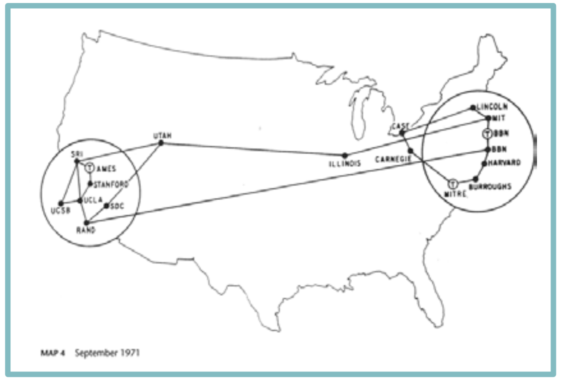
- ARPANET
- Four Nodes
- Network Control Protocol (NCP) first host-host protocol

ARPA Network



1971

First e-mail program



1972

Public demonstration

1982

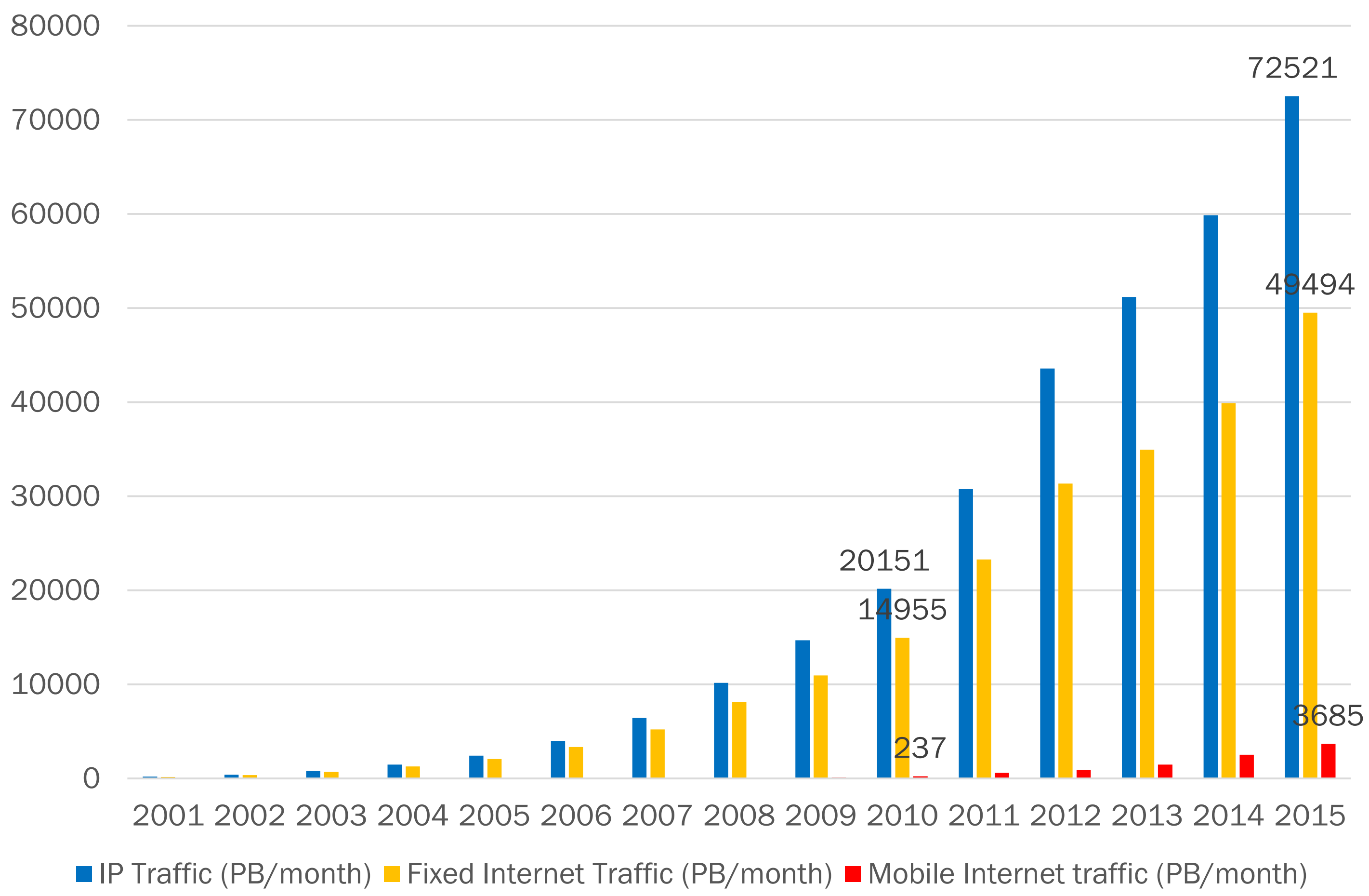
Total number of hosts: 213

1983

Start of Modern Internet: NCP replaced by TCP/IP protocols (Flag day: 1st Jan 1983)

Images retrieved from:
https://en.wikipedia.org/wiki/History_of_the_Internet

Historical Internet Traffic Growth



Data retrieved from:

https://en.wikipedia.org/wiki/Internet_traffic

Useful Resources: Cisco Visual Networking Index

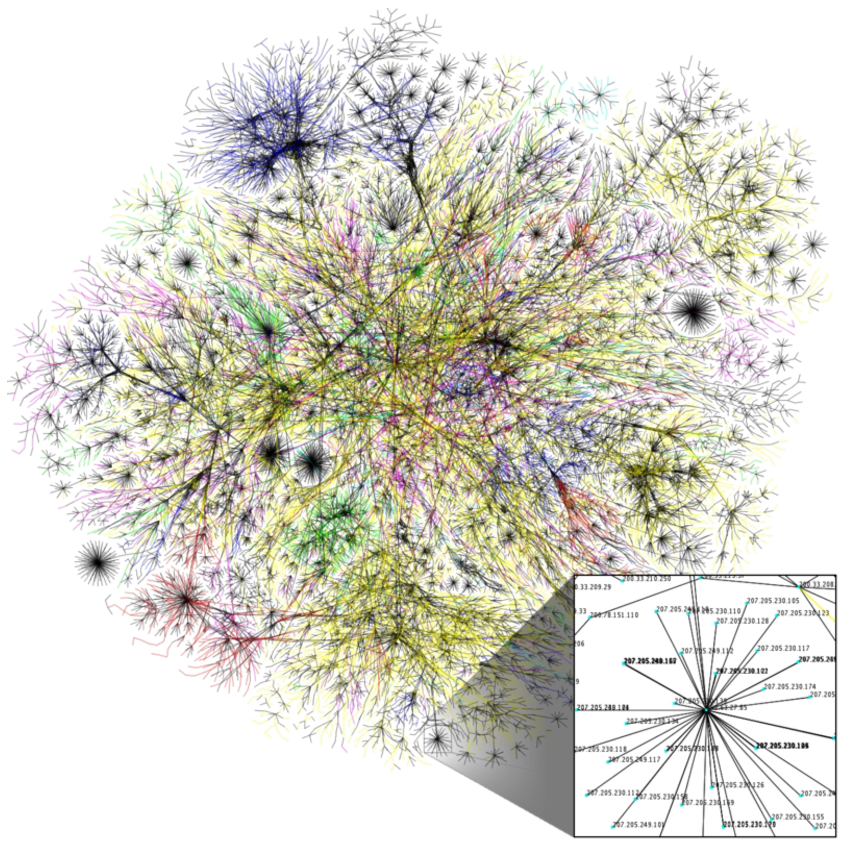


Image retrieved from:

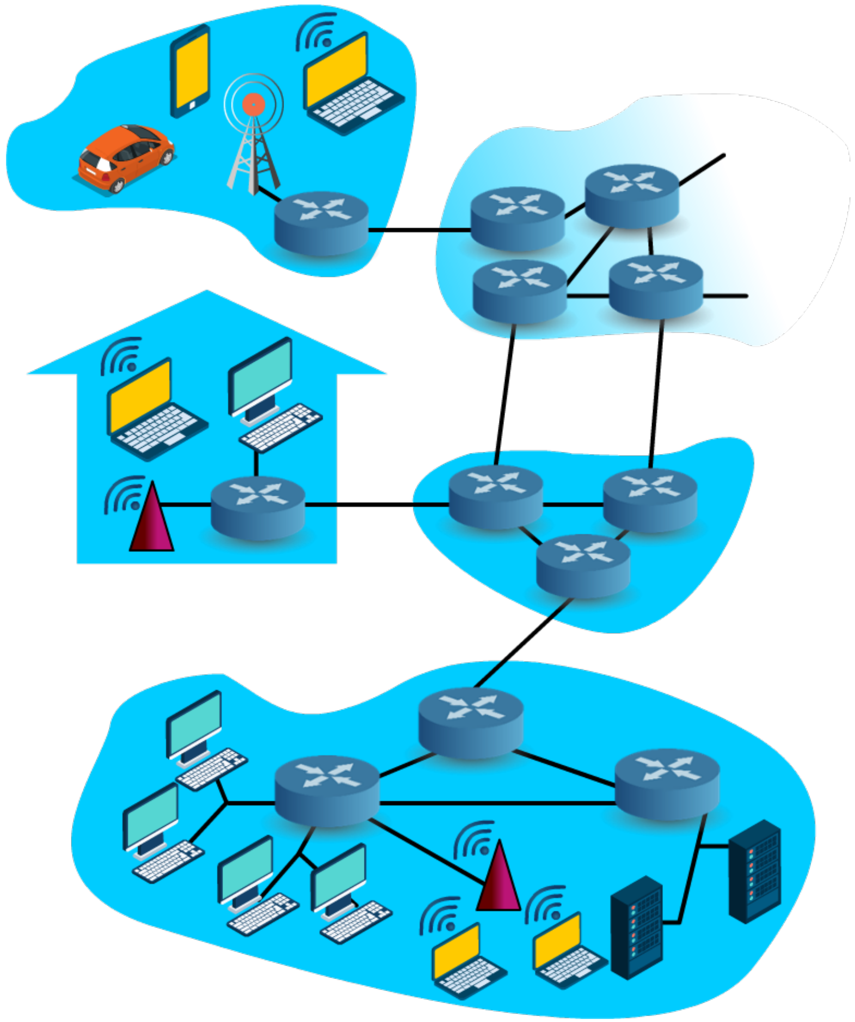
https://en.wikipedia.org/wiki/History_of_the_Internet

Introduction to Computer Network

- What is a computer network?
  A collection of network **nodes**, network **links** and network **protocols**.

- Network offers one **basic service**:
  - Transmission of information.

- Distinguished services:
  - Latency, bandwidth, loss rate, number of end systems, etc.



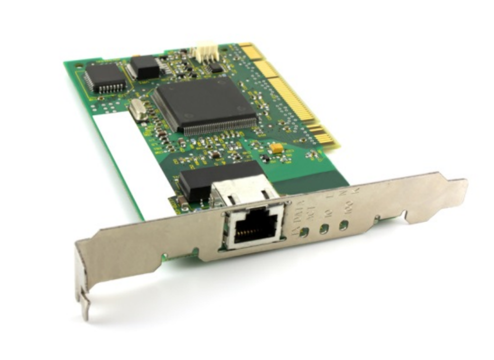
Introduction to Computer Network

- **Communication medium:**
  - o Electronic voltage/current, radio, light.
- **Network components:**
  - o **Nodes:**
    - ➢**Hosts:** Communication endpoints: workstations, laptops, mobile phones, tablets.
    - ➢**Switches/routers:** Nodes used to interconnect links.
  - o **Links:** Carry bits from one place to another (or maybe multiple places): fiber, copper, satellite, …
  - o **Interfaces:** Attach nodes to links.
- **Protocols:** Rules governing communication between nodes.
  - o TCP/IP, ATM, MPLS, SONET, Ethernet, X.25, …
- **Applications:** Web browser, emails, FTP, …

Network Components

Links	Interfaces	Switches/routers

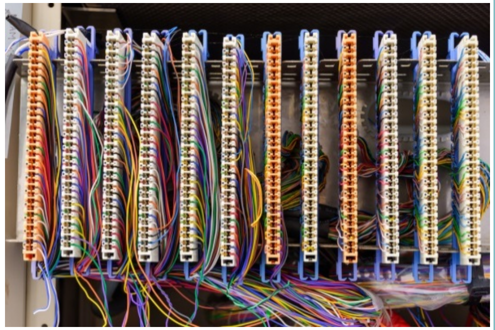

Fibers


Ethernet card


Large routers


Coaxial cable

Wireless card


Telephone switch

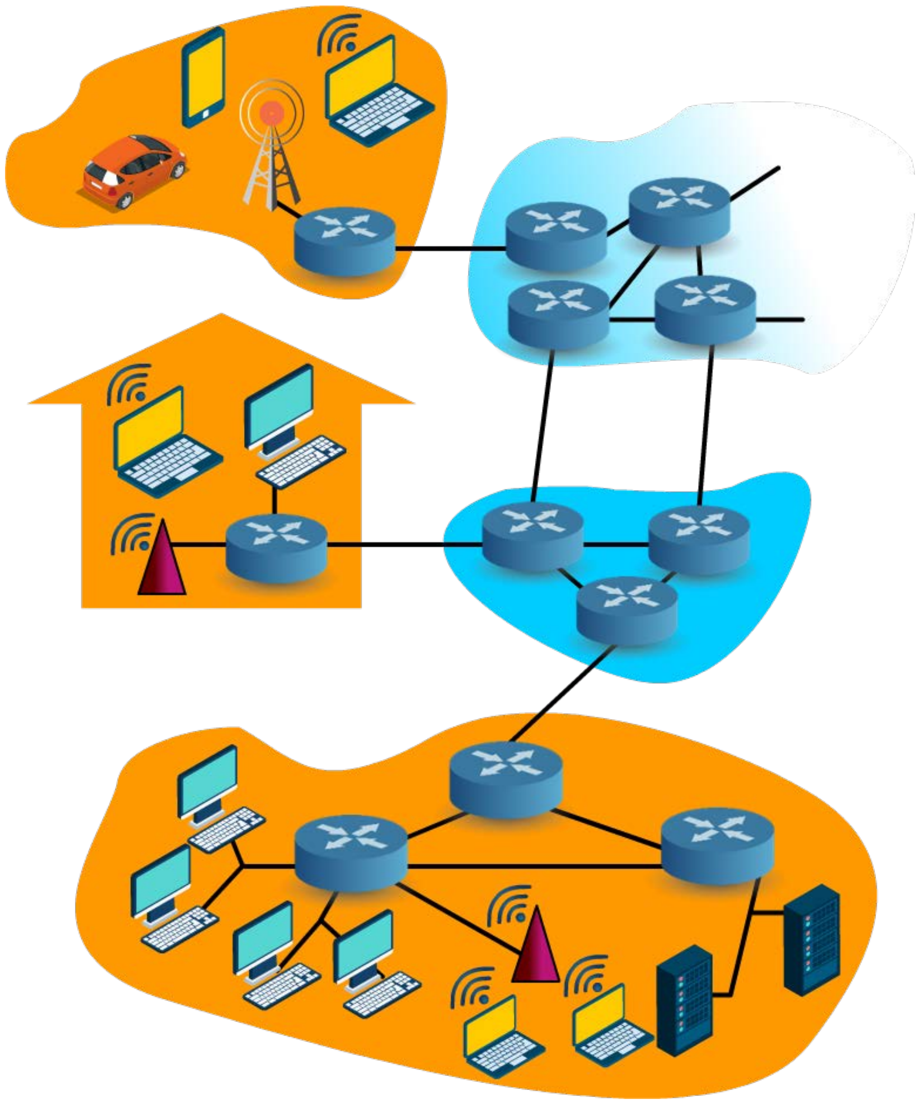
Network Architecture

Network Architecture

- **Network edge:**
  Applications and hosts

- **Access networks:**
  Communication links between ‘edge’ and ‘core’

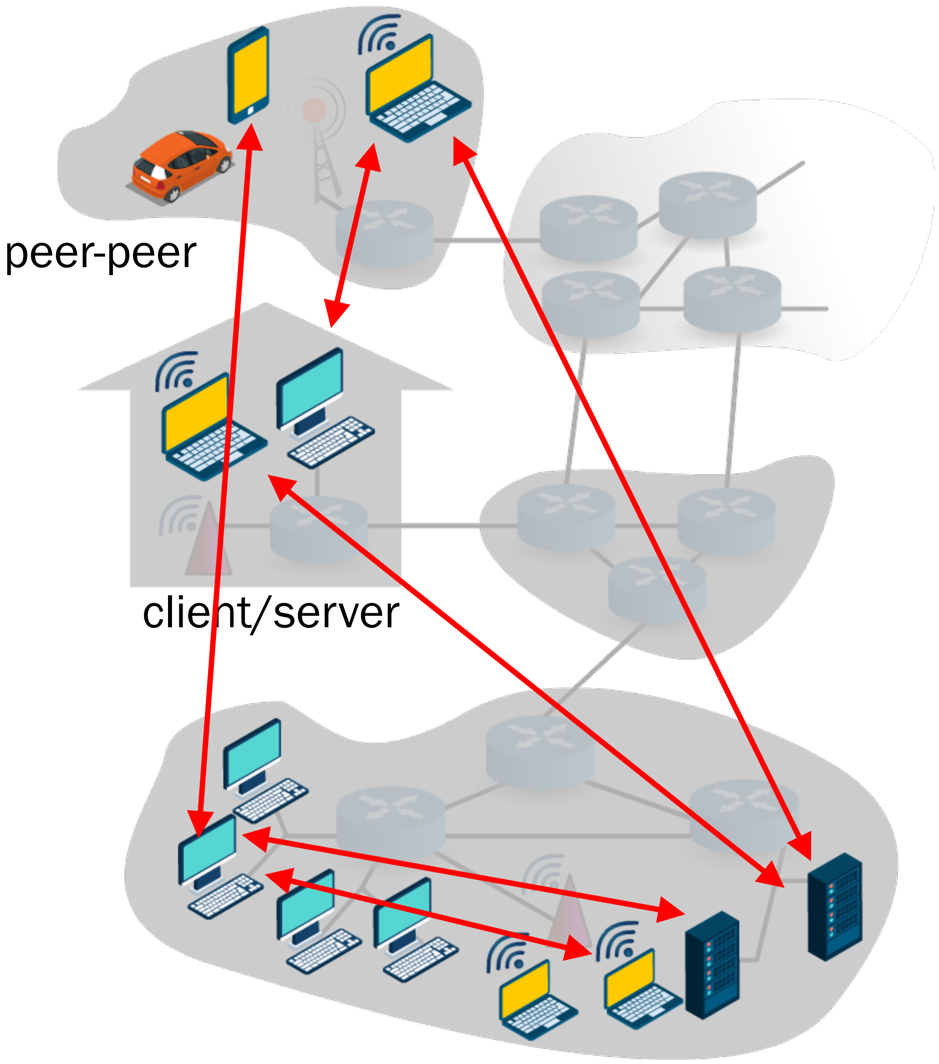
- **Network core:**
  - o Routers
  - o Network of networks



- **End systems (hosts):**
  - Run application programs
  - At "edge of network"

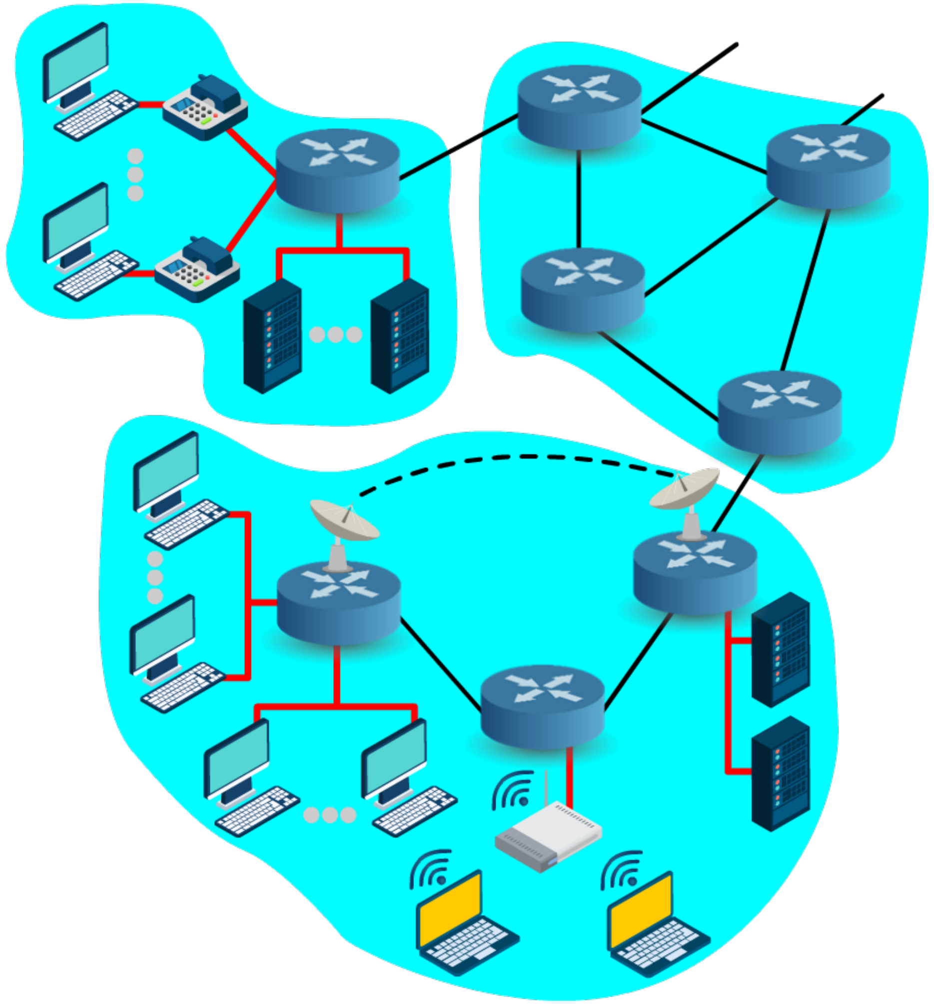
- **Client/server model:**
  - Client hosts requests, receives service from server host.
  - E.g. WWW, email

- **Peer-peer model:**
  - Hosts interact symmetrically, working as both server and client.
  - E.g. BitTorrent



Access networks connect network edges to the network core:

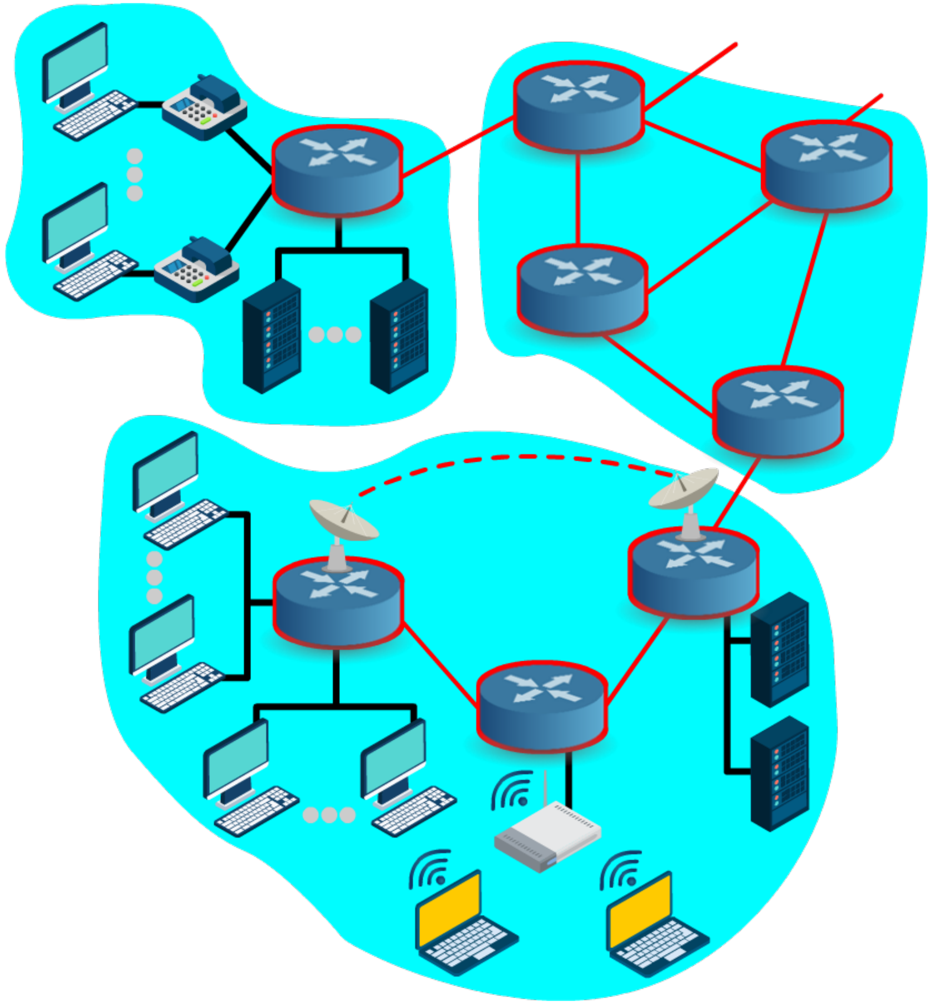
- **Residential access networks:**
  Dial-up modem, ADSL, cable modem, FTTH
- **Institutional access networks (school, company):**
  Ethernet access
- **Mobile access networks:**
  Wireless LAN, wide-area wireless access



- Network core is a mesh of interconnected routers.

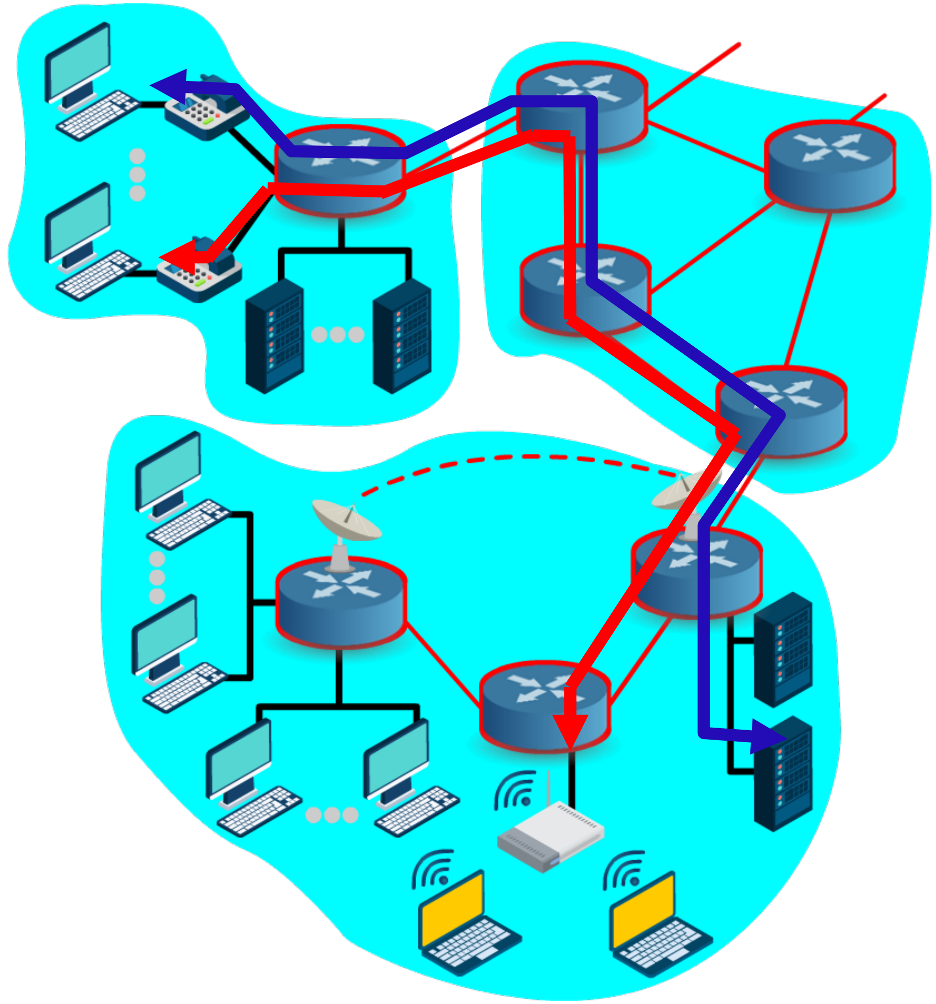
- Types of communication:
  - **Circuit switching**: Dedicated circuit, e.g. telephone network.

  - **Packet (datagram) switching**: Data is sent through network in discrete "chunks".



Circuit Switching vs. Packet Switching

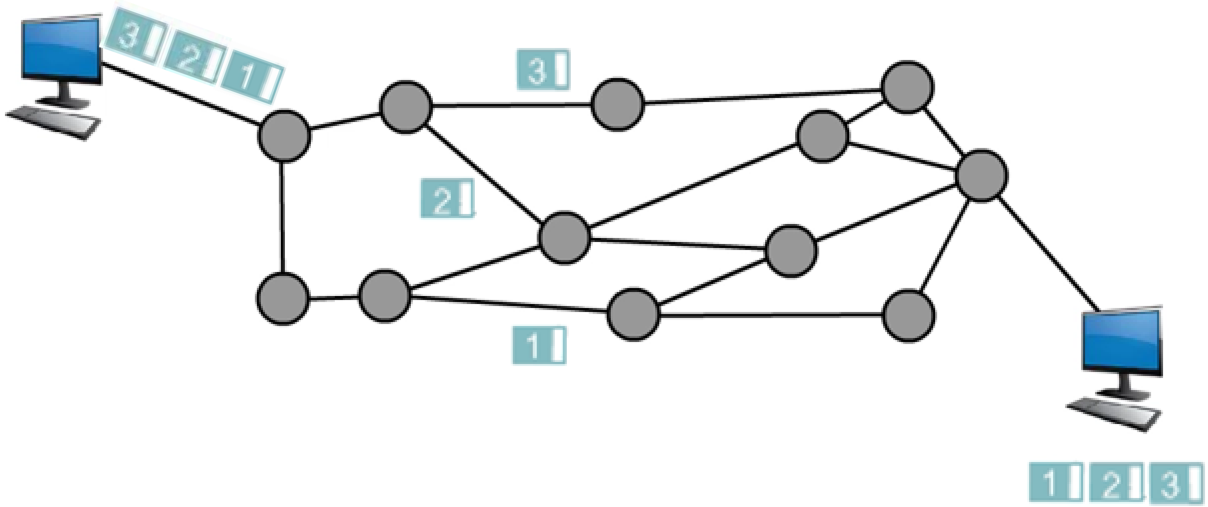
- **Dedicated** communication path between the end nodes:
  - No sharing of resources
- Connection setup required:
  - Must have switching capacity and channel capacity to establish connection
- Limits:
  - Resource wasted if no data transmission
  - Setup/tearing down takes time
- Guaranteed performance
- *E.g. telephone network*



- Network consists of interconnected routers or switches.

- Data transmitted in small packets:
  - Data is organised into packets of multiple of bytes.
  - Each packet contains a portion of user data (Data Payload) and some control information (Header).

- Packets are received, stored briefly (buffered) and passed on to the next node.



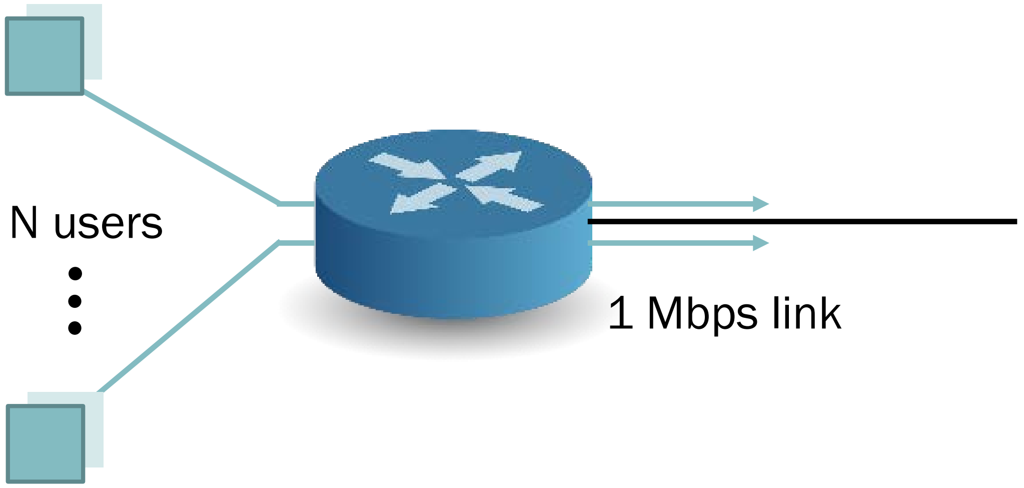
Note that the **Header** is actually "overhead", because what we really want to send is just the **Data**.

Packet Switching vs. Circuit Switching

Packet switching allows more users to share the bandwidth.

Each user:

- 100Kbps when "active".
- Active 10% of the time.



Circuit switching	Packet-switching
1Mbps/100Kbps =10 Support maximum 10 users	with 35 users, probability {**number of active users > 10**} < 0.0004

- Among the 35 users, the probability for *n* users to be active is

$$Pr(n) = \frac{35!}{n! \cdot (35-n)!} \cdot 0.1^n \cdot (1-0.1)^{35-n}$$

- Therefore, the probability for more than 10 users to be active at the same time is

$$\sum_{n=11}^{35} Pr(n) = \sum_{n=11}^{35} \frac{35!}{n! \cdot (35-n)!} \cdot (1-0.1)^{35-n} \cdot 0.1^n \ = 0.0004$$

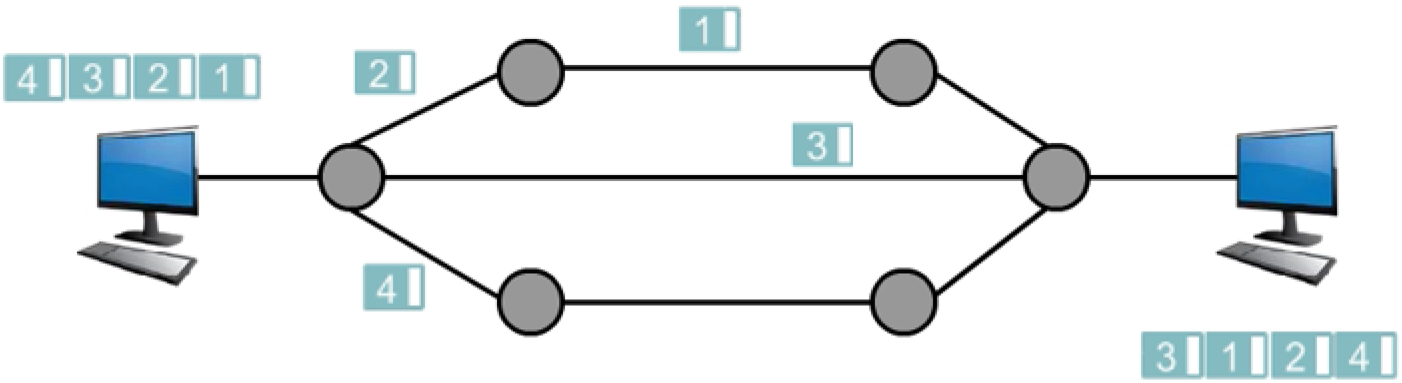
- For a majority of time, a packet-switching network can serve much more users than its capacity actually allows.

- How about the ‘greedy’ case?

Packet Switching Implementation

- Station breaks long message into packets.
- Packets are sent one at a time to the network.
- Packets are handled in two approaches:

Datagram service



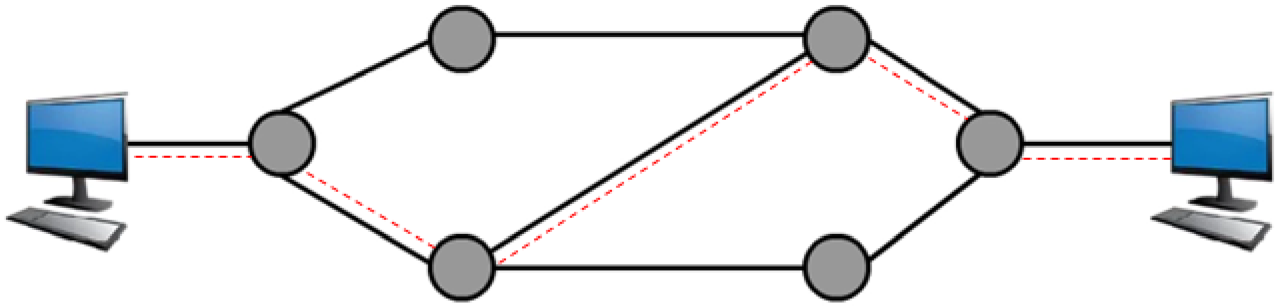
For each packet, each node makes its own decision as to how to forward it so that it eventually reaches its destination.

E.g. IP network.

Packet Switching Implementation

- Station breaks long message into packets.
- Packets are sent one at a time to the network.
- Packets are handled in two approaches:

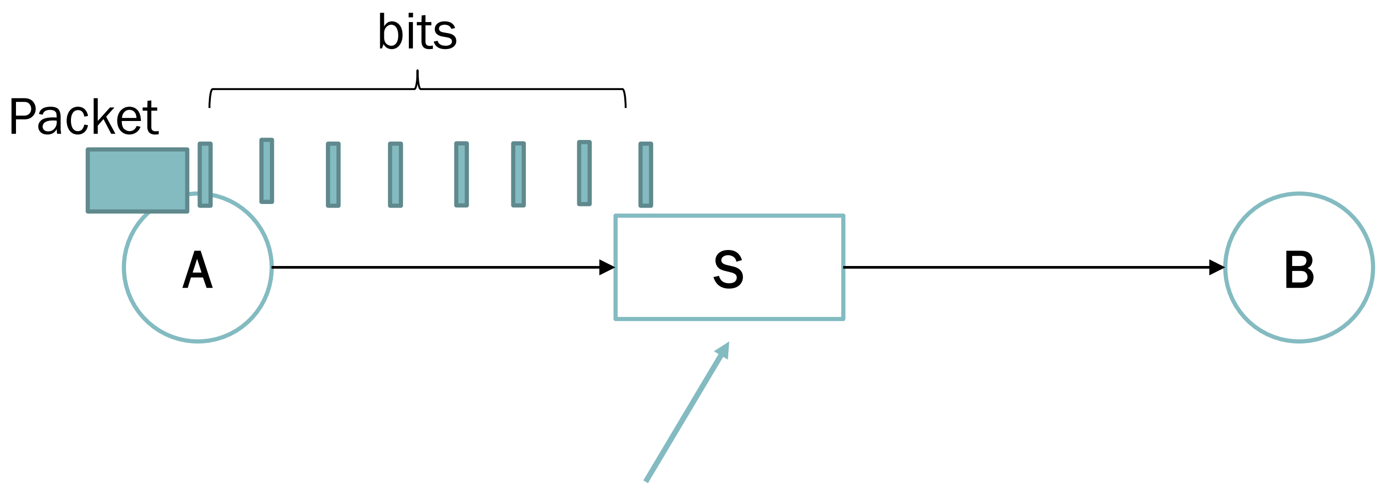
Virtual circuit



A path for the packet is pre-determined for the sender-receiver pair and the packet always follows this path.
E.g. ATM network.

Pipelining in Packet Switching Network

Forwarding Mode



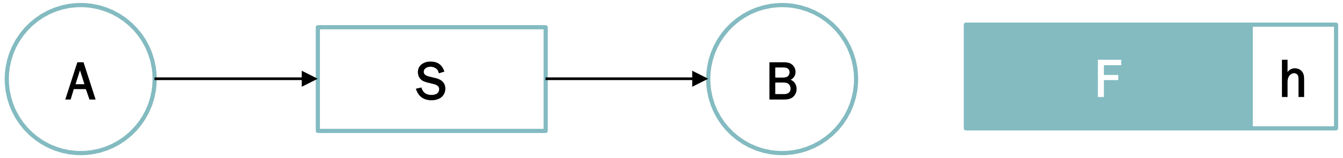
What should switch S do when bits are received?

- **Store-and-forward**: Entire packet must be received completely by the network node before it can be transmitted onto next link.

- **Cut-through**: At the network node, frame is forwarded from input to output port without first being received completely.
    - o Reduced delays, no error checking

Store-and-Forward

Example



F+h bits

R bits/s   R bits/s

$\Rightarrow$ Link speed **R** bits/s (bps)

How long does it take?

Time to transfer message from **A** to **S** $= \dfrac{F+h}{R}$ seconds

Time to transfer message from **S** to **B** $= \dfrac{F+h}{R}$ seconds

Total time (delay) to transfer message from **A** to **B**

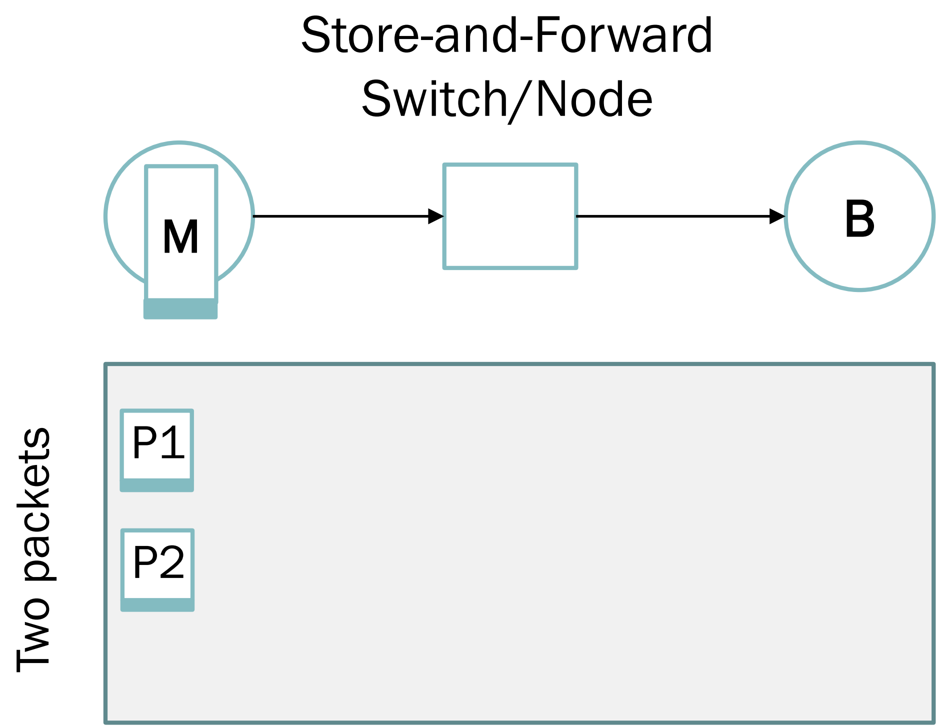
$$= 2\left(\frac{F+h}{R}\right) \text{ seconds}$$

Can we do better than this?

Pipelining

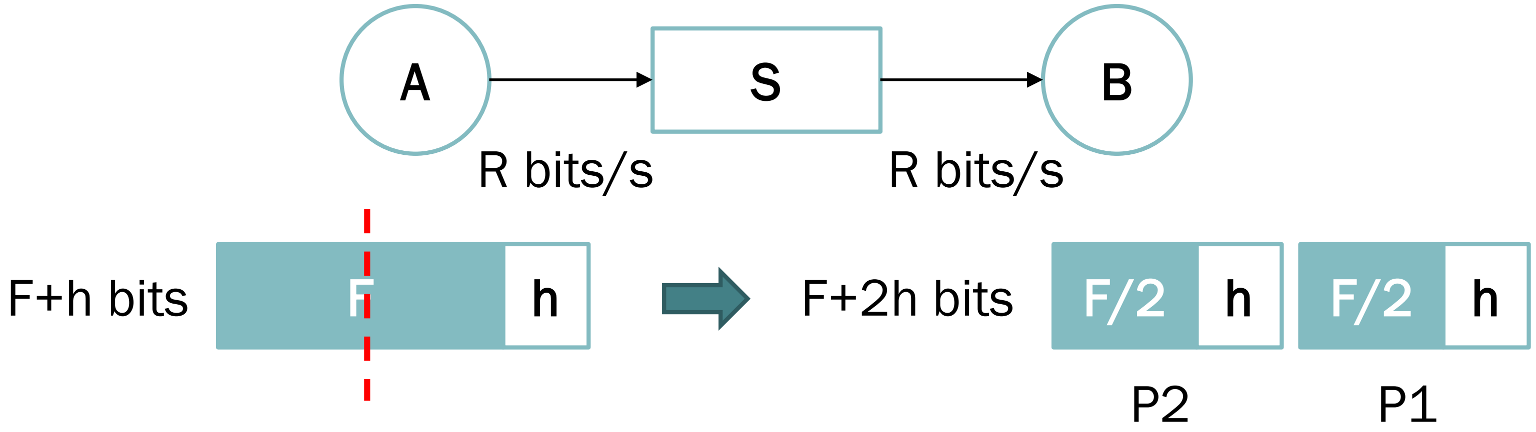
Pipelining packets (of a multi-packet message) may improve **overall delay performance** when packets go through **store-and-forward** nodes in between.

- **A** transmits message (M) of **F** bits as two segments of **F/2** bits each.
- Header of **h** bits attaches to each segment to form packet of **F/2 +h** bits.
- After receiving packet 1 (P1), the **Switch** starts forwarding it to **B** while it receives packet 2 (P2).
- After P2 is received (and P1 is transmitted), the **Switch** starts forwarding P2 to **B**.



Pipelining

Example (cont'd)



P1 from **A** to **S** $= \dfrac{F/2 + h}{R}$ seconds

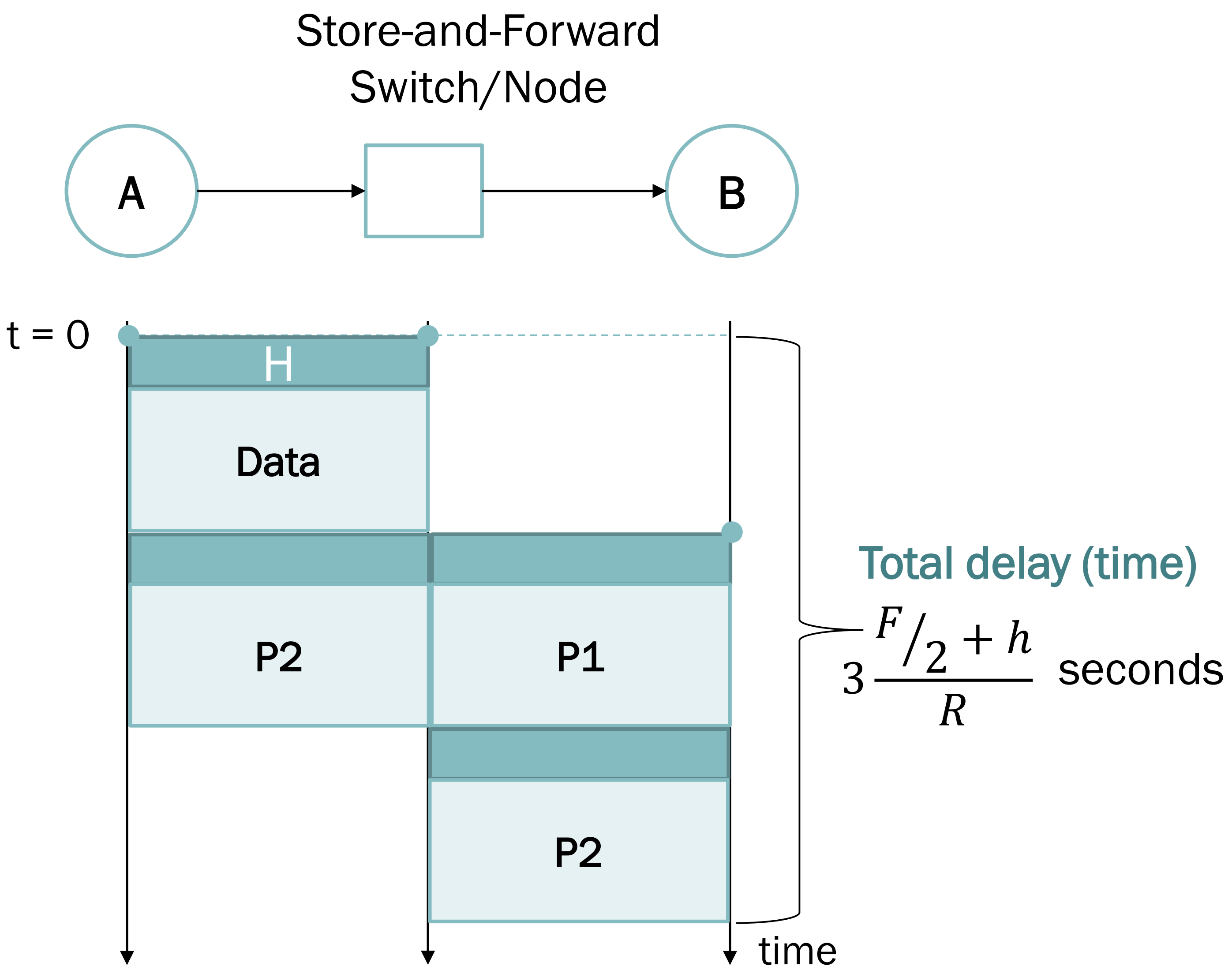
P1 from **S** to **B** and P2 from **A** to **S** $= \dfrac{F/2 + h}{R}$ seconds

P2 from **S** to **B** $= \dfrac{F/2 + h}{R}$ seconds

All the links are working.

Total time $= 3\dfrac{F/2 + h}{R}$ seconds $< 2\left(\dfrac{F + h}{R}\right)$ If F/2>h

Pipelining



Store-and-Forward
Switch/Node
A
B
t = 0
H
Data
P2
P1
P2
time
Total delay (time)
$3\frac{F/_2 + h}{R}$ seconds

Pipelining in Store-and-Forward Transmissions

- When dividing a long message into smaller segments, **header bits have to be added to each segment** to form each packet (i.e. more overhead bits).
- **Pipelining** can then be used to **reduce** the **overall** **message transfer delay**.
- This would be even more effective when there are multiple switches between the source and the destination nodes.
- Avoid making the segments too small as the added overhead of ***h*** **bits per packet** will begin to have a detrimental effect (i.e. higher delays).

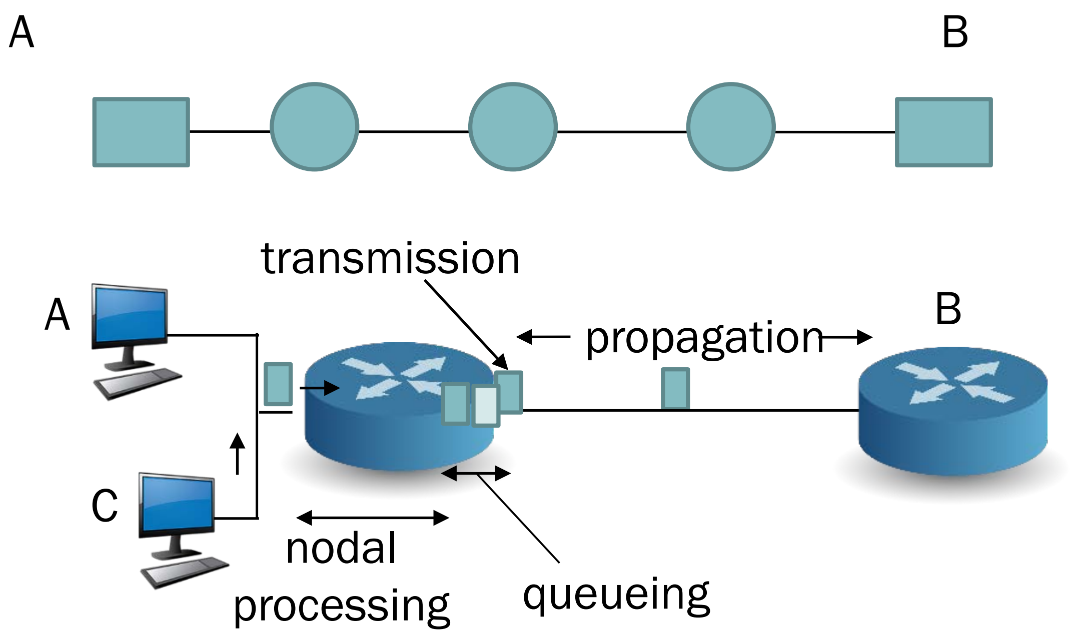
Do Tutorial 1 and Tutorial 2 to get more insight.

Delay in Packet Switching Networks

Packets experience delay on an end-to-end path.
Four types of delay at each hop:

1. Nodal processing delay:
   - Check bit errors
   - Determine output link

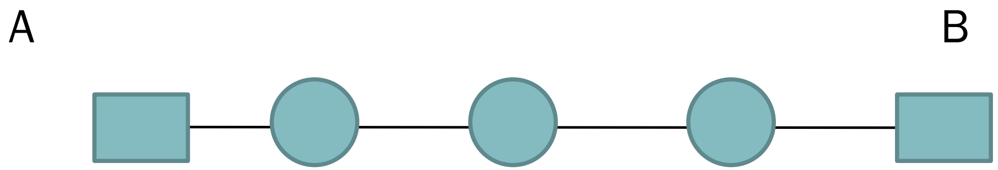
2. Queuing delay:
   - After node processing, packets will join a queue; A packet can be transmitted onto the outbound link only if no other packet is ahead in queue and no other packet is being transmitted on the link
   - Waiting time in queue before being transmitted onto the output link

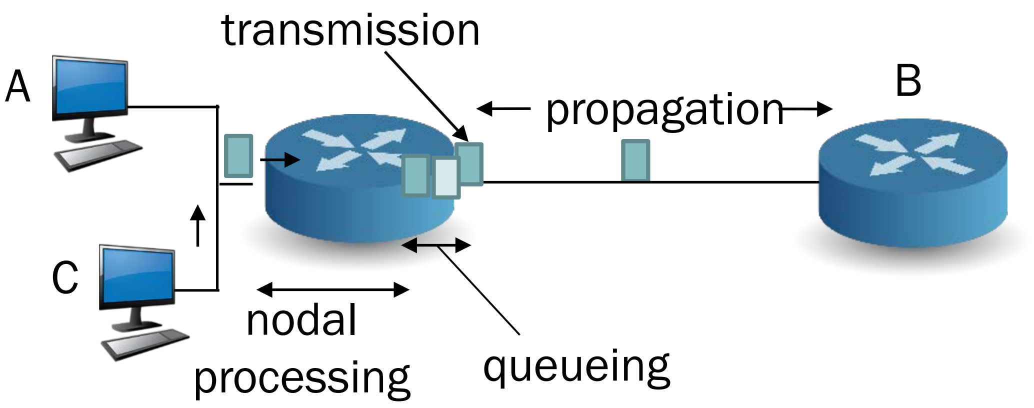


Delay in Packet Switching Networks

Packets experience delay on an end-to-end path.

Four types of delay at each hop:





3. Transmission delay:
   - R = link bandwidth (bps)
   - L = packet length (bits)
   - Transmission delay = time to send bits into link
     = L/R

4. Propagation delay:
   - d = length of physical link (m)
   - s = propagation speed in medium (~$2\text{x}10^8$ m/sec)
   - Propagation delay = d/s

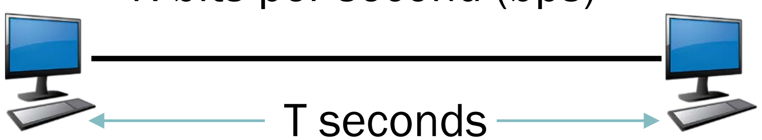
Transmission Delay and Propagation Delay

Packet length: *P* bits

Bandwidth: *R* bps

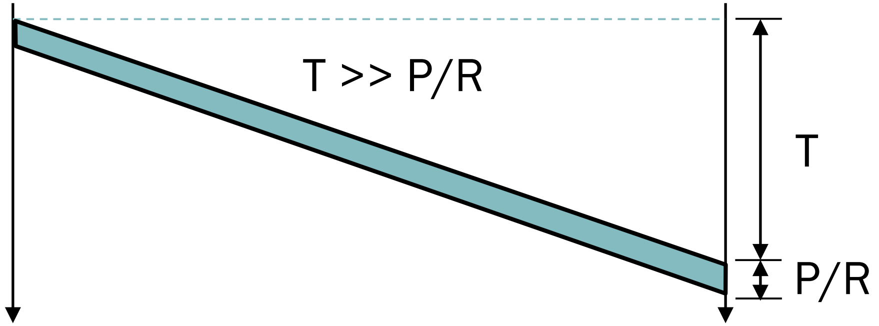
Propagation delay: *T* sec

Propagation speed: 2×10^8 m/sec



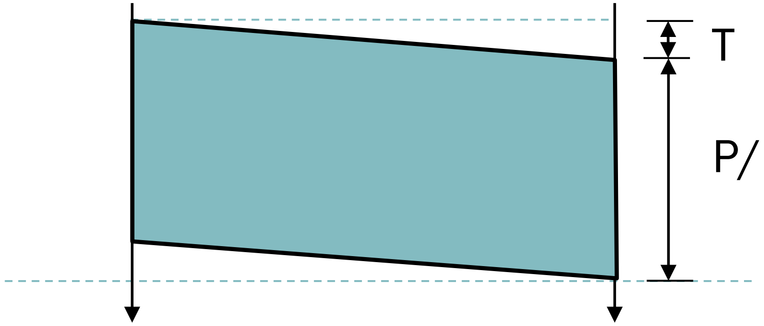
Transmission delay = *P*/*R*

P = 1 Kbyte
R = 1 Gbps
100 Km, fiber =>
T = 500 usec
P/R = 8 usec



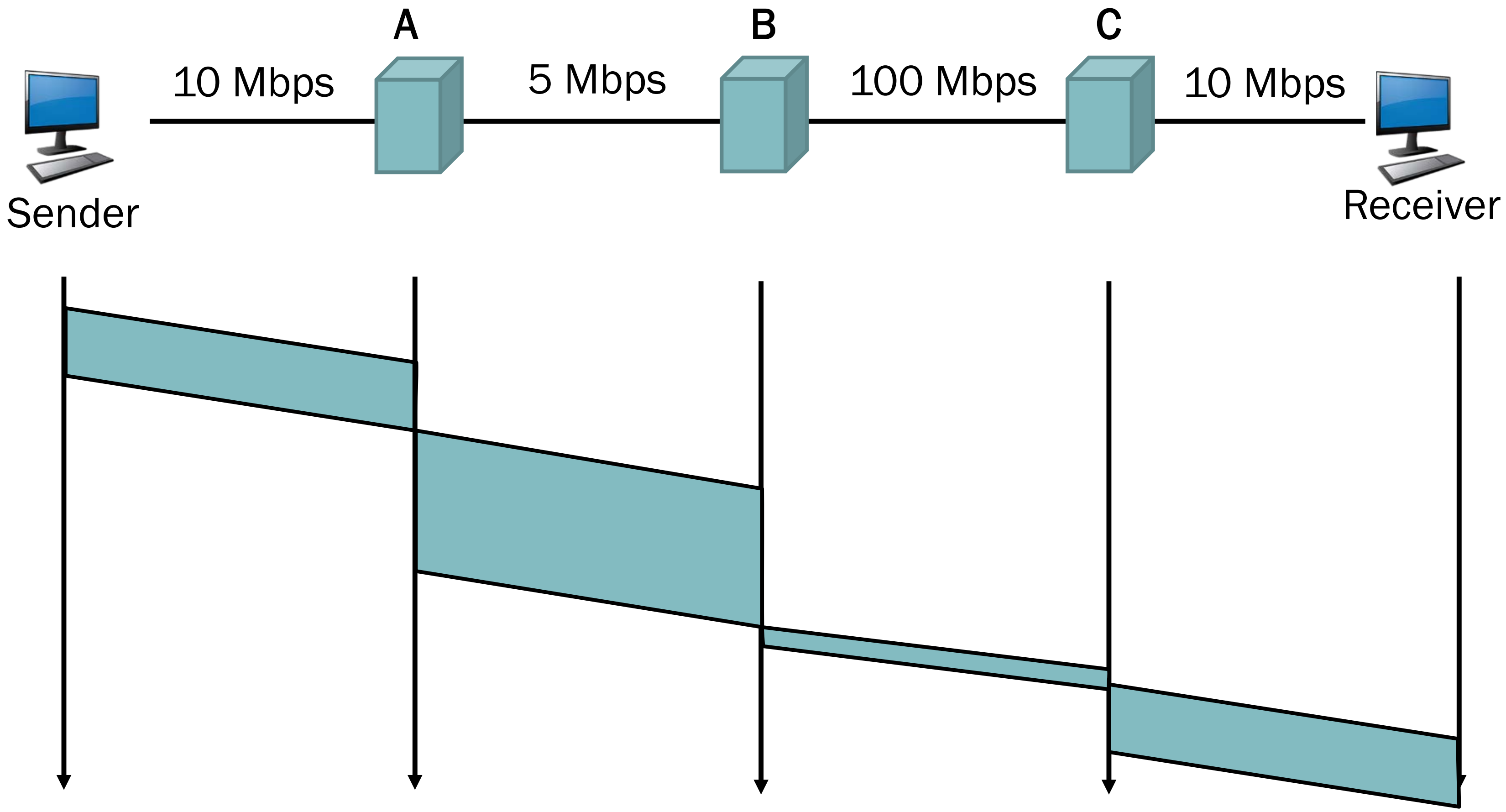
Propagation delay:
T = Length/speed

P = 1 Kbyte
R = 100 Mbps
100 m, copper =>
T = 0.5 usec
P/R = 80 usec



Packet Switching: Store-and-Forward

At a store-and-forward node, a packet must be received completely from the input link before being forwarded onto the output link.



Layered Protocol Architecture

Layered Protocol Architecture

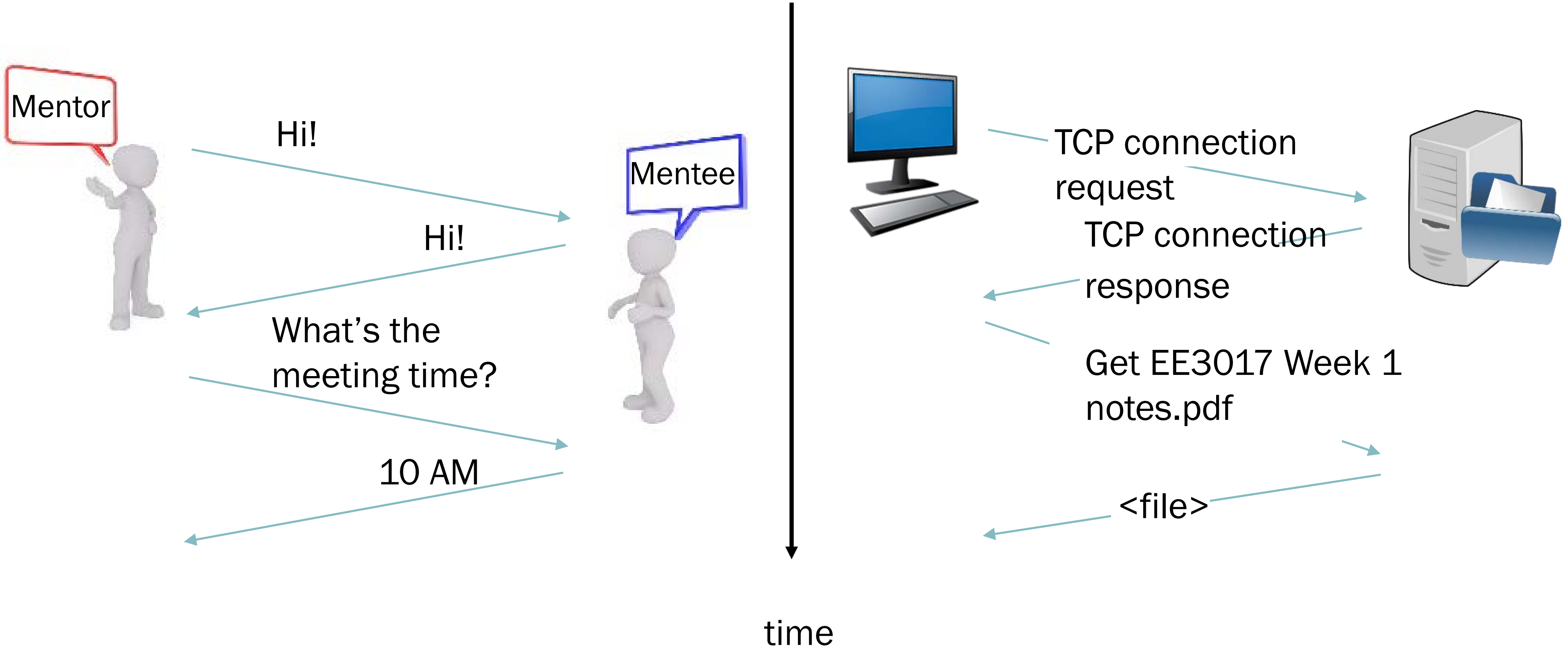
> *To destroy communication completely, there must be no rules in common between transmitter and receiver – neither of alphabet nor of syntax.*

- On Human Communication,

Colin Cherry

What's a Protocol?

A human protocol and a computer network protocol:



Protocol “Layers”

Networks are complex!

Question:
How to organise the structure of the network?

This is where “layering” helps!

Many “pieces”:
- Hosts
- Routers
- Links of various media
- Applications
- Protocols
- Hardware
- Software

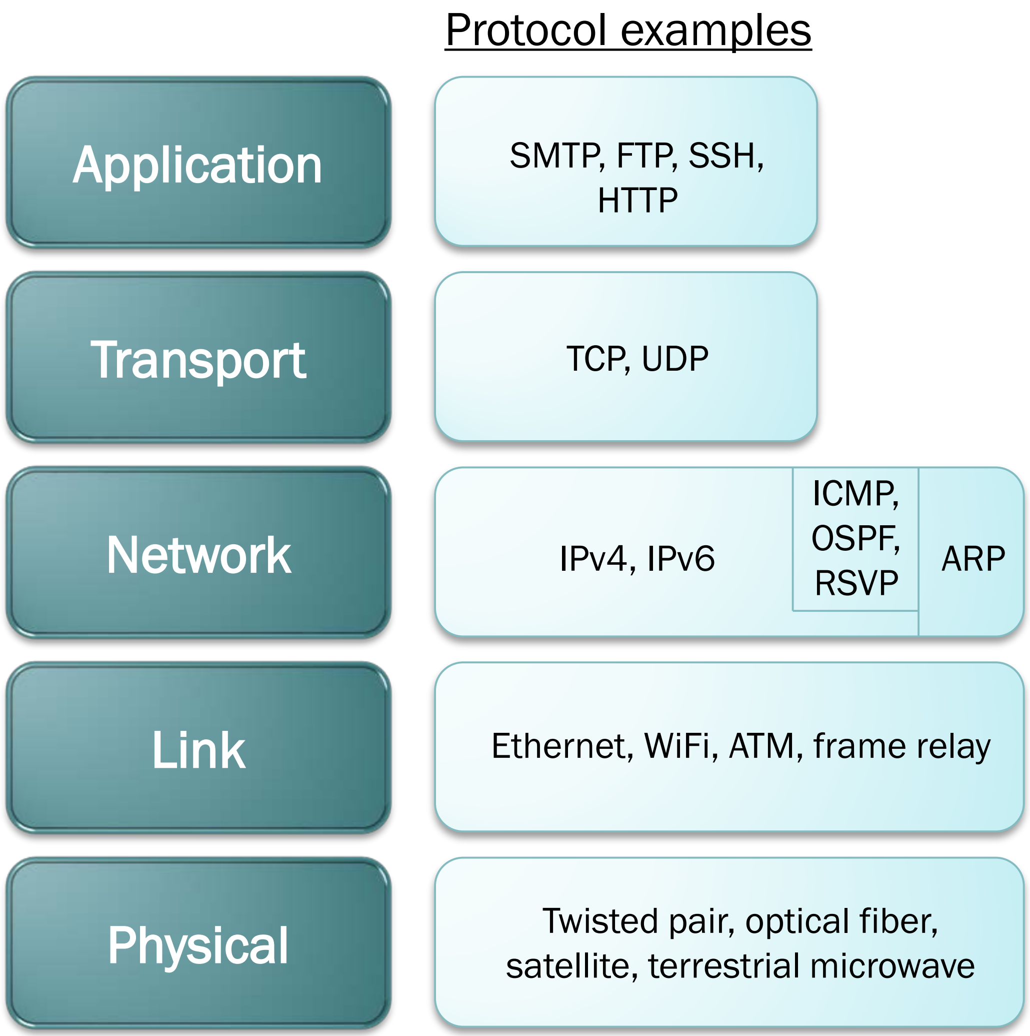
Break into **layers (subtasks)** to:
- “Hide” complexity from the layers above.
- Use services provided by the layers below.
- **Peer layers communicate with a protocol.**
- Modular structure is easier to work with.

Organisation of Air Travel

Departure airport	Intermediate air-traffic control centers	Arrival airport
Ticket (purchase)		Ticket (feedback)
Baggage (check)		Baggage (claim)
Gates (load)		Gates (unload)
Runway takeoff		Runway landing
Airplane routing	Airplane routing	Airplane routing

TCP/IP Protocol Stack

- **Application:** Supporting network applications.
  - *E.g. FTP, SMTP, HTTP*
- **Transport:** Host-host data transfer.
  - *E.g. TCP, UDP*
- **Network:** Routing of datagrams from source to destination.
  - *E.g. IP, routing protocols*
- **Link:** Data transfer between neighbouring network elements.
  - *E.g. Ethernet (802.3), WiFi (802.11)*
- **Physical:** Bits “on the wire”.



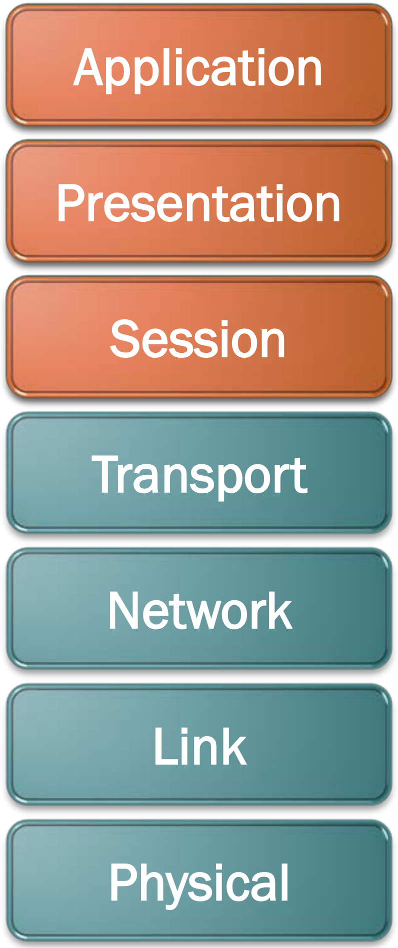
Open System Interconnect (OSI) Model

Supplementary Notes

- **Presentation:** Allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions.

- **Session:** Synchronisation, checkpointing, recovery of data exchange.

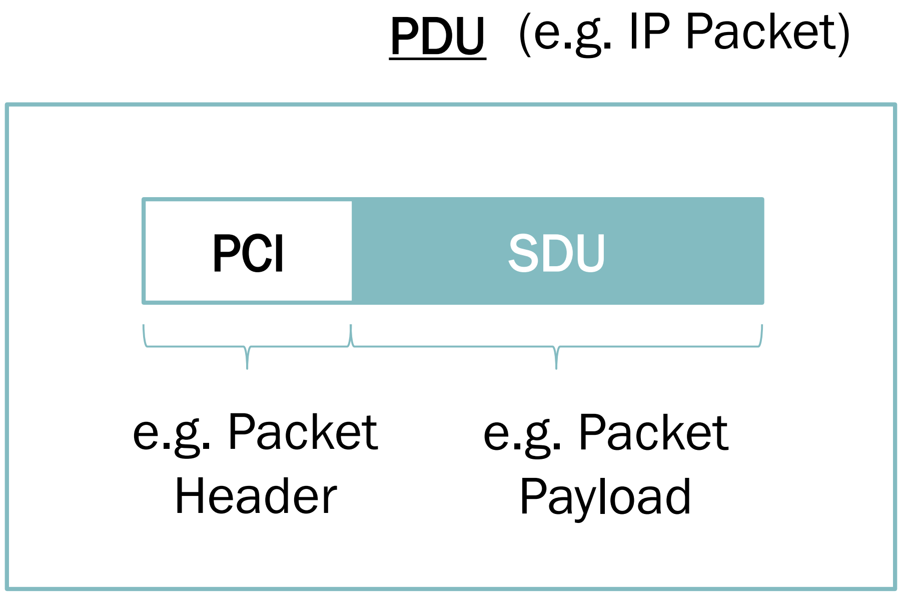
- TCP/IP stack “missing” these layers!
  - o These services, if needed, must be implemented in application.



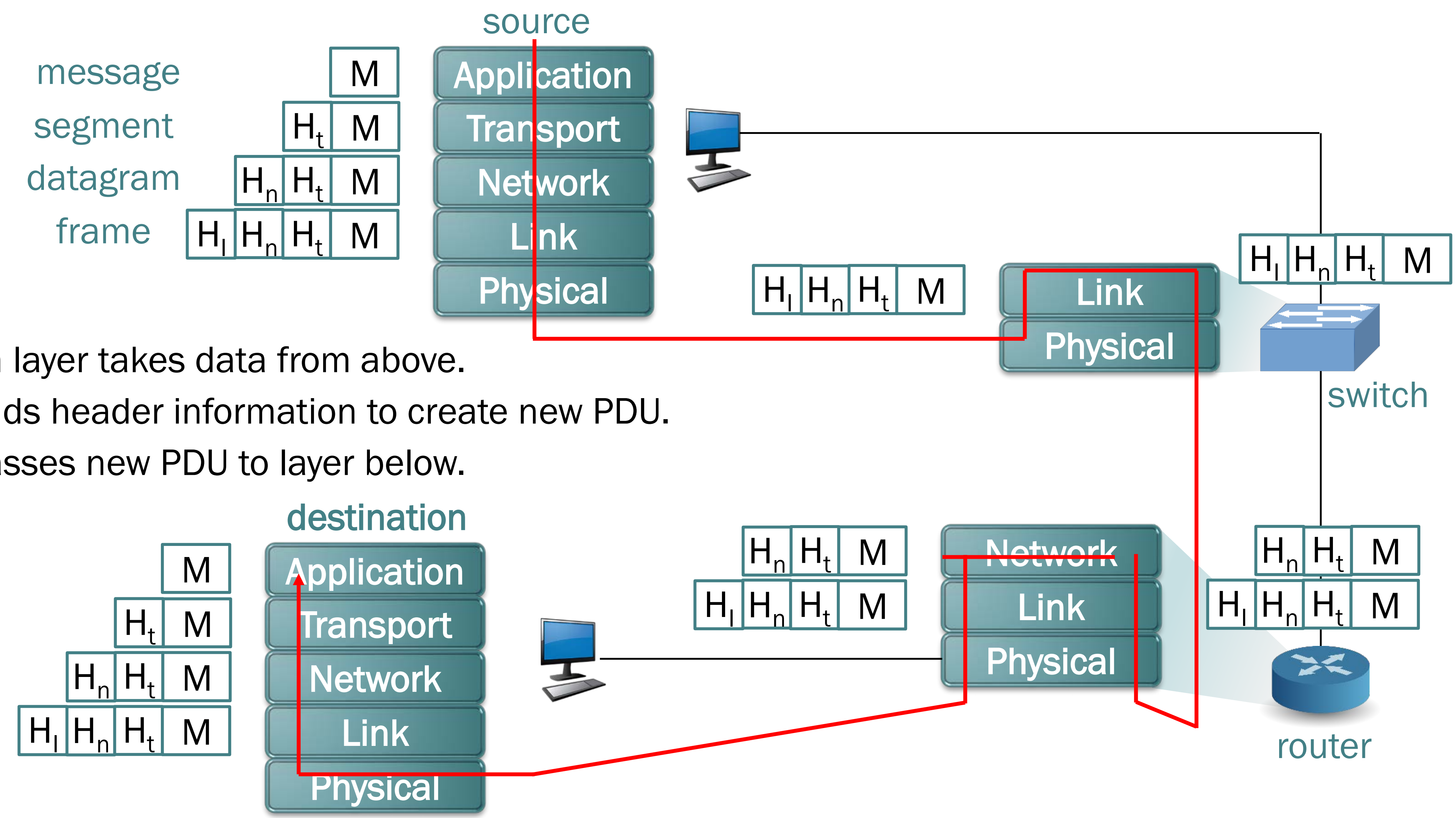
Encapsulation

- Encapsulation is the addition of Protocol Control Information (PCI) or **overhead** to a Service Data Unit (SDU) or **data payload** to form the Protocol Data Unit (PDU) of a particular layer by a communication protocol.

- Example: PDU, i.e. IP packet, of Network layer consists of packet header and packet payload.



Encapsulation



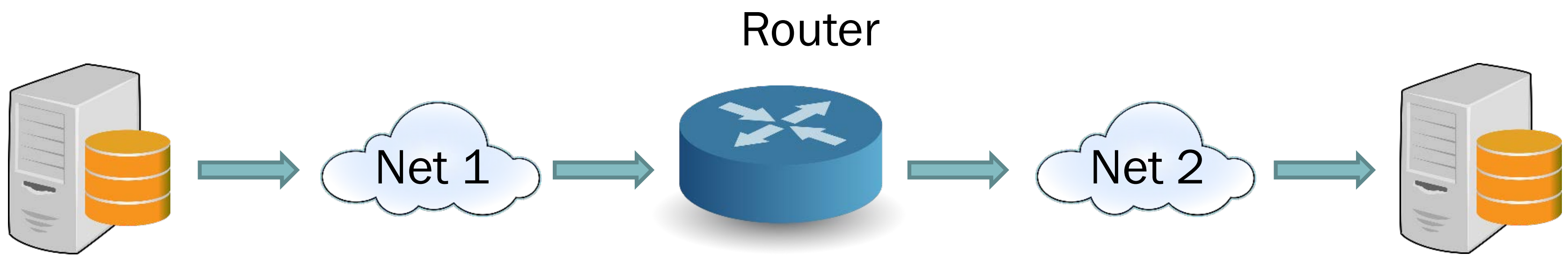
Each layer takes data from above.

- Adds header information to create new PDU.
- Passes new PDU to layer below.

Internet Protocol (IP)

Supplementary Notes

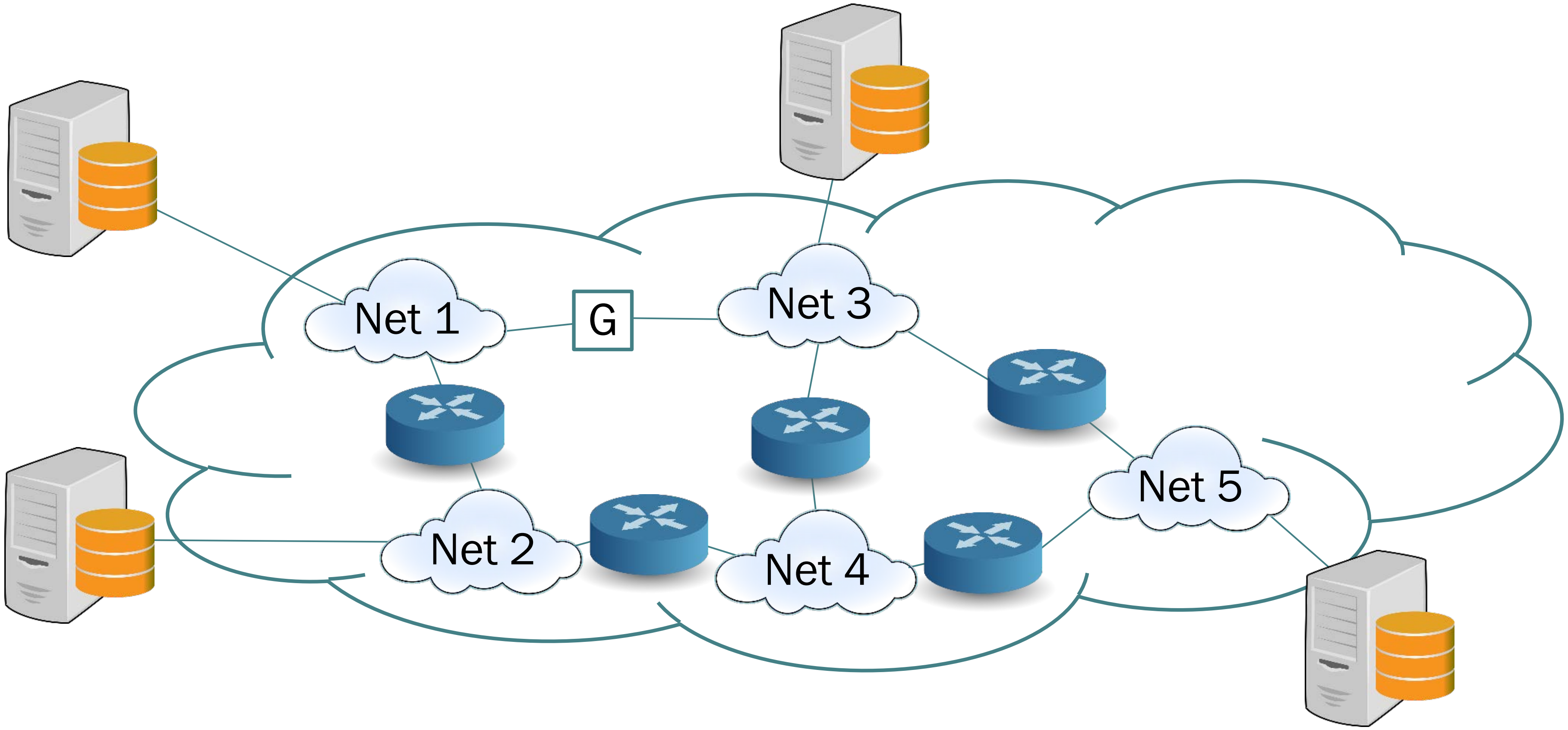
- **This is the protocol most commonly used for the Network Layer.**
- Routers interconnect different networks.
- Host computers prepare IP packets and transmit them over their attached network (complexity at edge).
- Routers forward IP packets across networks.
- **Best-effort** IP transfer service, no retransmission.



Addressing and Routing

Supplementary Notes

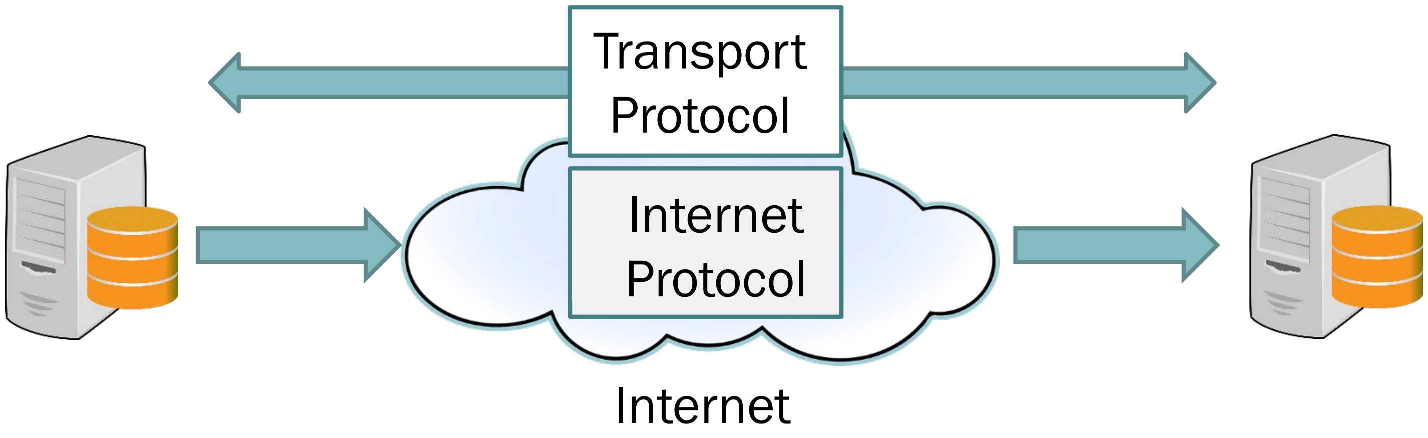
- Hierarchical address: Net ID + Host ID (e.g. 128.100.11.1)
- IP packets routed according to Net ID
- Routers compute routing tables using distributed algorithm



Transport Protocols

Supplementary Notes

- Host computers run two transport protocols on top of IP to enable process-to-process communications.
- **User Datagram Protocol (UDP)** enables **best-effort** transfer of individual blocks of information.
- **Transmission Control Protocol (TCP)** enables **reliable transfer** of a stream of bytes.



Summary

Summary

Key points discussed in this topic:

- The internet was born in 1969 and was called the ARPANET. The development of the TCP/IP protocol suites in the 1970s to 1980s and adoption of TCP/IP as the standard made Internet possible to expand to its size and capability as today.

- A computer network is a collection of network nodes, network links and network protocols.

- The network architecture consists of the network edge, the access networks and the network core.

Summary

Key points discussed in this topic (cont'd):

- The two types of communication are:
  - Circuit switching: Dedicated communication path between the end nodes.
  - Packet switching: Network consists of interconnected routers or switches.
    - Packets are handled in two approaches: Datagram service and Virtual circuit.

- In packet switching network, the four types of delay at each hop are:
  - Nodal processing delay
  - Queuing delay
  - Transmission delay
  - Propagation delay

- Pipelining packets can be used to improve the overall delay performance when packets go through store-and-forward nodes in between.

Summary

Key points discussed in this topic (cont'd):

- Protocol layers help to organise the structure of the network as networks can be complex.

- The TCP/IP protocol stack consists of the following:
  - Application
  - Transport
  - Network
  - Link
  - Physical

- Encapsulation is the addition of PCI or overhead to SDU or data payload to form the PDU of a particular layer by a communication protocol.