

# CSC4200/5200 – COMPUTER NETWORKING

## NETWORK FUNDAMENTALS

Instructor: Susmit Shannigrahi  
[sshannigrahi@tnitech.edu](mailto:sshannigrahi@tnitech.edu)

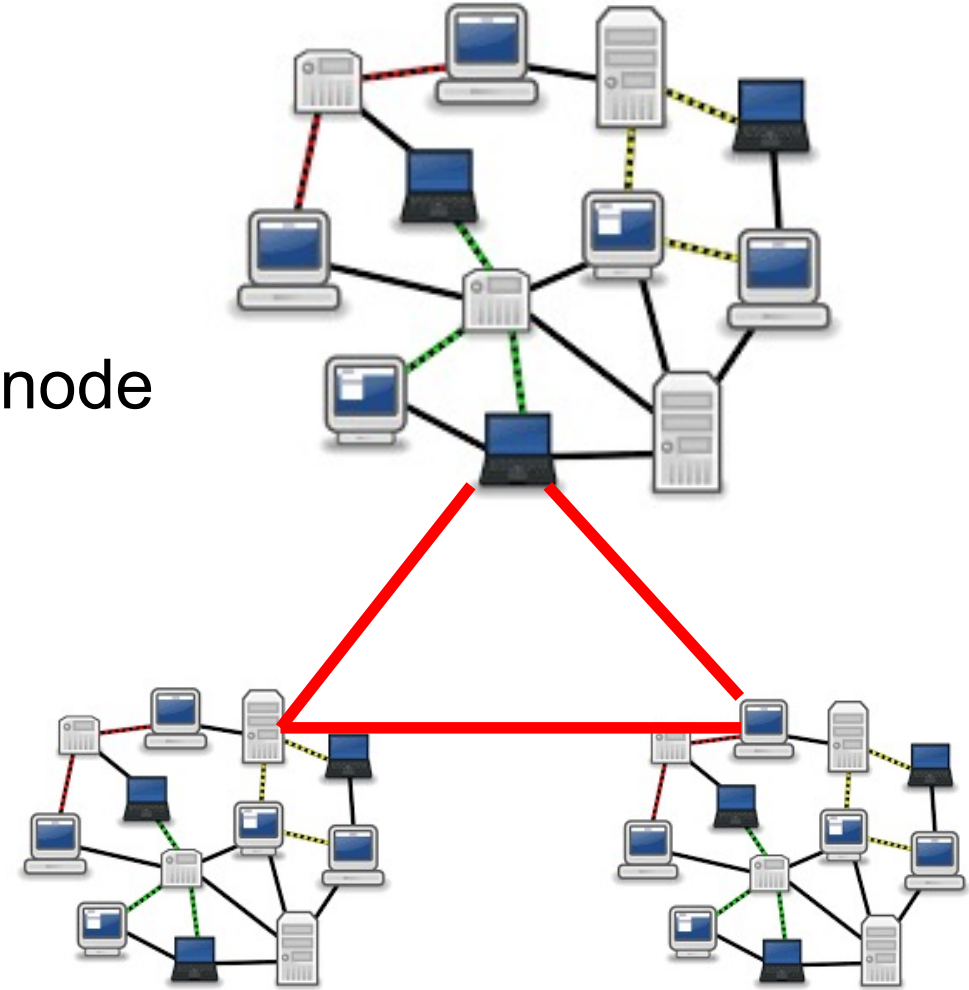
# Chapter 1: Fundamentals

---

- Networking is ubiquitous
  - What did you use it for today?
- First things first:
  - Terminology
  - Basic tools
  - What does it take to build an Internet?

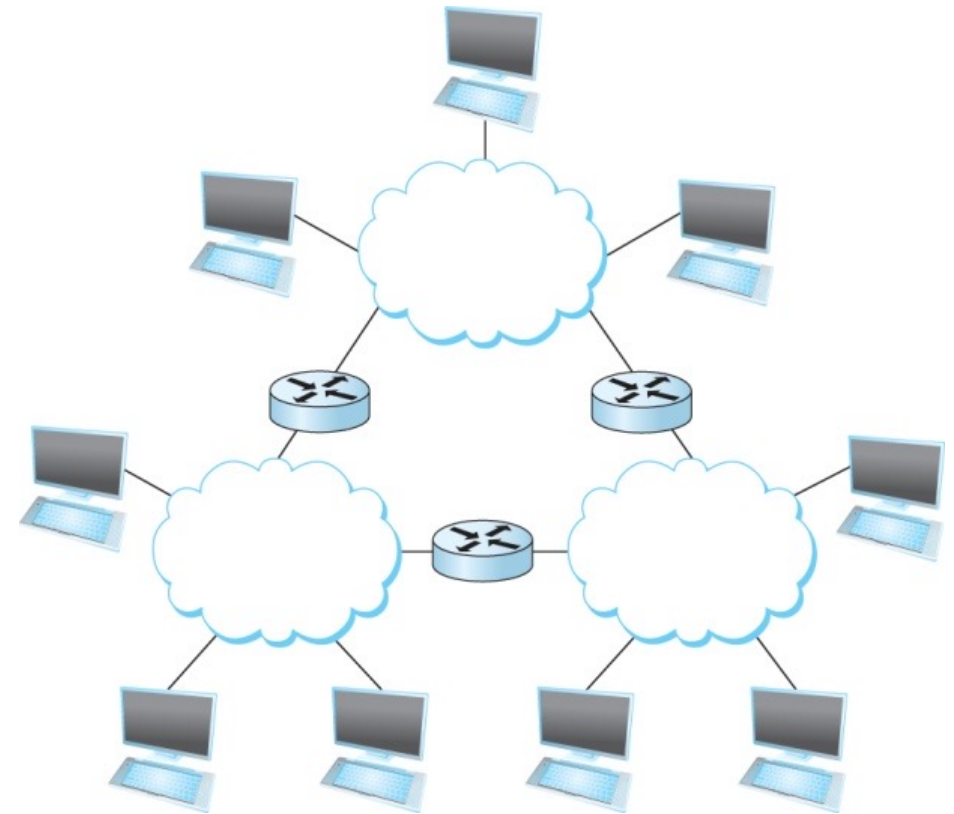
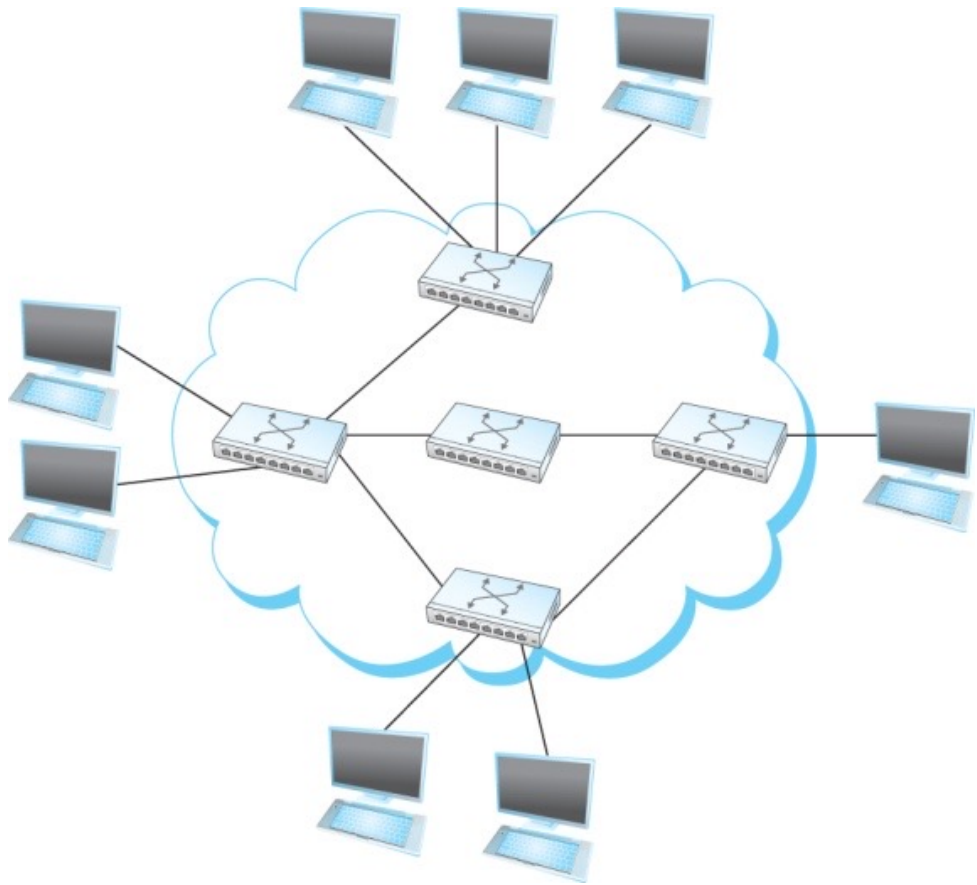
# Links, Nodes, Network, Internet

- You can view the network as a graph
- Each device (a phone, a computer) is a node
- Each connection is a link
  - Wires = real links
  - Bluetooth, Radio, Infrared = virtual links
- Nodes + links = a network
  - Many connected networks = Internet



# A Network and the Internet

---



# Links, Nodes, Routers, Switches

---



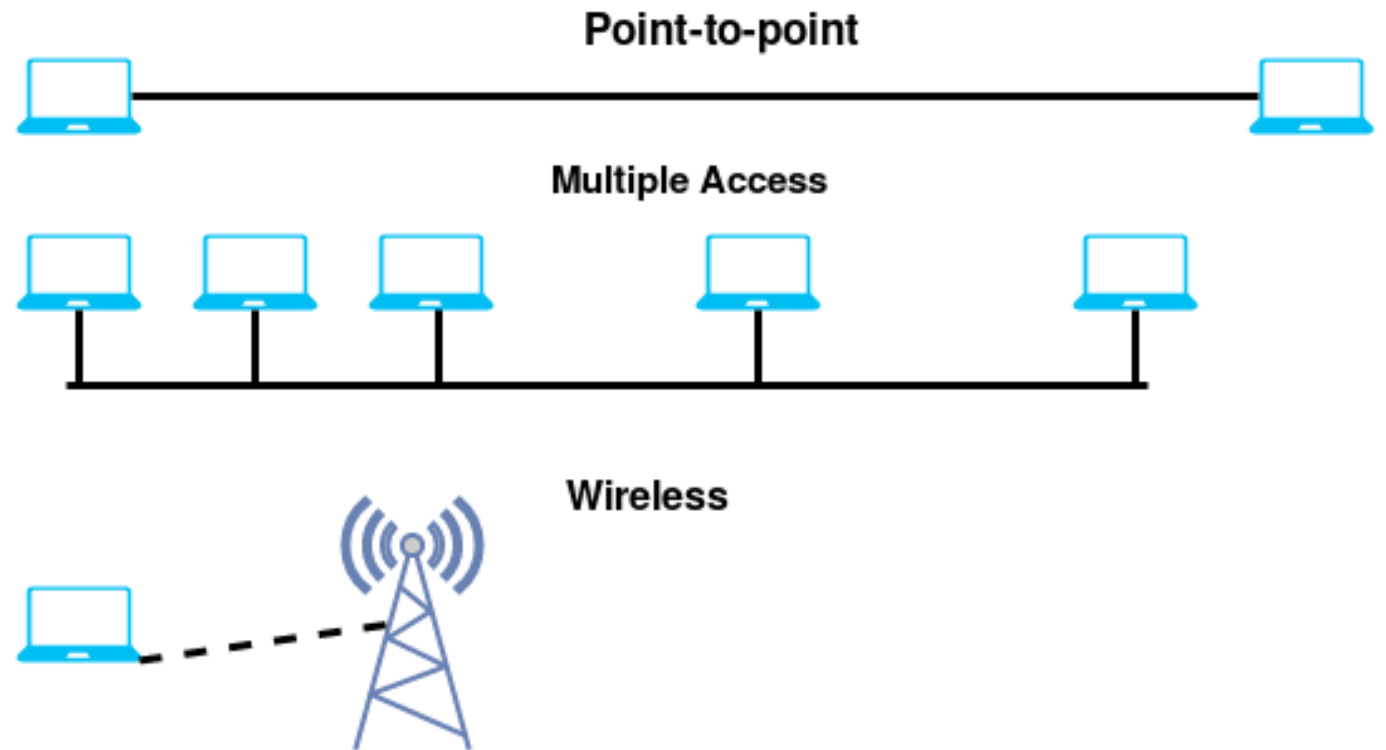
# Client and Server

---

- My laptop with a browser = client
  - It requests a service
  - Email, chat, video, youtube
- A node running a program that serves the requests = server
  - Runs a service
  - Chat, video, messaging
- A node can both be a client and a server

# Connectivity

- Point-to-Point
- Multiple access
- Wireless

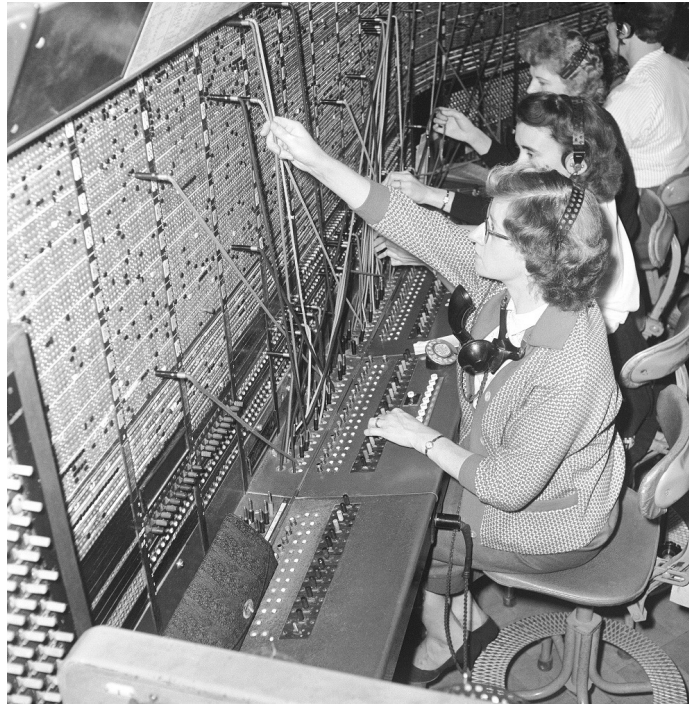


# Circuit Switching – Old telephone networks

---



Operator, get me  
the navy



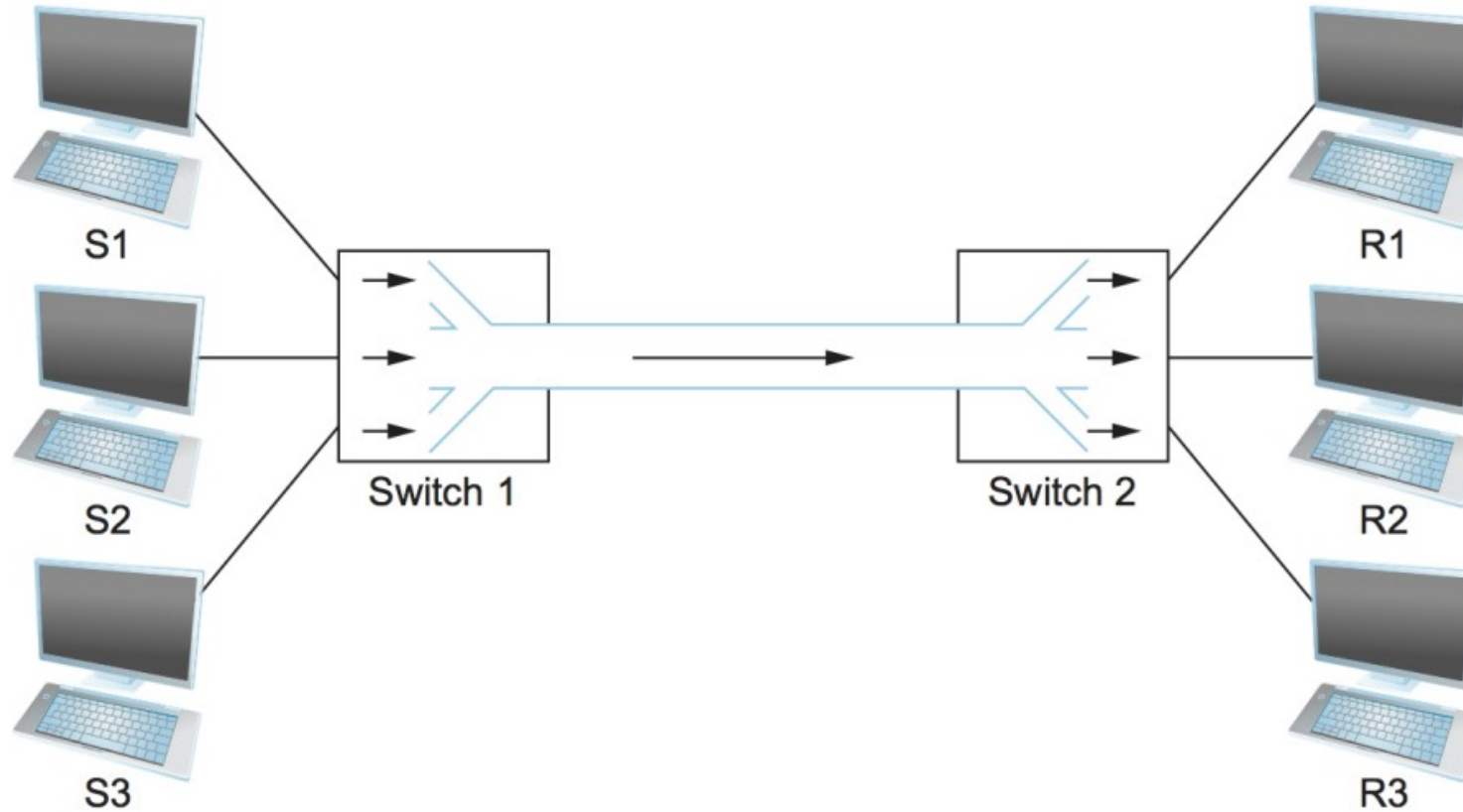
- Build physical wire:
  - Guaranteed resources

- **Why change a working system?**



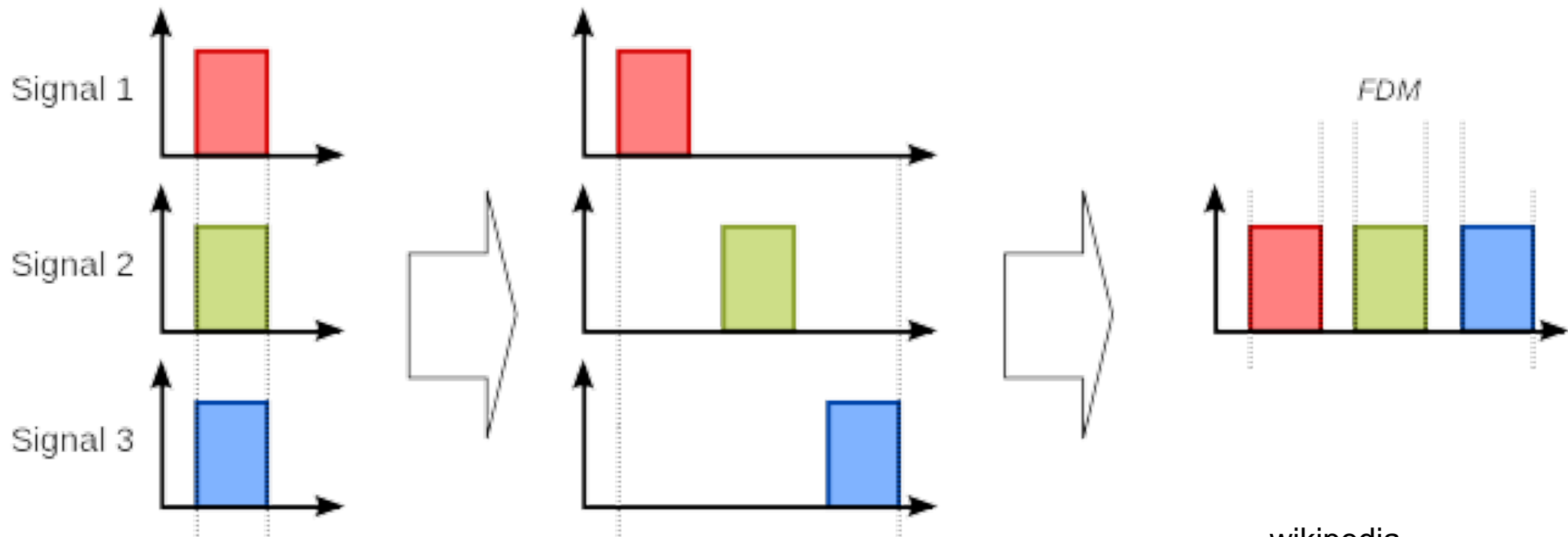
# Circuit Switching

---



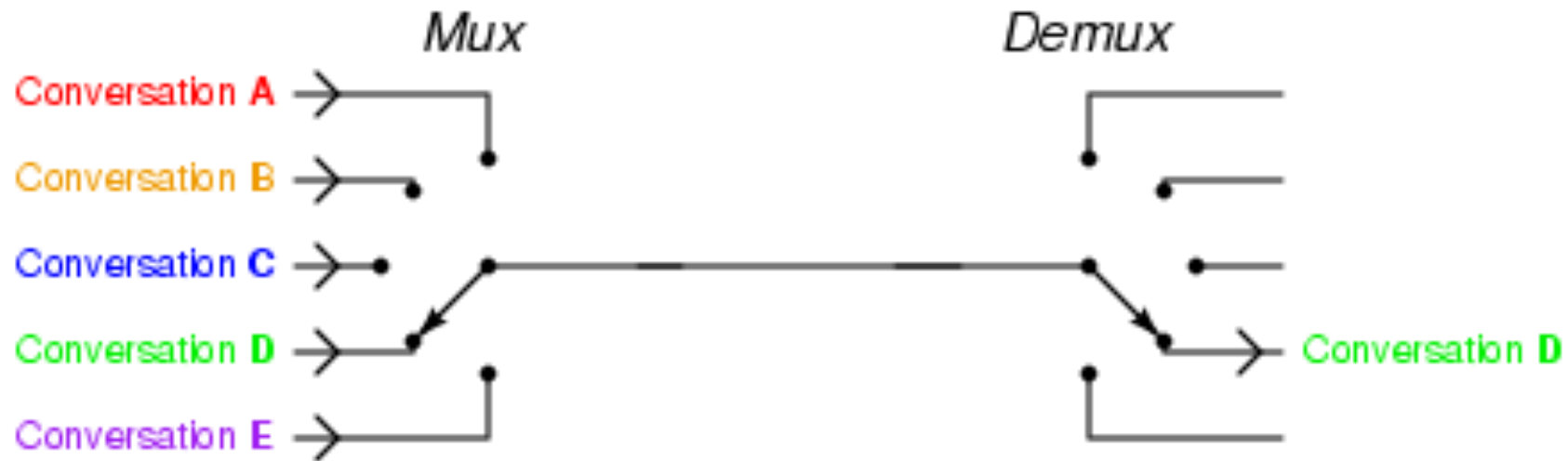
- **What are the problems?**

# Frequency Division Multiplexing for Circuit Switching



wikipedia

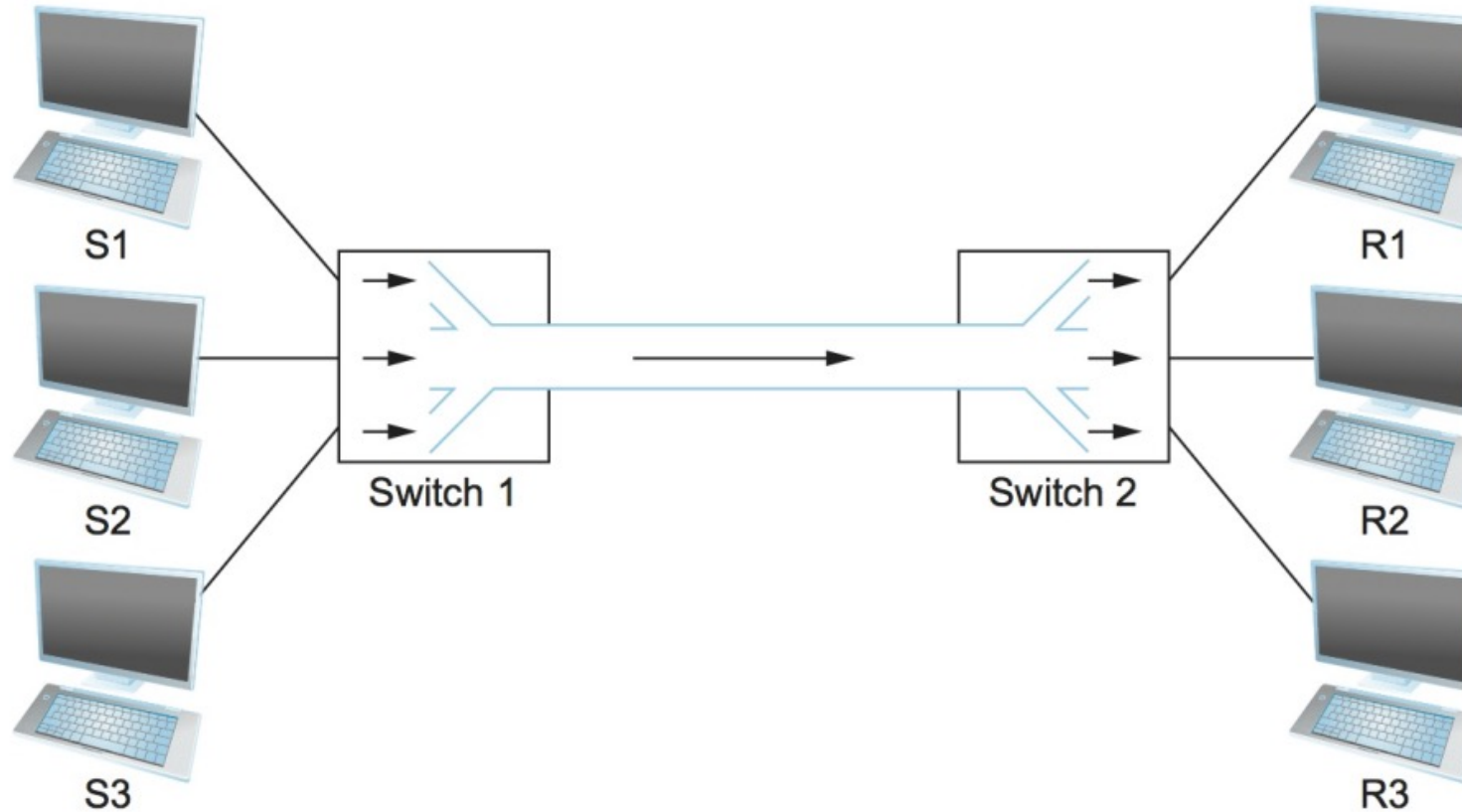
# Time Division Multiplexing for Circuit Switching



wikipedi  
a

# Circuit Switching – TDM and FDM

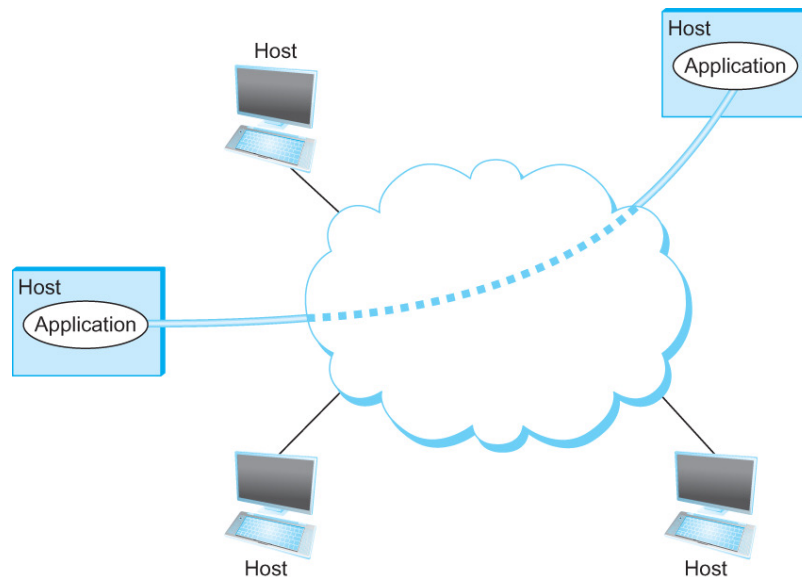
---



- **Problems solved? Or do they still exist?**

# Support for Common Services

- Logical Channels
  - Application-to-Application communication path or a pipe



Process communicating over an  
abstract channel

# Packet Switching

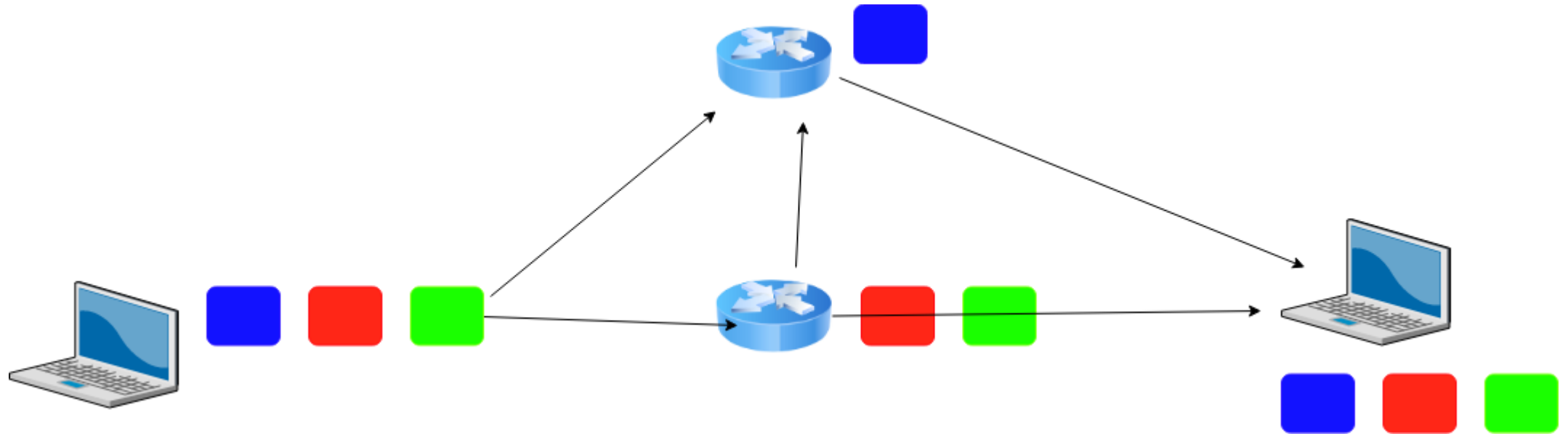
---



- Packets are low level components
- Multiple kind of traffic with different requirements
  - Gaming vs Phone
- Dumb network – How do you ensure quality of service?
- End points must be smart

# Packet Switching

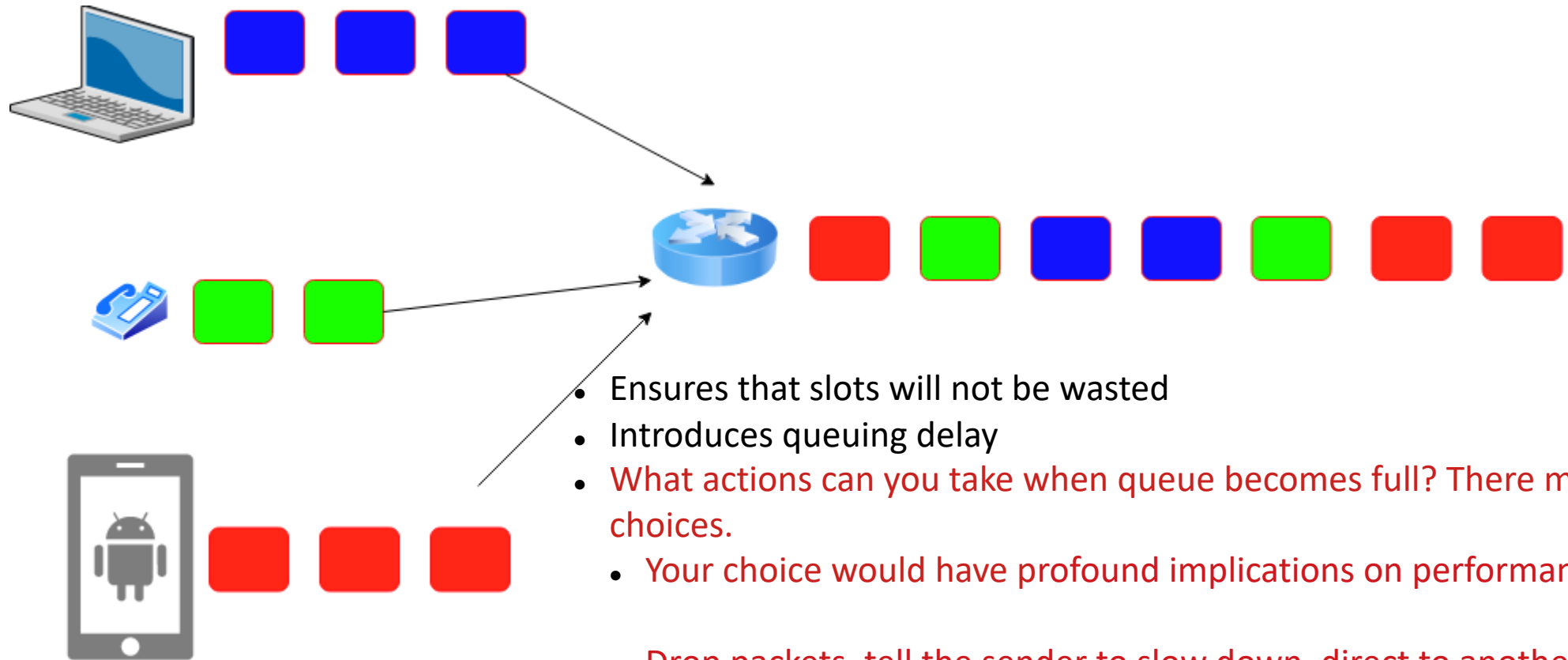
---



Analogy?

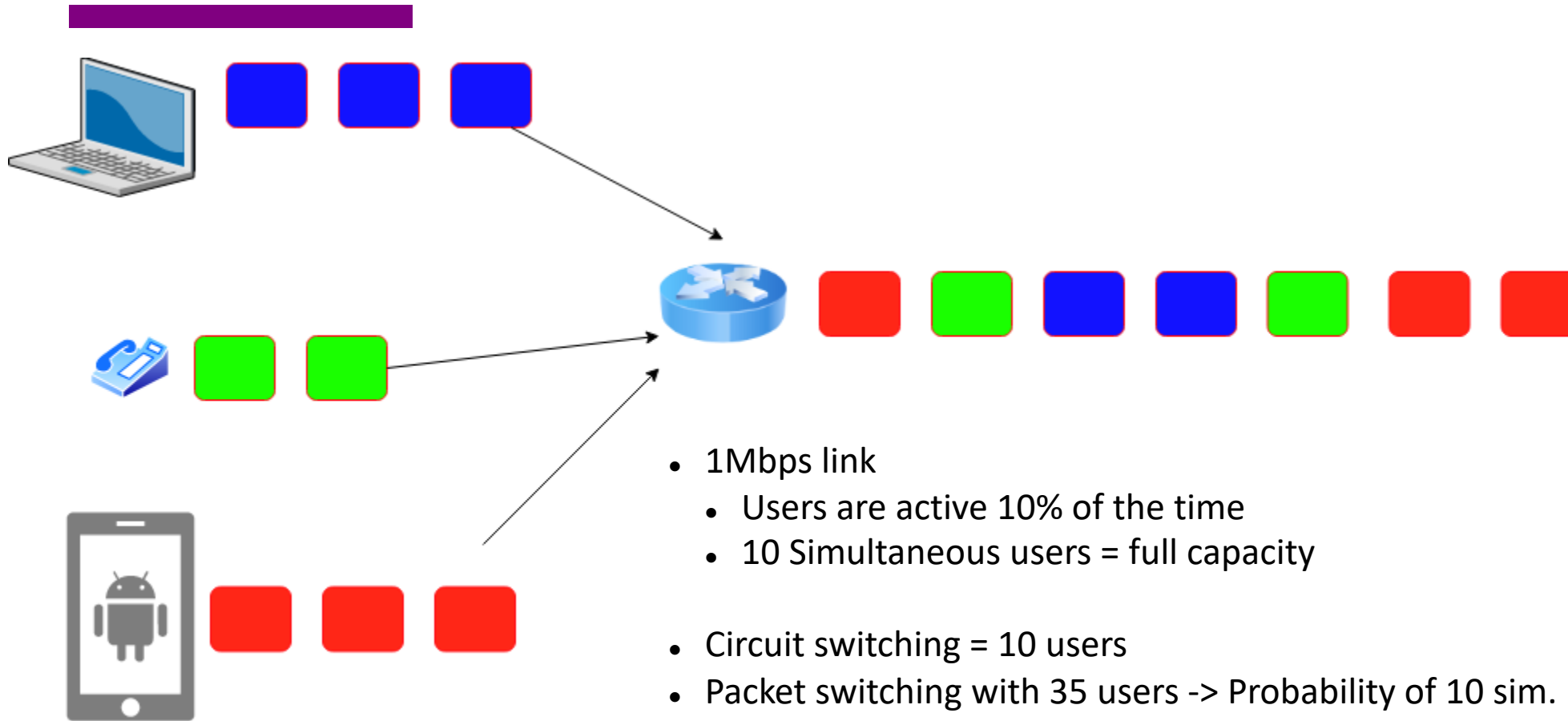
# Statistical Multiplexing for Packet Switching

---





# How many users can you support?



- 1Mbps link
  - Users are active 10% of the time
  - 10 Simultaneous users = full capacity
- Circuit switching = 10 users
- Packet switching with 35 users -> Probability of 10 sim. Users < 0.0004

<https://math.stackexchange.com/questions/918861/probability-problem-in-networking>

# Circuit vs Packet Switching



- **Circuit Switching**

- Dedicated resource divided among participants
- Requires setup, guaranteed performance (unless the link breaks)

- **Packet Switching**

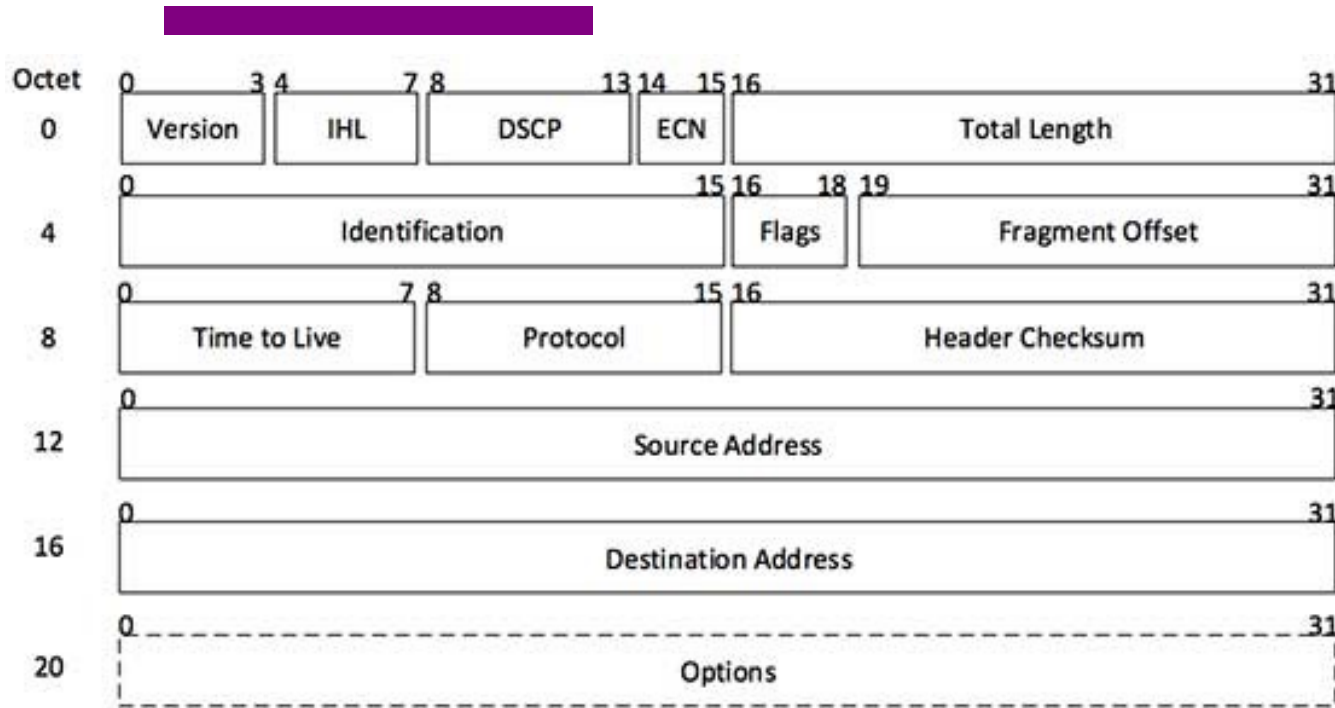
- Shared resource
- Use small chunks of data (packets), send as soon as possible
- Store-and-forward packets

# Internet Topology Zoo



<http://www.topology-zoo.org>

# But What is a Packet?



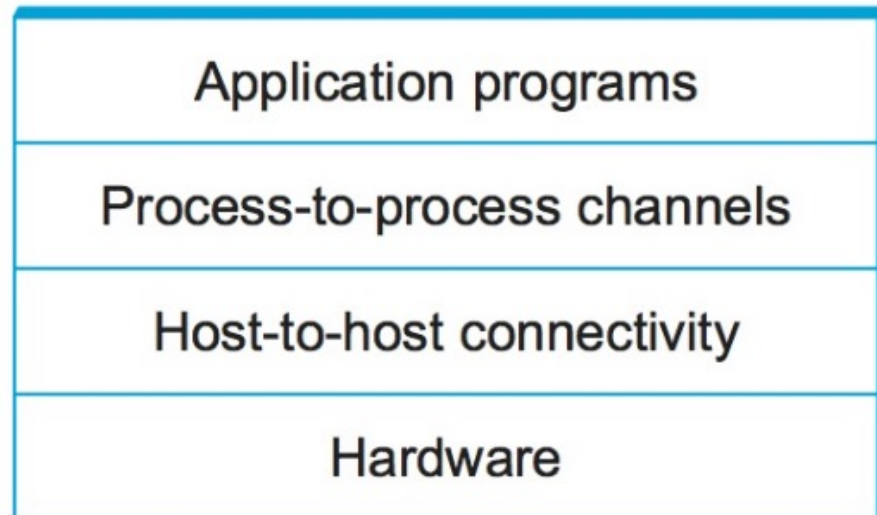
[Image: IP Header]

- Self-contained data unit
- Has two parts (generally)
  - Control information
  - Payload
- How do we transmit “Hello World?”
- How do we transmit a dictionary?

# Network Architecture

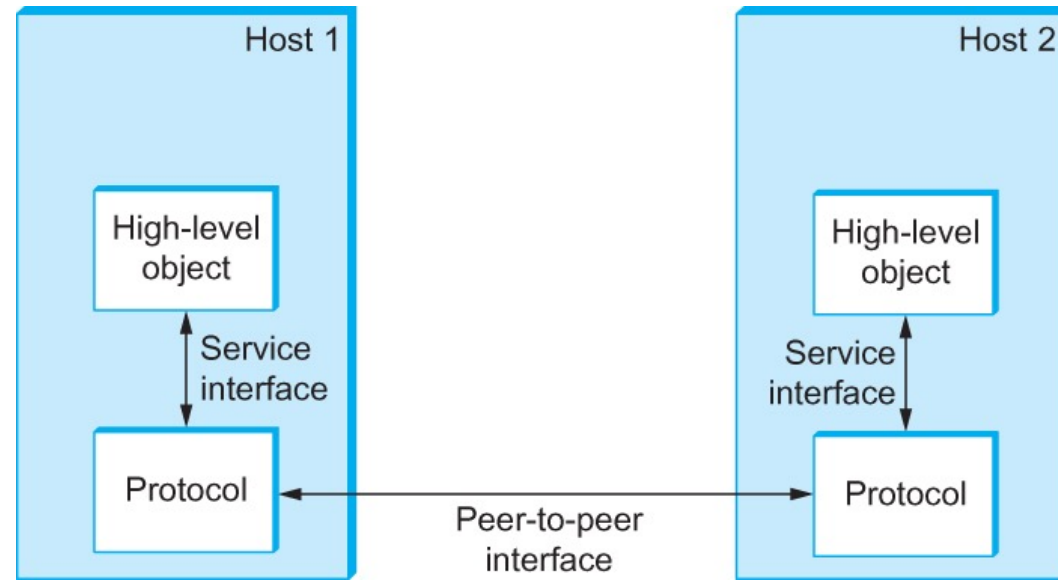
---

- What are the requirements from a network?
- Architecture = High-level blueprint
  - Protocols = Building blocks of the architecture
  - Layering = Break down the problem in smaller pieces



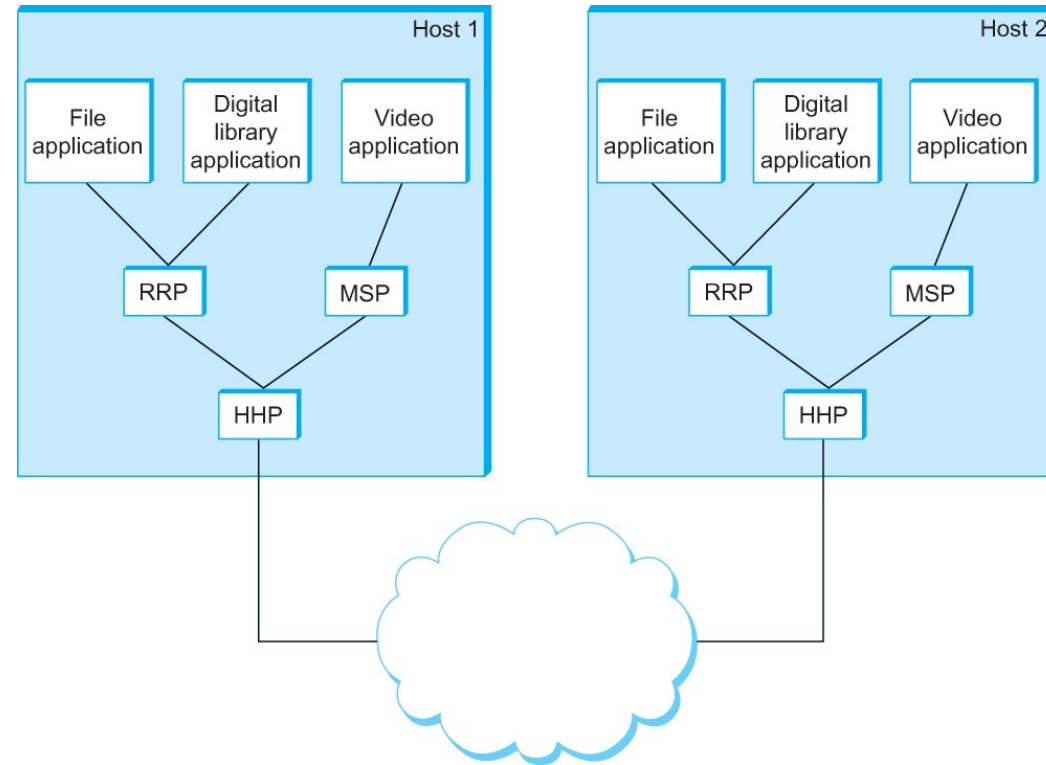
# Interfaces

---



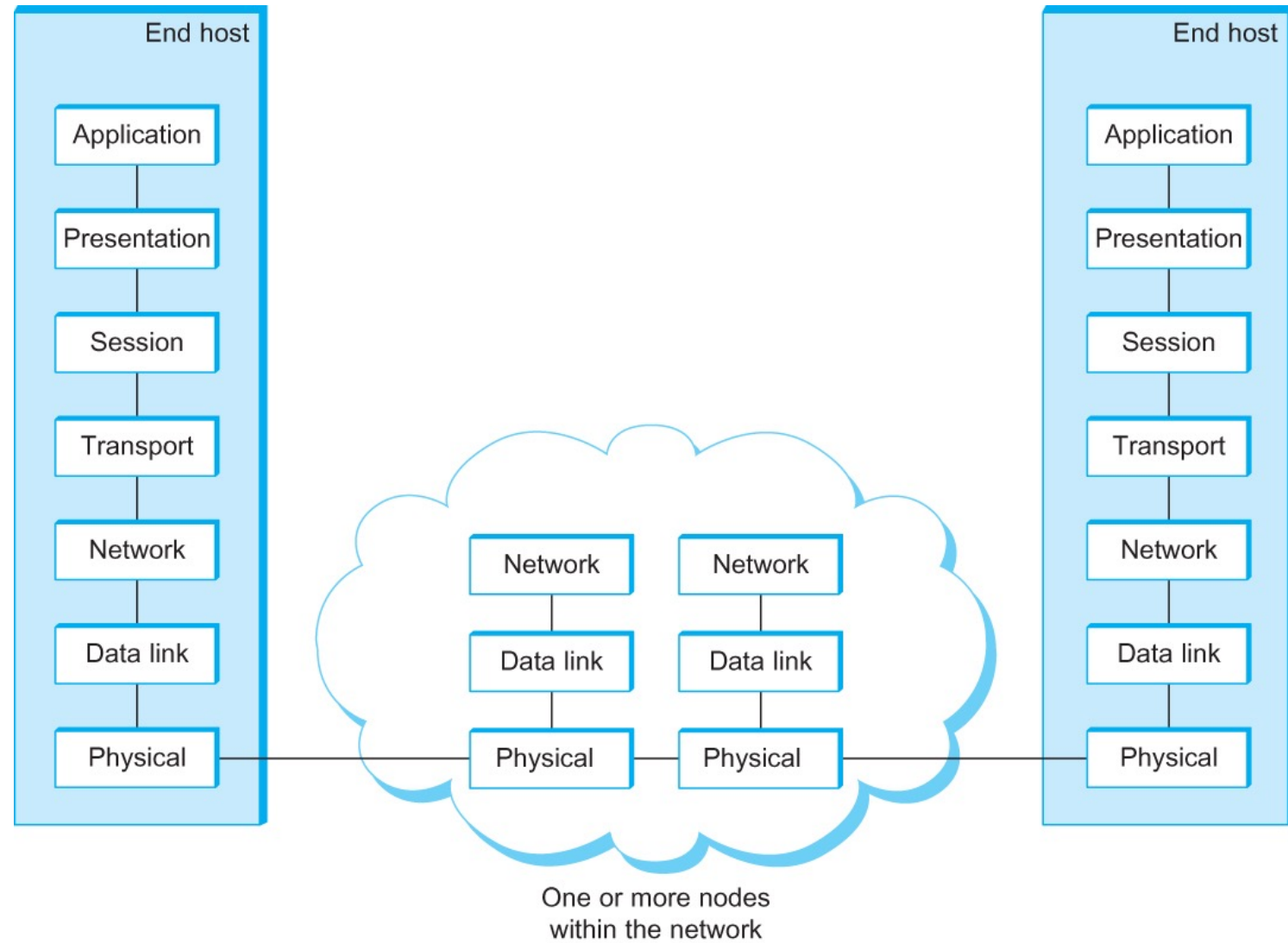
Service and Peer Interfaces

# Protocol Graph




Example of a protocol graph  
nodes are the protocols and links the “depends-on” relation

# OSI Architecture





# 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

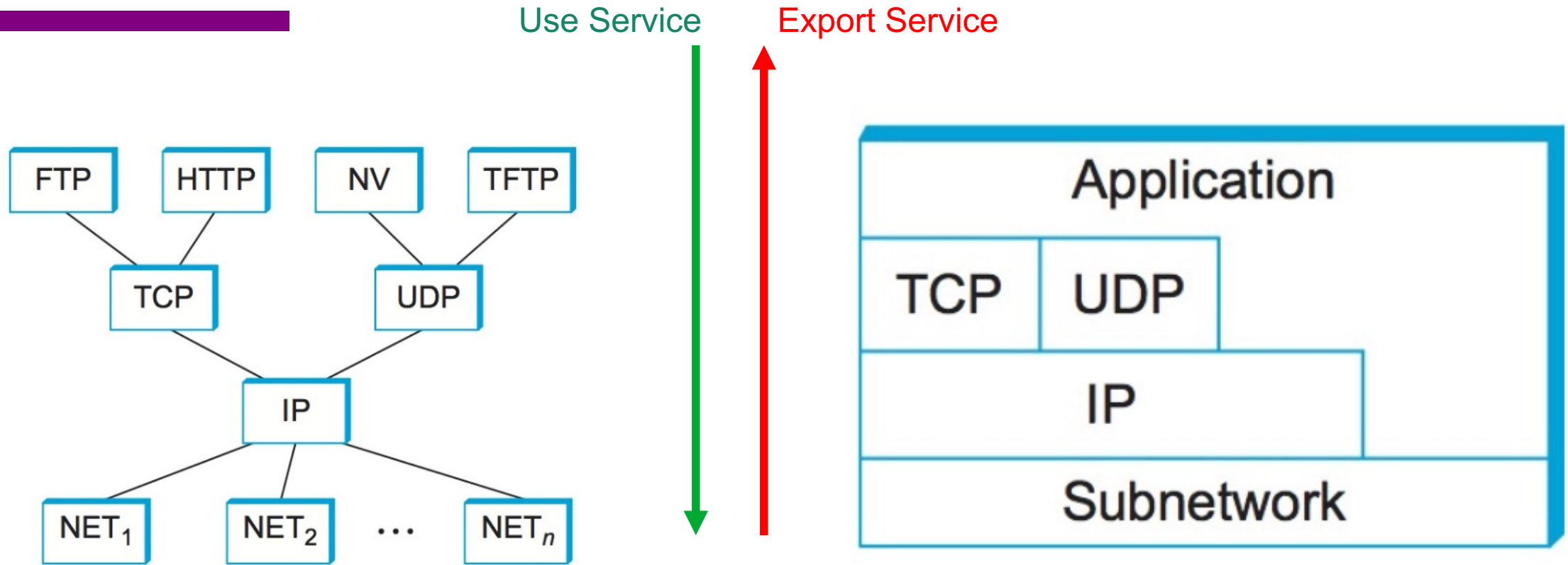
# Description of Layers

---

- 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

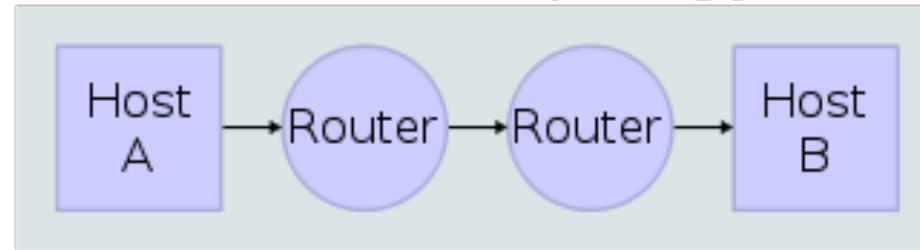
# Network Layers



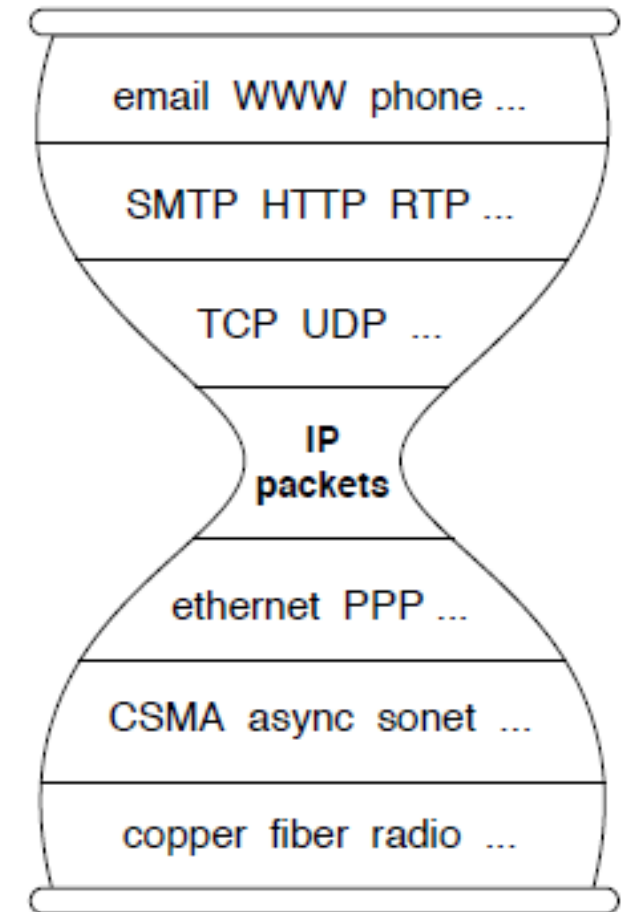
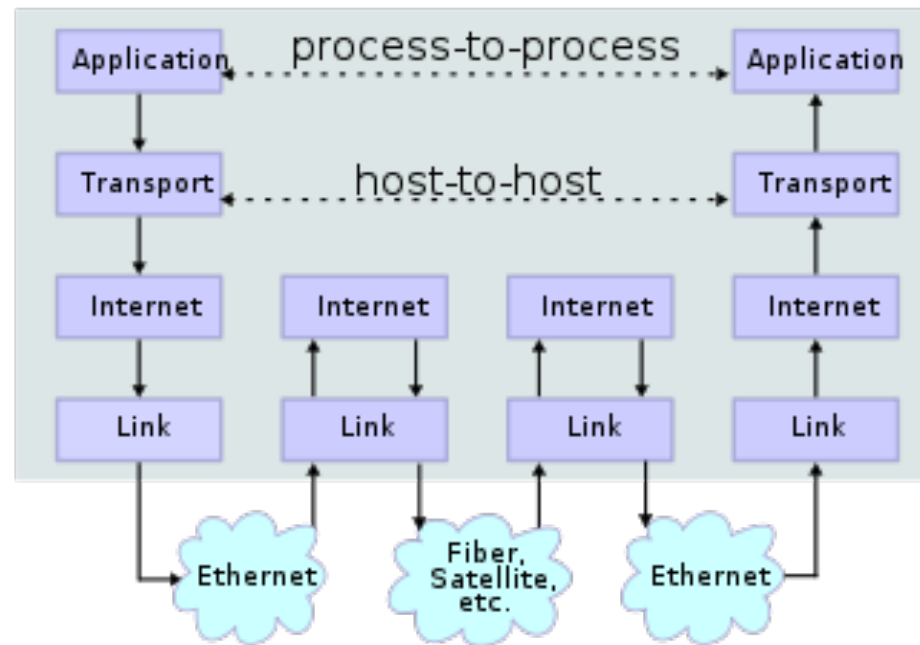
- Makes it easier to divide functionality
- Hides implementation details
- Few other reasons?

# IP Suite

## Network Topology



## Data Flow



We reject kings, presidents, and voting. We believe in rough consensus and running code. (David Clark, IETF, July 1992)

wikipedia

# Reading Assignment

---

- Read the overview of Chapter 1 - “Problem: Building a Network”
  - <https://book.systemsapproach.org/foundation/problem.html#problem-building-a-network>
  - **About 5 minutes**
- Read Chapter 1.2
  - <https://book.systemsapproach.org/foundation/architecture.html#architecture>
  - **About 45 minutes**
- Reach Chapter 1.3
  - <https://book.systemsapproach.org/foundation/architecture.html#architecture>
  - **About 45 minutes**