

Introduction To Networks

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 01

Semester 2, 2021

Important Contacts

- **Lecturer:** Prof. Ling Li

Room: 314.430

Phone: 9266 7939

Email: l.li@curtin.edu.au

Contact via email preferred*



- **Tutor:** Dr. Nadith Pathirage

Email: nadith.pathirage@curtin.edu.au

Contact via email preferred*



Unit Overview

▪ Lecture Component:

- 12 Lectures (1x2 hours)
- 1st week onwards

▪ Tutorial Component:

- 11 Tutorials (1x2 hours)
- Supervised
- 2nd week onwards

▪ Practical Component:

- 12 Lab Sheets (1x2 hours)
- Unsupervised (an online help session every fortnight)
- 1st week onwards

The content will be assessed in the **Final Examination**

Week 1~5: Incremental Task 01
Week 5~8: Incremental Task 02
Week 8~13: Incremental Task 03

Unit Contents

- **Fundamental knowledge on the design and implementation of networks and network protocols**

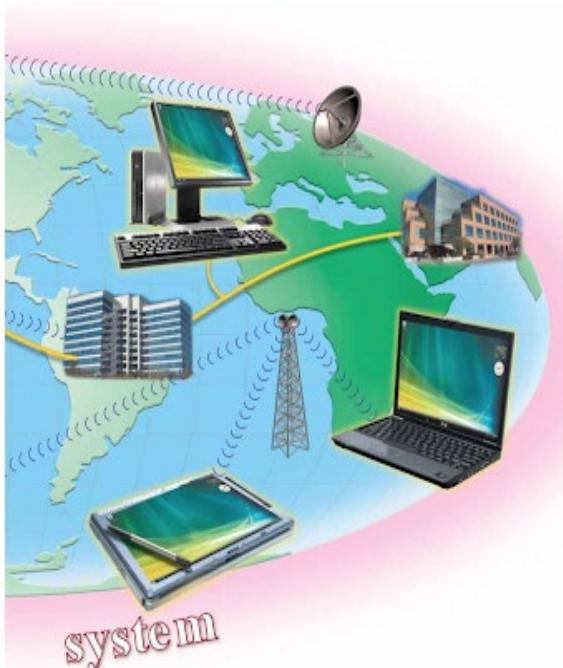
- Types of Network
- Basic Network Components
- Networking Protocols and Standards
- The Internet
- Emerging topics such as Internet of Things, Blockchain etc.



FIGURE 8-1 An example of a communications system. The communications channel consists of telephone and power lines, cable television and other underground lines, microwave stations, and satellites.

Why study Computer Communications

- A **core topic** in the IEEE Body of Knowledge (BoK) for **all** computing related degrees
- **Foundation** for more advanced units in networking and cyber security



Time table

- **Lectures:**

Mondays 10 to 12, Online via Collaborate Ultra

- **Workshops/Tutorials:**

- Tuesdays 8~10 am: B400.R230
- Thursdays 12~2 pm: Online via Collaborate Ultra
- Fridays 10~12 noon: B201.R309
- Fridays 12~2 pm: B212.R107

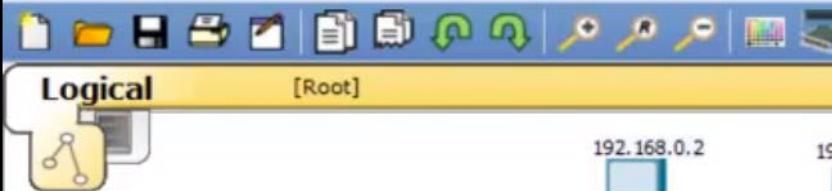


Time table

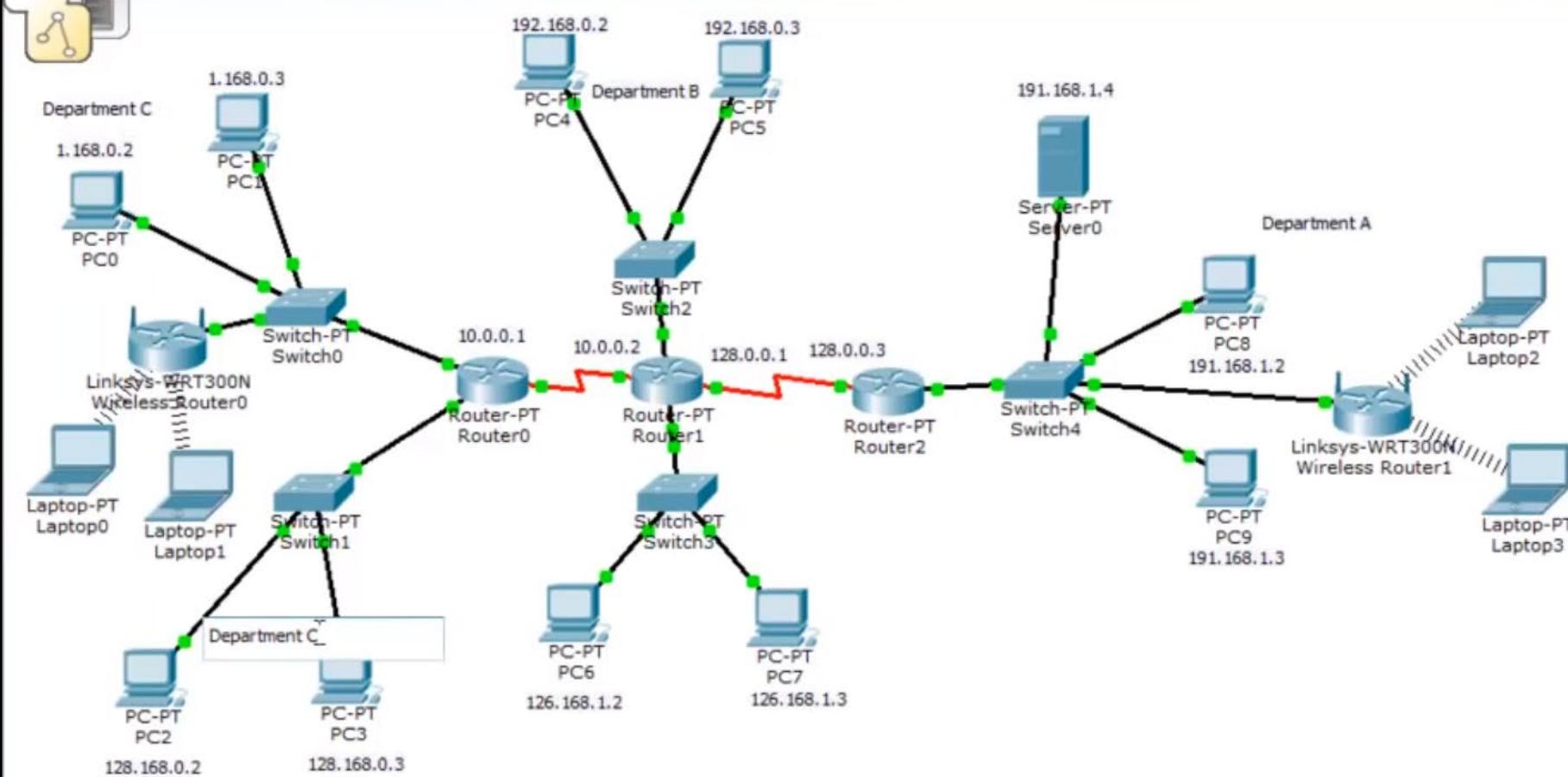
▪ **Practicals:**

- Practical aspect of networking with simulation tools (Cisco Packet tracer, Wireshark etc.)
- Weekly practical sheets starting from Week 1
- Detailed step-by-step instructions
- **Essential to complete them every week**
- A help session conducted every fortnight (starting from Week 2) at 2 to 4 pm Wednesdays, online via Collaborate Ultra. Please come with your questions.





Logical [Root] New Cluster Move Object Set Tiled Background Viewport



Packet Tracer

- Cross-platform visual simulation tool
- Emulate different network devices
- Simulate networks in action
- Monitor data flow
- Observe operations of protocols
- More & more !

Time: 00:03:42 Power Cycle Devices Fast Forward Time



Assessments

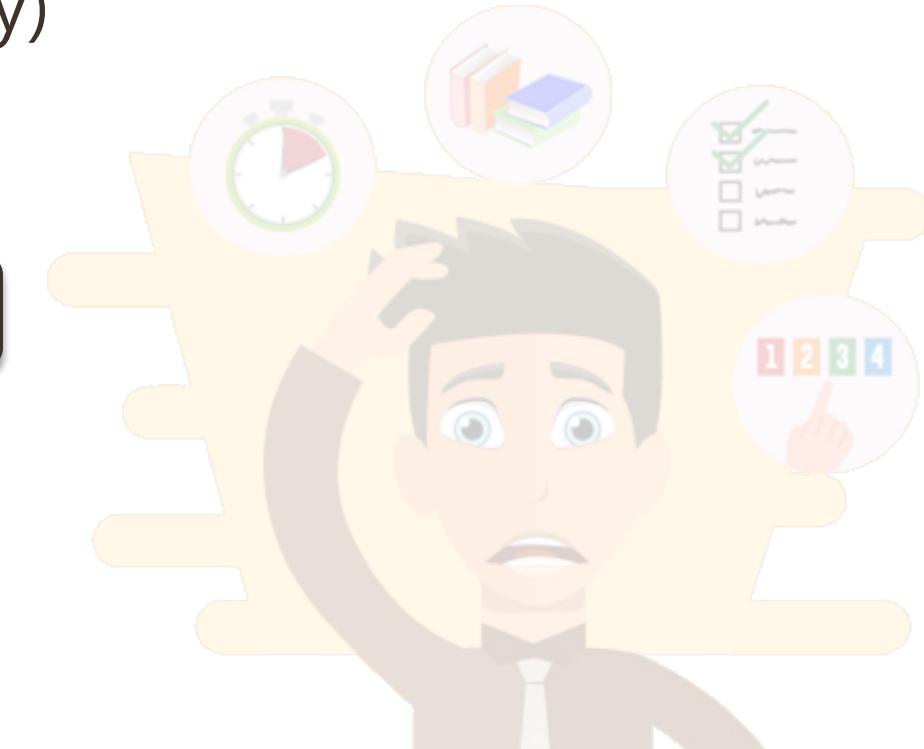
- Practical Incremental Tasks – **50%**
(3 tasks worth 15%, 15%, 20% respectively)
- Final Examination – **50%**

Compulsory conditions to pass the unit

Minimum mark of **40%** in the **Final Examination**

Minimum **overall mark** of **50%**

Must Fulfil all assessment components



Semester Week	Teaching Week	Lecture Name	Tutorial	Lab Sheet	Practical Incremental Tasks	
1	1 (July 26)	01 Introduction to Networks	No Tutorial in Week 1	P0 Packet Tracer Basics	Task No.1 Announced	
2	2 (Aug 02)	02 Physical Layer	T1 Introduction to Networks	P1 Working with Physical Workspace		
3	3 (Aug 09)	03 Data Link I	T2 Physical Layer	P2 Working with Logical Workspace		
4	4 (Aug 16)	04 Data Link II	T3 Data Link I	P3 Networking with Layer 1 Devices, Simulating Internet		
5	5 (Aug 23)	05 Network Layer I	T4 Data Link II	P4 Networking with Layer 2 Devices, VLAN I	Task No.1 due	Task No.2 announced
6	6 (Aug 30)	06 Network Layer II	T5 Network Layer I	P5 Networking with Routers (IPv4, IPv6, VLAN II, VOIP)		
7	(Sep 06)	Tuition Free				
8	7 (Sep 13)	07 Transport Layer I	T6 Network Layer II	P6 Static/Dynamic Routing (RIP, RIPv2, EIGRP, OSPF)		
9	8 (Sep 20)	08 Packet Tracer	T7 Transport Layer I	P7 Networking with TCP (Client-Server)	Task No.2 due	Task No.3 announced
10	9 (Sep 27)	09 Transport Layer II	Mock Paper Discussion	P8 Networking with UDP (Client-Server)		
11	10 (Oct 4)	10 Application Layer I	T8 Transport Layer II	P9 Networking with Application Layer I (Telnet, FTP, HTTP, Email)		
12	11 (Oct 11)	11 Application Layer II	T9 App Layer I	P10 Networking with Application Layer II (DHCP, DNS, P2P)		
13	12 (Oct 18)	12 Emerging Networking Tech	T10 App Layer II / T11 Emerging Network Tech	P11 Networking with IoT (Internet of Things)		
(Oct 25)	(Oct 25)	Study Week				Task No.3 due
		Final Examination				

Color indicate the difficulty

Text, Video and References

- **Official Text:** None
- **Recommended references:**
 - All materials (text, web, video) relevant to the unit are listed on Blackboard under each week.
- **Lecture Slides / Tutorial Sheets / Lab Sheets**
 - Available electronically through Blackboard.



Blackboard

Computer
Communications
(Semester 2 2020 Bentley
Campus - INT[1])

Announcements

Unit Information

Assessments

My Grades

Message Board

Calender

iLecture

Collaborate Ultra

Reading List

Learning Materials

Packet Tracer

Help Me!

Discussion Board

Glossary

Wiki

Help Me! 

Build Content 

Assessments 

Tools 



Discussion Board

 subscribe

This is a place (**discussion forums**) where people can post questions, ideas or thoughts. It's a way of starting a conversation, about something important to you, that you hope other people will engage with and respond to.



Glossary

The important terms found throughout the course of study.

These terms will be helpful in studying for the **Examinations of this Unit**.



Wiki(s)

Wikipedias maintained for the unit.

HELP IS
AVAILABLE





Discussion Board (Forums)

- Unit FAQs Questions on unit assignments, structure, examinations.

- Learning Material FAQs Questions on learning materials, concepts, tutorials (**both Computing and Engineering students**).

- CNCO2000 Practicals FAQs Questions on lab sheets (Packet Tracer) for CNCO2000 (**Computing students only**).

- CMPE2000 Practicals FAQs Questions on CCNA labs, netcad, etc for CMPE2000 (**Engineering students only**).

- 27/07/20 13:07 P3: Networking with Layer 1 Devices, Simulating Internet

- 27/07/20 13:07 P2: Working with Physical & Logical Workspace

- 27/07/20 13:02 P1: Packet Tracer Basics

- 10/07/20 00:17 General

Forum: Learning Material FAQs

In a thread, you can view the post and information abo

Subscribe

- 27/07/20 12:23 L03/T03: Data Link Layer I

- 27/07/20 12:23 L02/T02: Physical Layer

- 27/07/20 12:17 L01/T01: Introduction To Networking

- 10/07/20 00:01 General



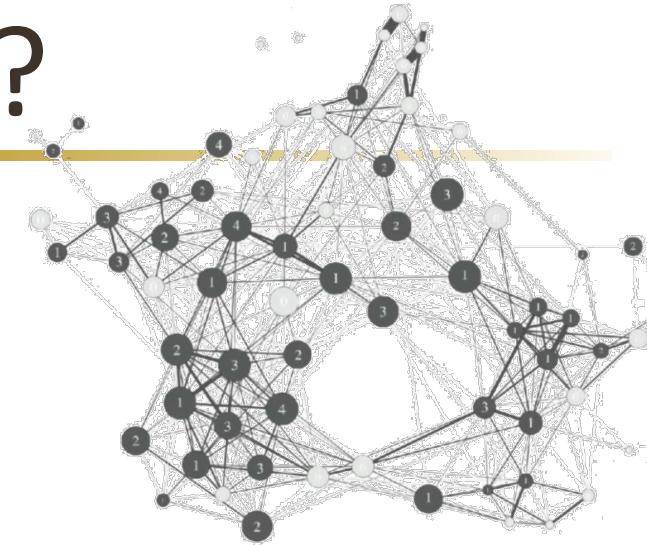
Networking

- What is a network?
- Data network elements
- Transmission Technology
 - Point-2-point (unicast)
 - Multi-point (multicast, broadcast)
- Scales of network
 - LAN, MAN, WAN,
 - WLAN, WMAN, WWAN, WPAN

What is a network?

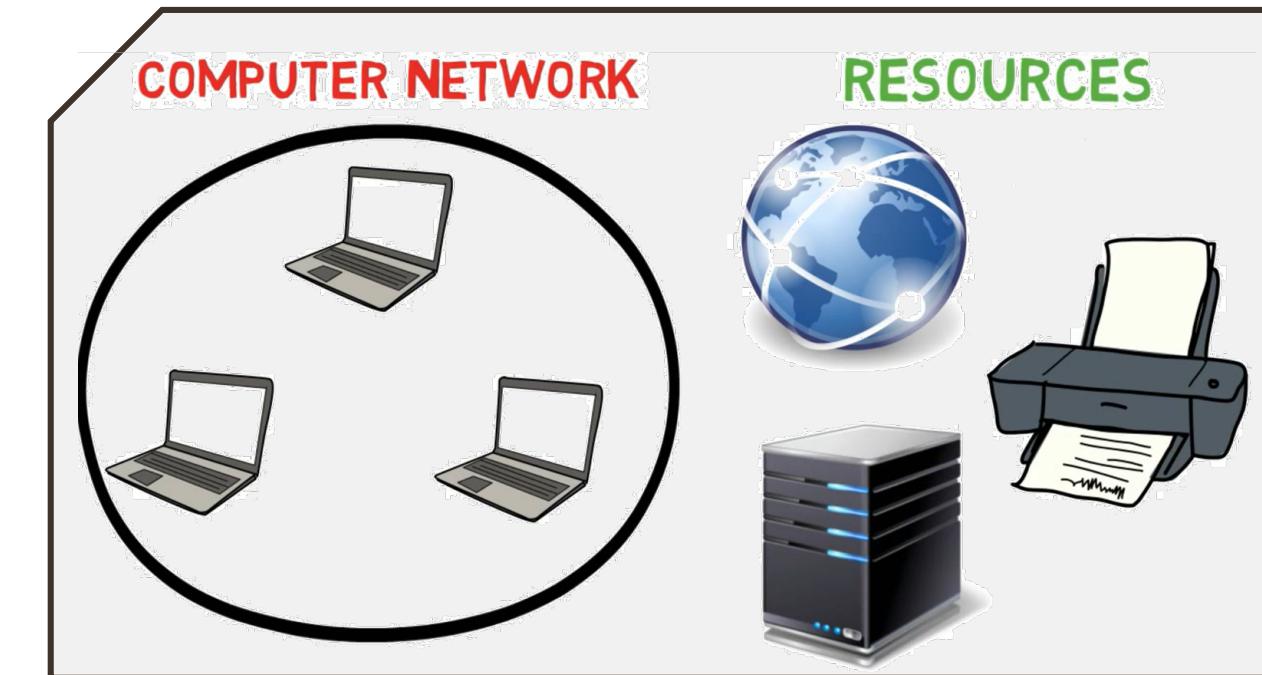
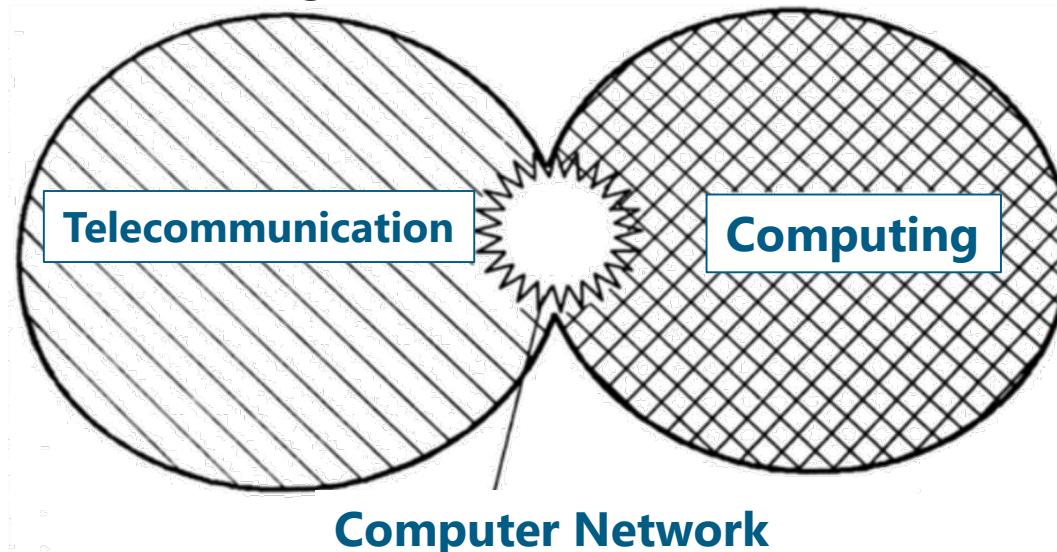
▪ What is a network?

- Group of interconnected things
 - E.g. Water supply system in ancient Rome

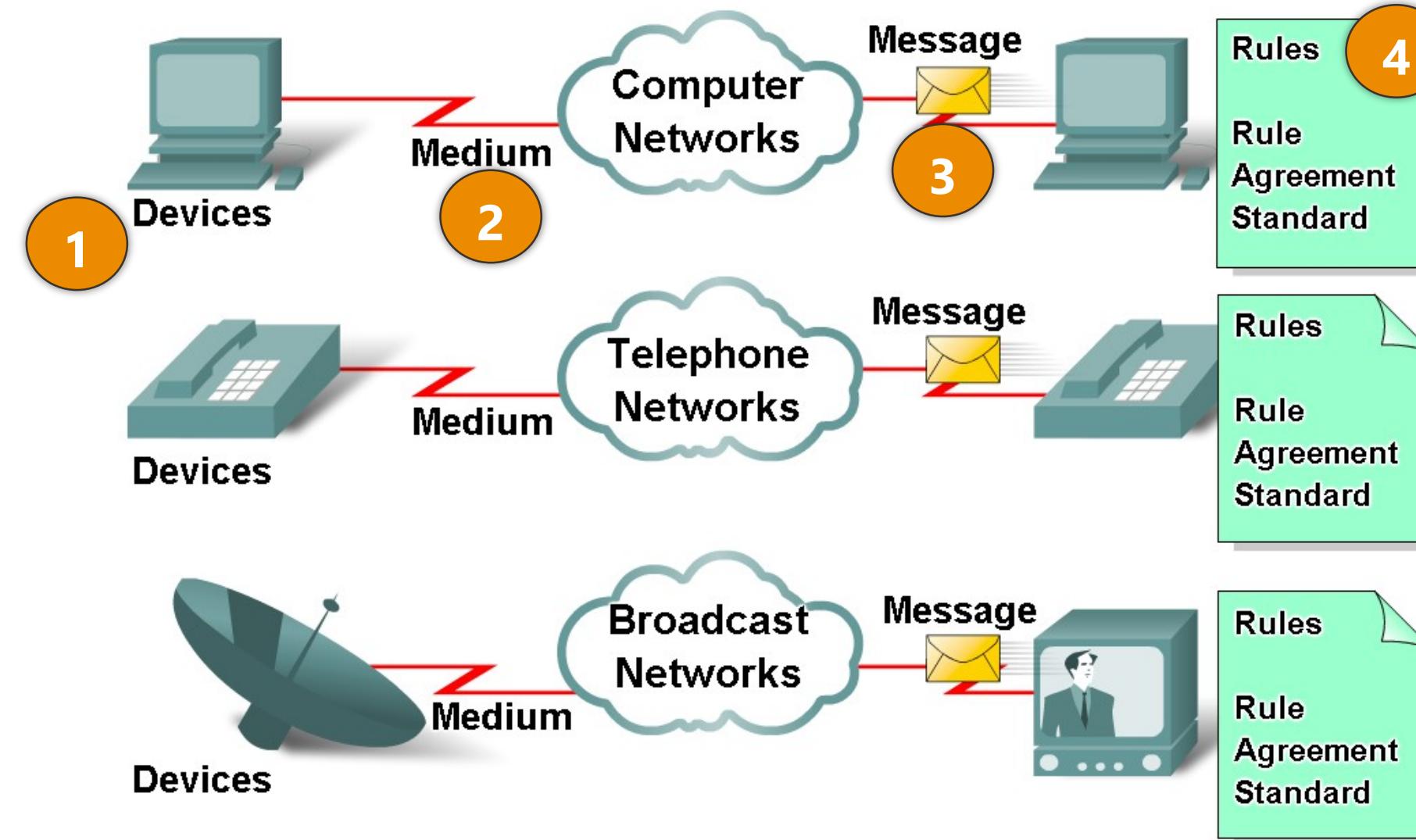


▪ What is a computer network?

- Group of interconnected computers
 - ✓ sharing resources, etc.



Data Networks & Elements



Rules (*protocols*)

- ✓ How messages flow across network
- ✓ How communication between peers will occur

Transmission Technology

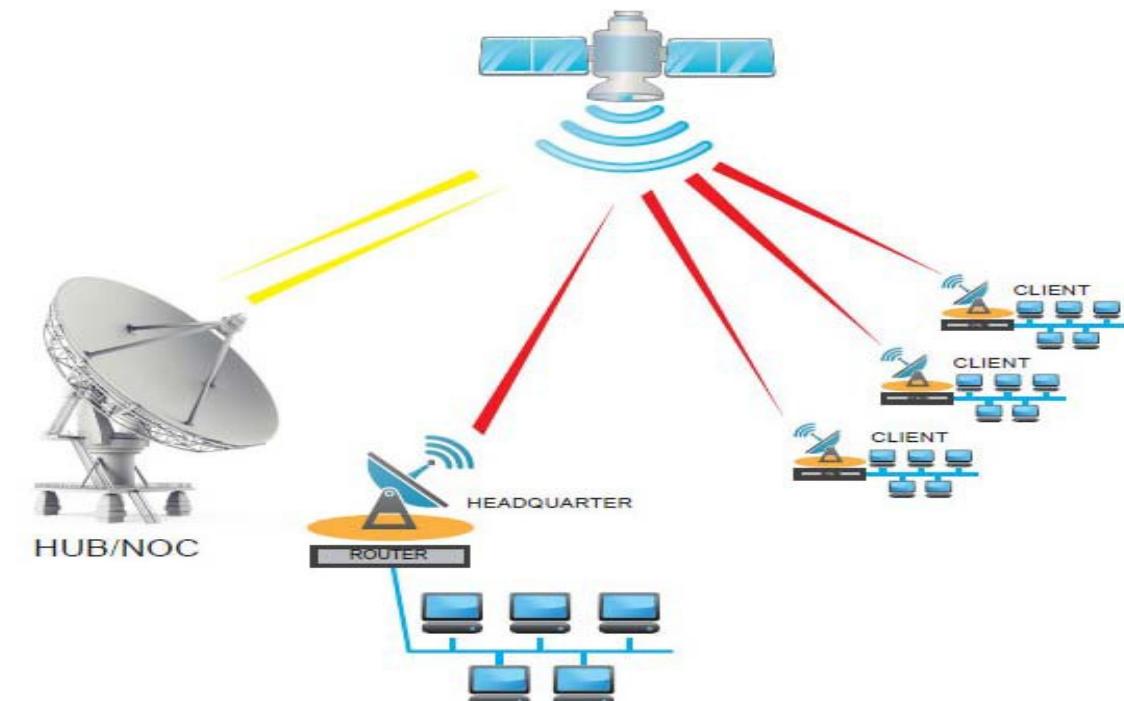
Point-to-point Network

- One sender and one receiver (dedicated links)
- **Unicasting**



Multipoint (Multidrop) Network

- Many nodes share the link capacity
- **Broadcasting, Multicasting**



Scales of Network

A **loosely coupled**
multiprocessor
system

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	
1 km	Campus	Local area network
10 km	City	
100 km	Country	
1000 km	Continent	Metropolitan area network
10,000 km	Planet	

Local Area Network (LAN)

- **Privately-owned networks**
- **Restricted in size**
 - within a building or campus.
- **Transmission technology**
 - a **single cable** – 10Mbps/100Mbps/10Gbps (400 Gbps in 2017?)
 - **wireless** transmission – 2Mbps/11Mbps/54Mbps/108Mbps/250Mbps
- Various **topologies** are possible
 - **Bus, Ring and Star**

Metropolitan Area Network (MAN)

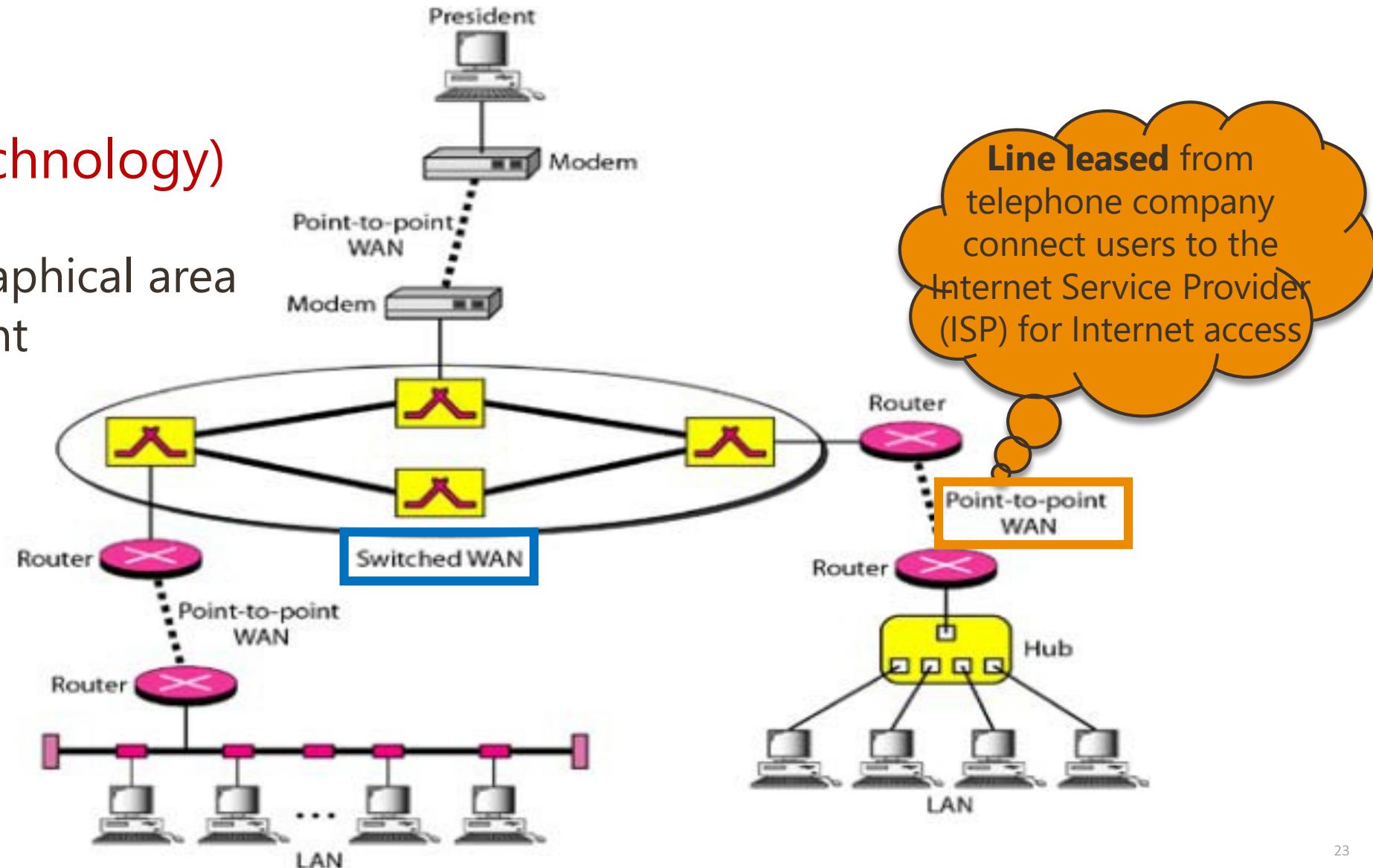
- **LARGE LAN** (uses same LAN technology)
- **Size**
 - Cover a group of nearby corporate office or a city
- **Transmission technology**
 - High-speed backbone linking multiple LAN's
 - ✓ Digital Subscriber Line (DSL)
 - ✓ TV cables
 - ✓ Fiber Distributed Data Interface (FDDI)
 - ✓ Distributed-Queue Dual-Bus (DQDB) network

Wide Area Network (WAN)

LARGE MAN

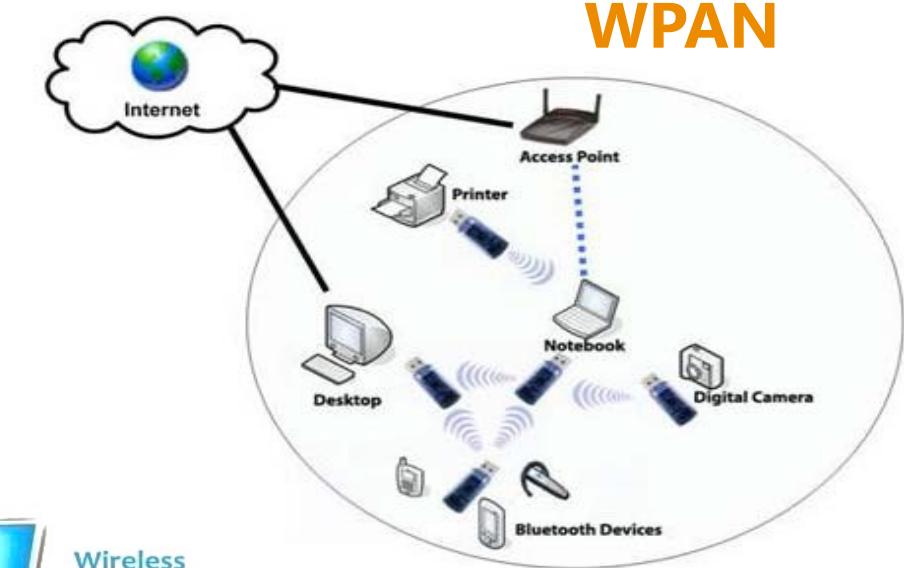
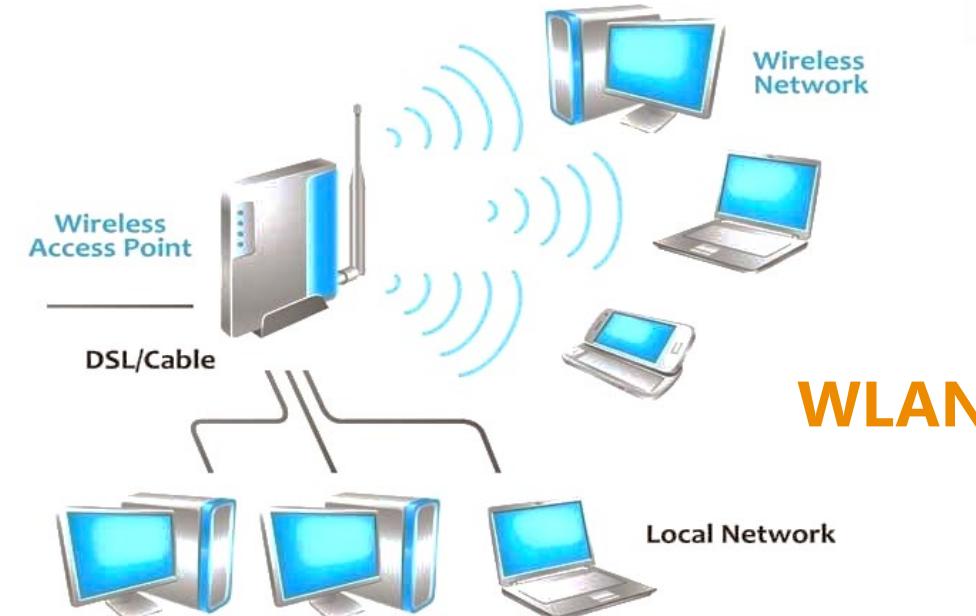
(uses different technology)

Size: A large geographical area
- country or continent



Wireless Networks

- Wireless Personal Area Networks (WPAN)
- Wireless LAN (WLAN)
- Wireless MAN (WMAN)
- Wireless WAN (WWAN)
 - Cellular Networks



WPAN



Classification Of Networks

- Circuit-switched networks
- Packet-switched networks
 - Datagram networks
 - Virtual Circuit networks

Classification of Networks

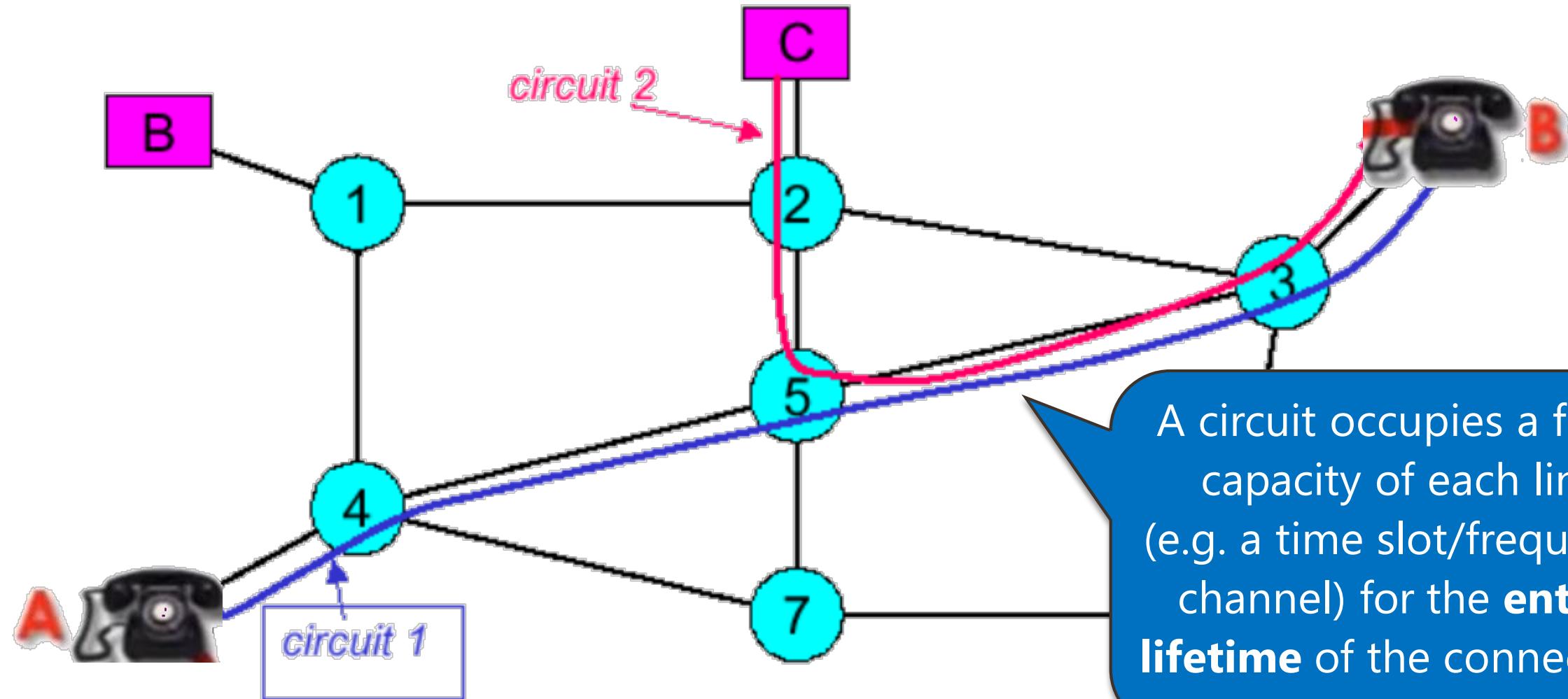
1. **Circuit-Switched** Networks

- ✓ Circuit Establishment
- ✓ Data Transfer
- ✓ Circuit Termination

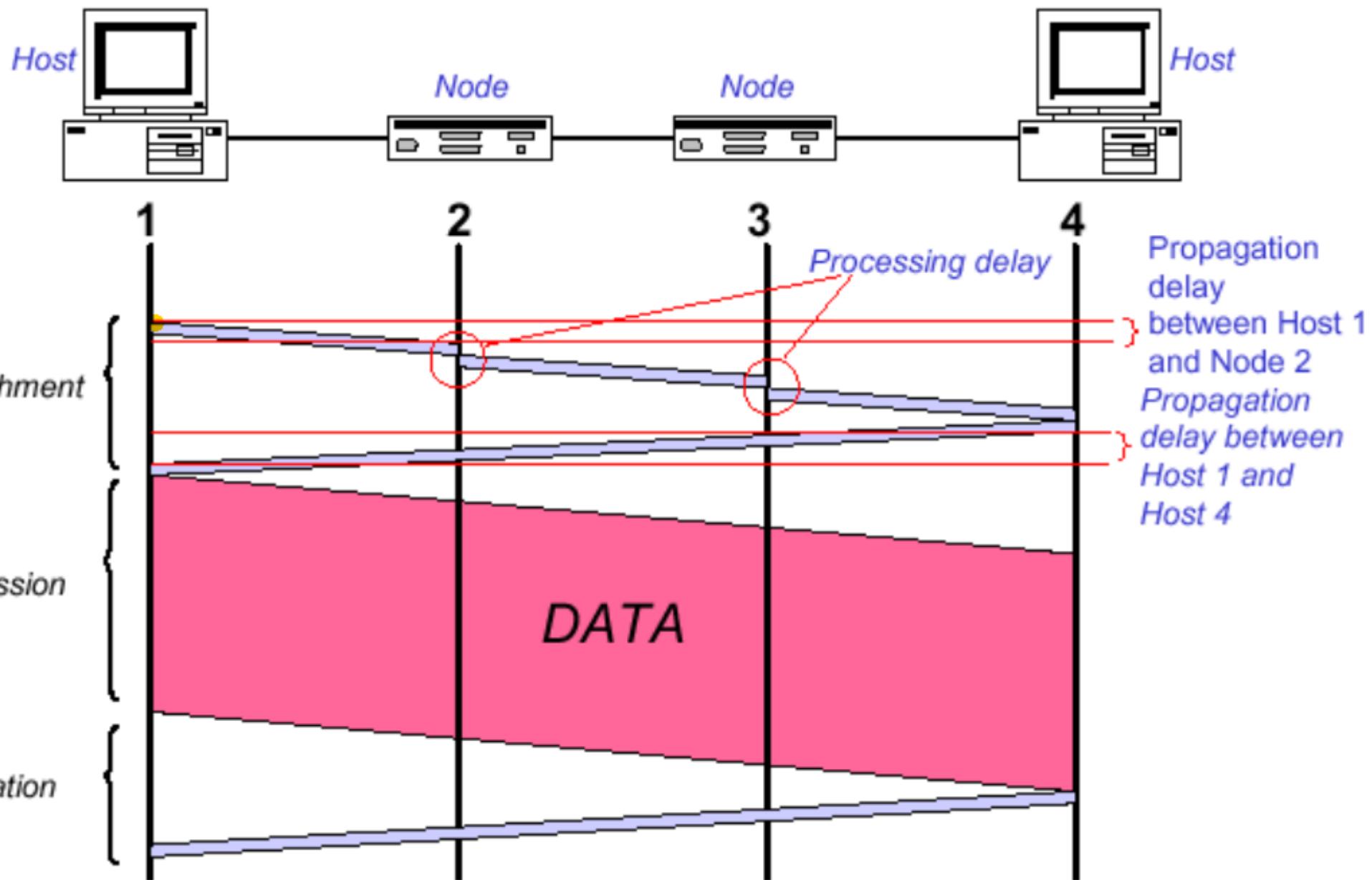
2. **Packet-Switched** Networks

- a) **Datagram** (Packet Switched) Networks
- b) **Virtual Circuit** (Packet Switched) Networks

Circuit Switched Network



Timing in Circuit Switched Network

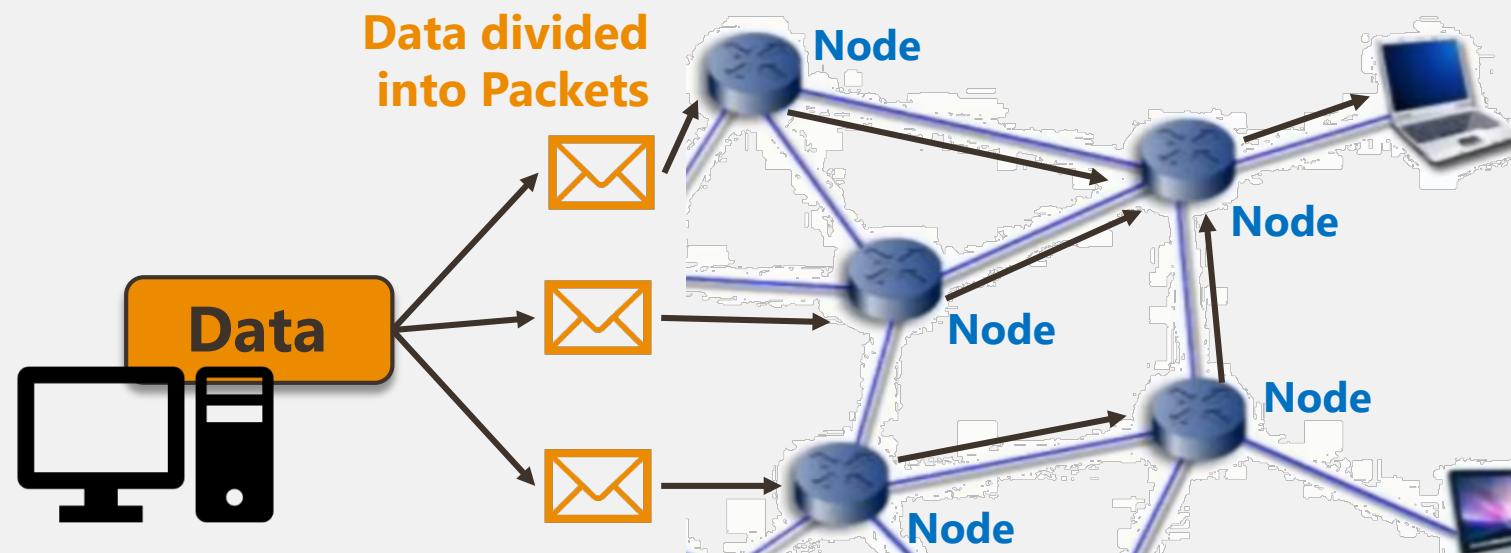


Datagram PS Network

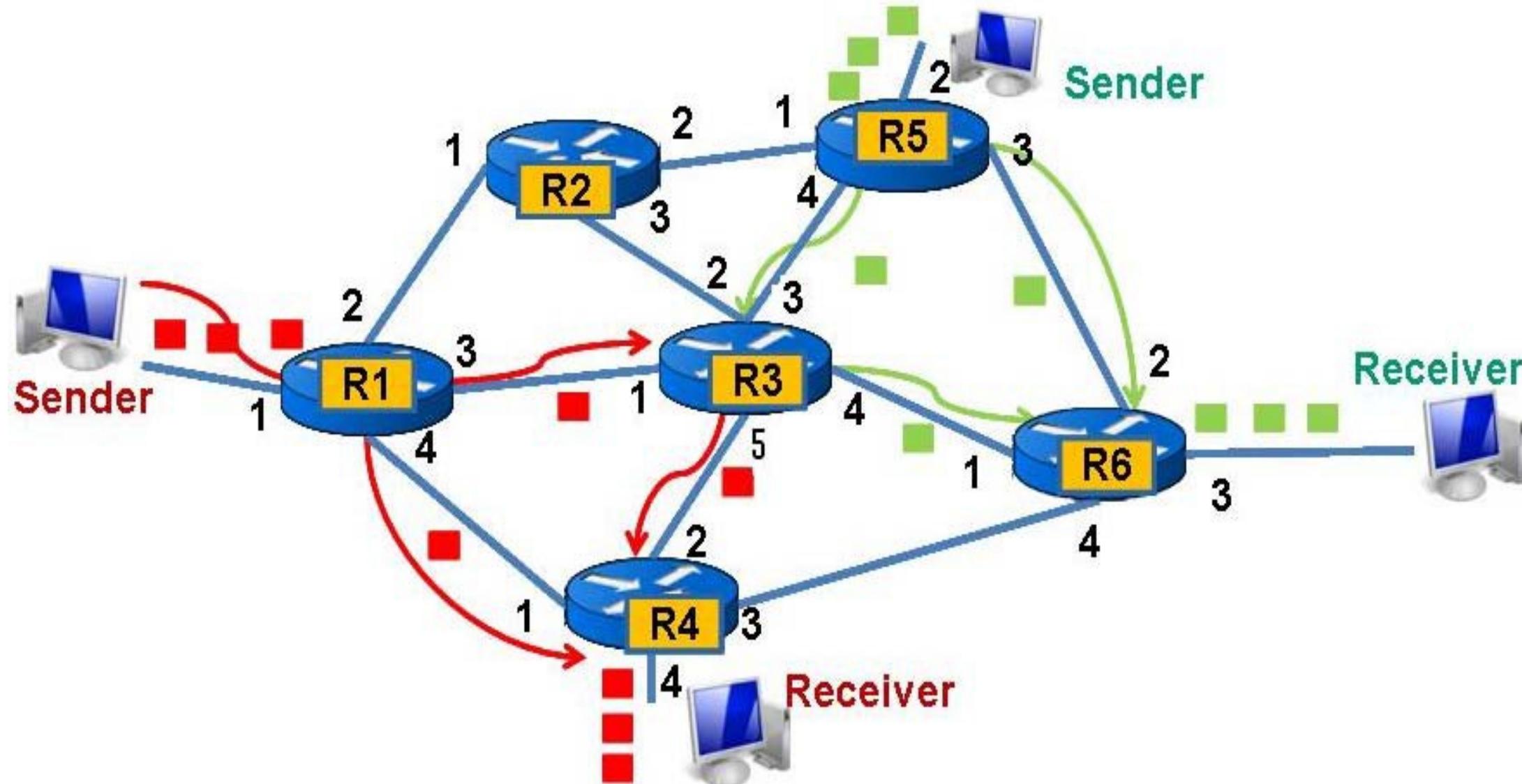
- **At a Node:**

packet stored briefly, and then forwarded to the next node (**store-and-forward**)

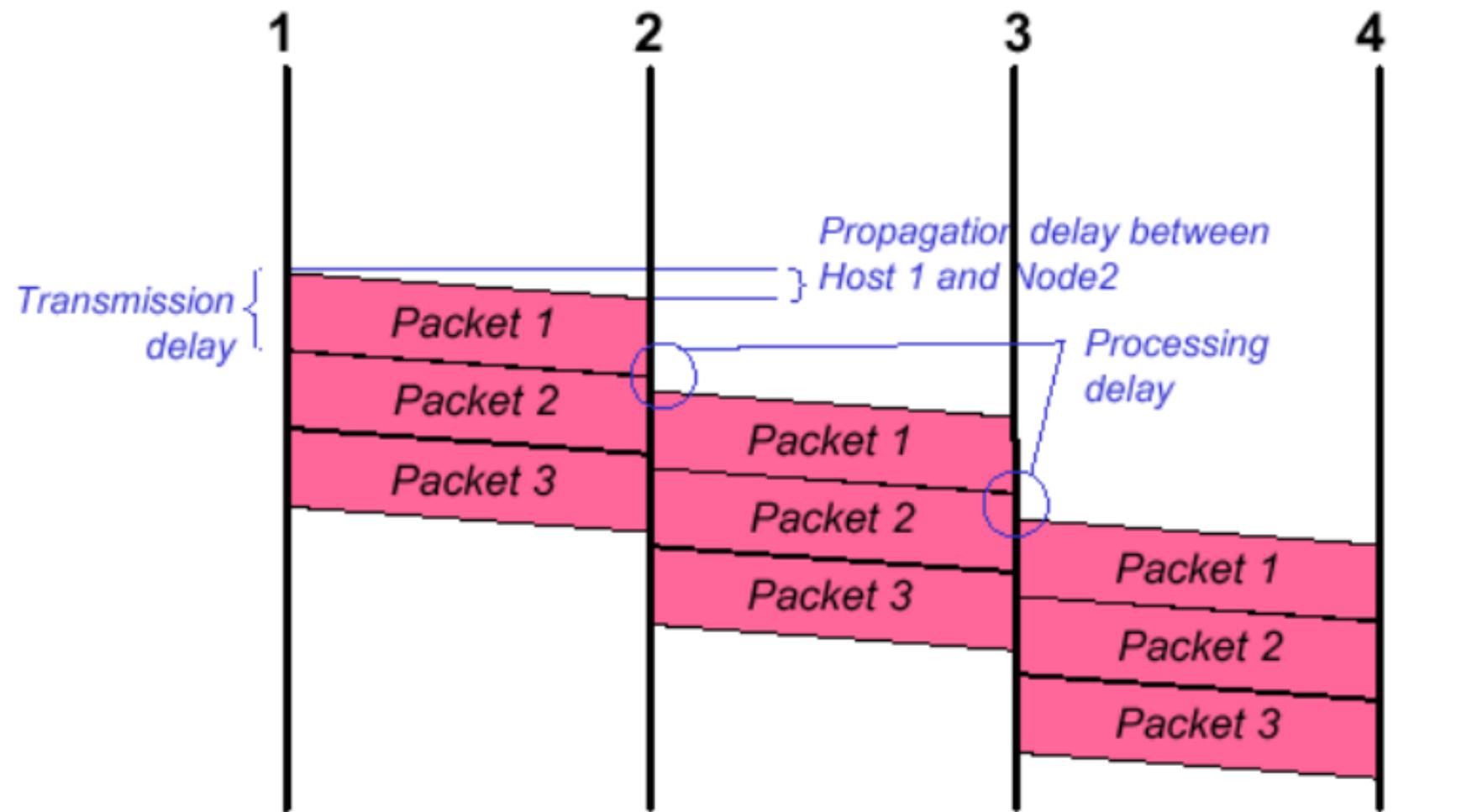
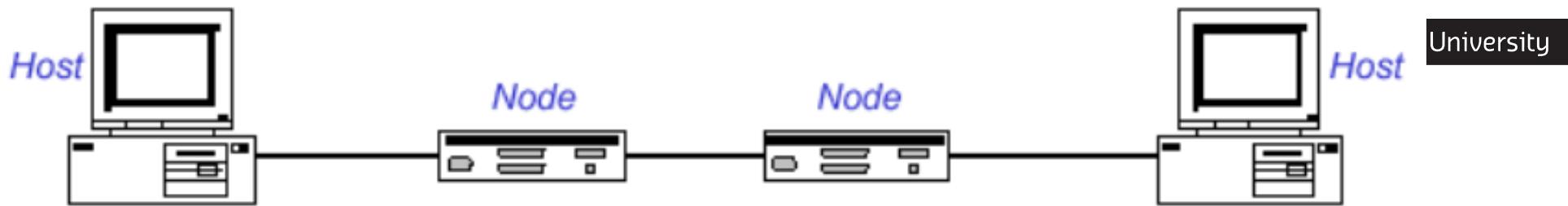
- **No physical capacity is allocated** for packets



Datagram PS Network



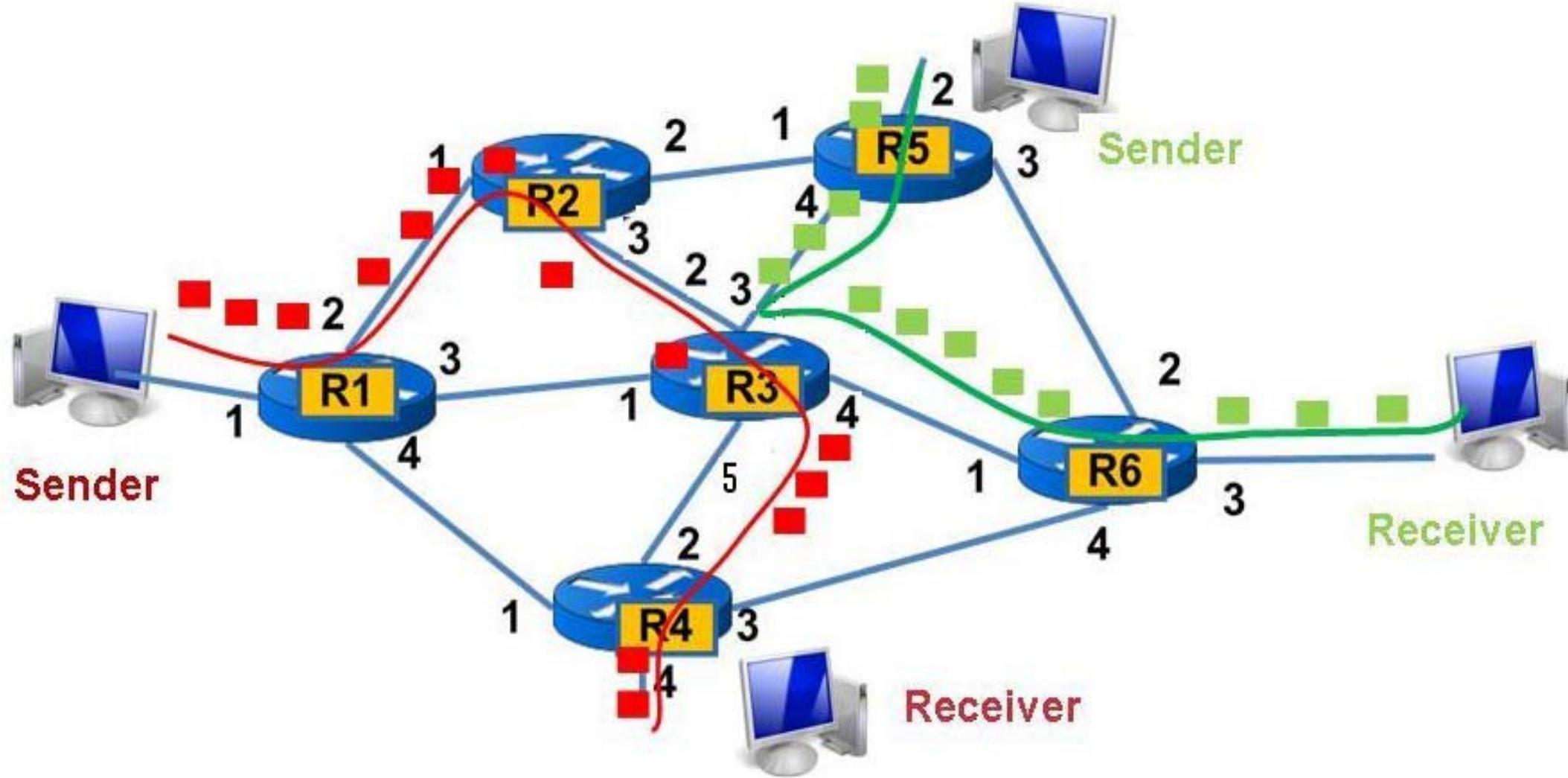
Timing in Datagram (PS) Networks



Virtual Circuit PS Network

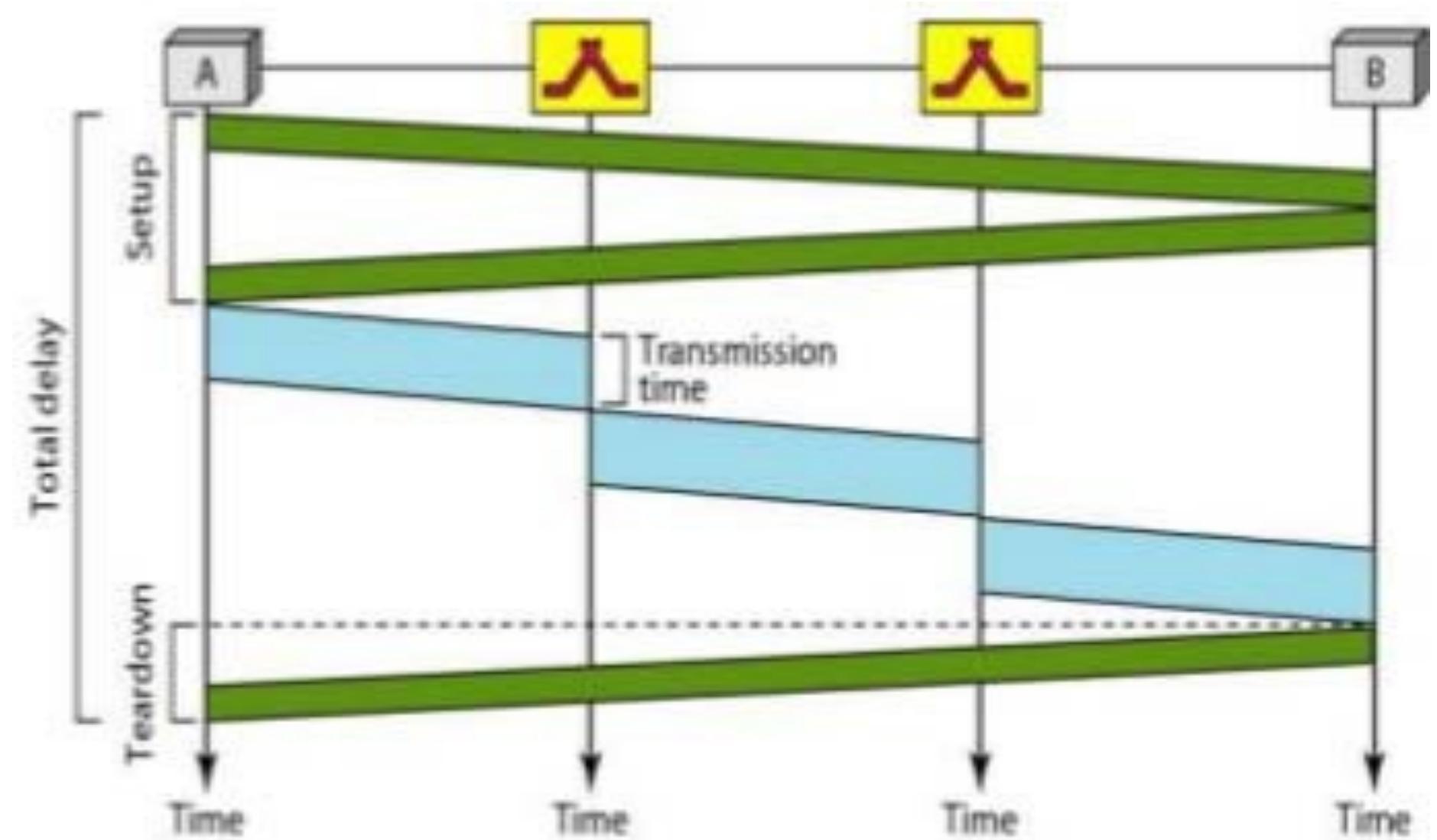
- **Hybrid:**
 - **Packet Switching + Circuit Switching**
- A **preplanned route** (between stations)
- All packets take the **same route**
 - *for the duration of the logical connections*
- **But, No dedicated path** between the two stations

Virtual Circuit PS Network



Virtual Circuit PS Network

Timing in Circuit Switched Network



Circuit Switching	Datagram Packet Switching	Virtual Circuit Packet Switching
Dedicated transmission path	No dedicated transmission path	No dedicated transmission path
Continuous transmission	Transmission of packets	Transmission of packets
Path stays fixed for entire connection	Route of each packet is independent	Path stays fixed for entire connection
Call setup delay	No setup delay	Call setup delay
No queuing delay	Queuing delay at switches	Queuing delay at switches
Busy signal overloaded network	Delay increases in overloaded networks	Delay increases in overloaded networks
Fixed bandwidth for each circuit	Bandwidth shared by all packets	Bandwidth shared by all packets
No overhead after call setup	Overhead in each packet	Overhead in each packet



Common Networking Technologies

- PSTN
- Internet
- ATM

Common Networking Technologies



Public Switched Telephone Networks (PSTN)

- ✓ The largest worldwide computer network, **specialized for voice**
 - ✓ Switching: **Circuit Switching**

The Internet

- ✓ A newer global and public information infrastructure
 - ✓ Switching: Datagram **Packet Switching** (Mostly)



ATM (asynchronous transfer mode; obsolete)

- ✓ **Flexibility** and **low-cost** ~ = Internet
 - ✓ **End-to-end service guarantees** ~ = telephone network
 - ✓ to replace telephone networks and data networks.
 - ✓ Switching: **VC Packet Switching**



Internet

- **Internet**

Network of geographically distributed computers

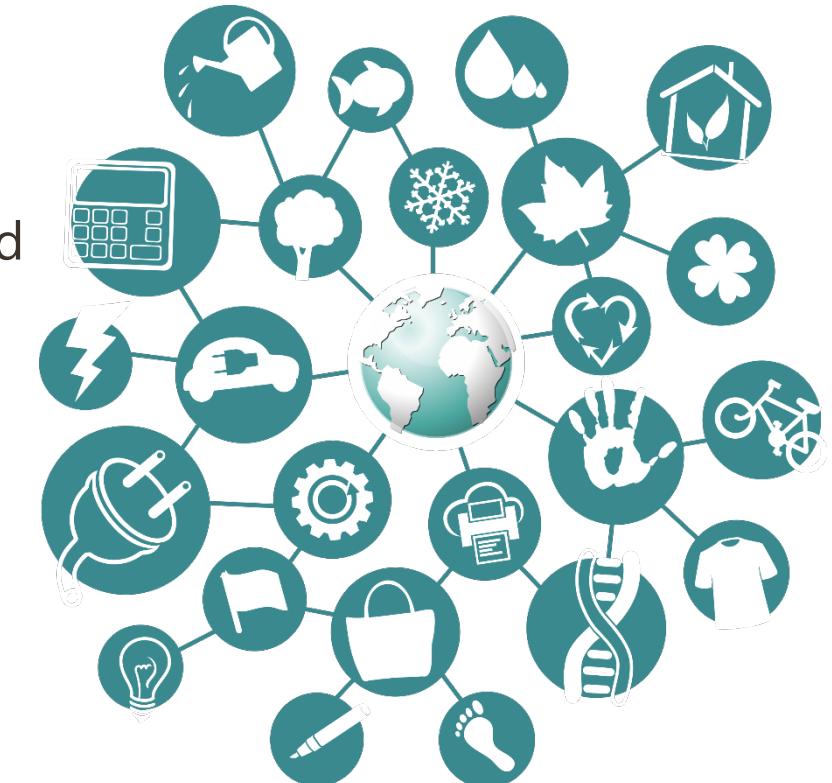
- Collection of interconnected network is known as **internetwork or internet**

- ✓ Common form of internet is a collection of LANs connected by a WAN

- End users use the internet via Internet Service providers (ISPs)

- **History**

- Early 1980: First personal computer
 - Early 1980: Commercial use of Internet
 - 1991: World Wide Web (www)





Layers, Services & Protocols

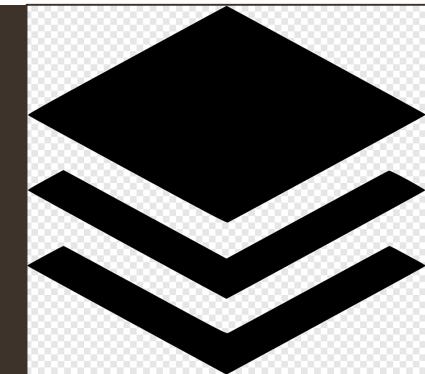
- Layering? Services? Protocol?
- Intro to OSI-7-layer model
- Intro to TCP/IP 5-layer model
- OSI 7-layer model vs TCP/IP 5-layer model

Layering, Services, Protocols

The overall communications process between two or more machines connected across one or more networks is **very complex**

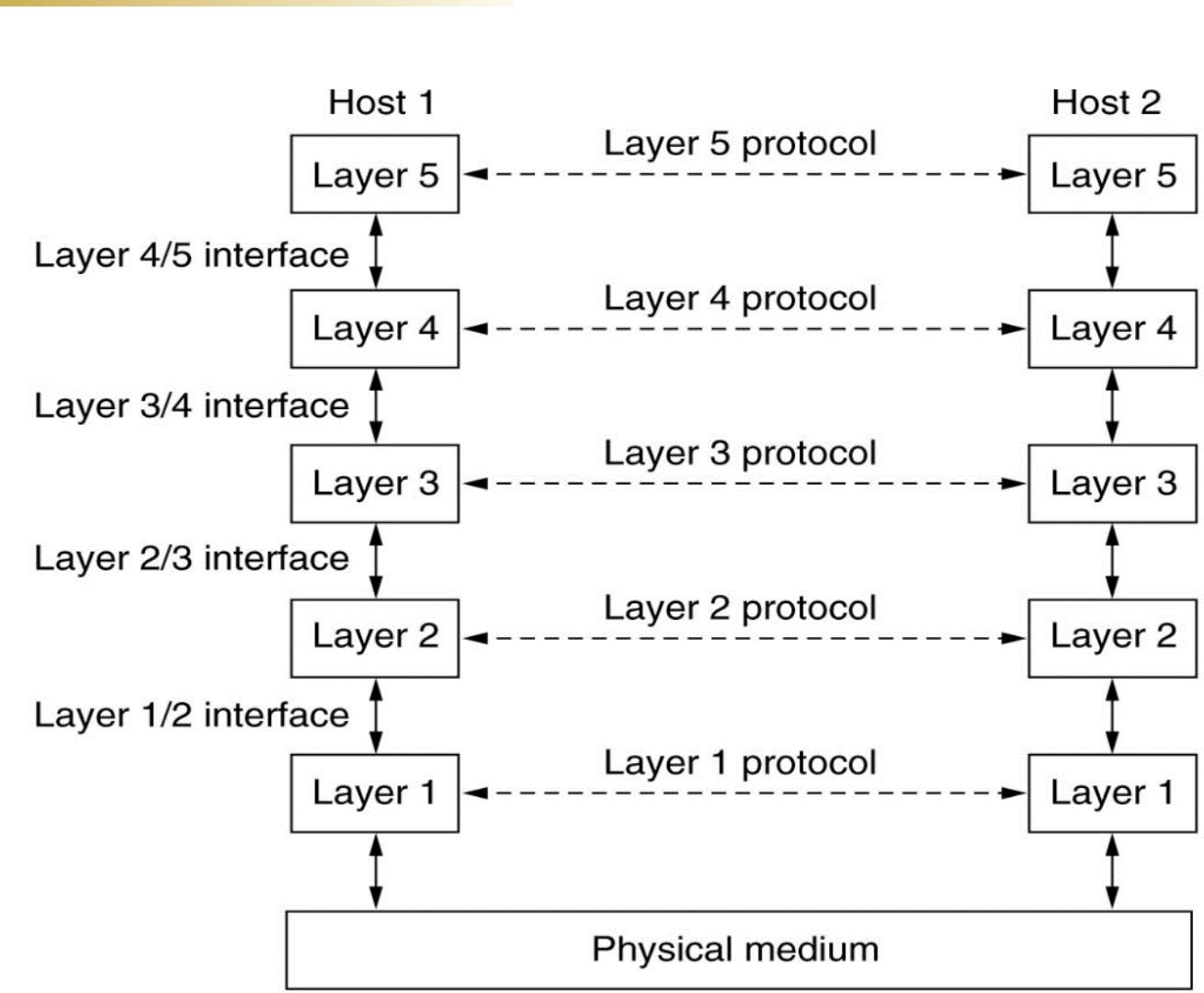


Layering simplifies design, implementation, and testing by partitioning overall communications process into parts



Why Layering?

- Each layer provides a service to the layer above
- Each layer operates according to a **protocol**
- Protocols can be changed without affecting other layers, higher or lower



Network **Protocol** ?

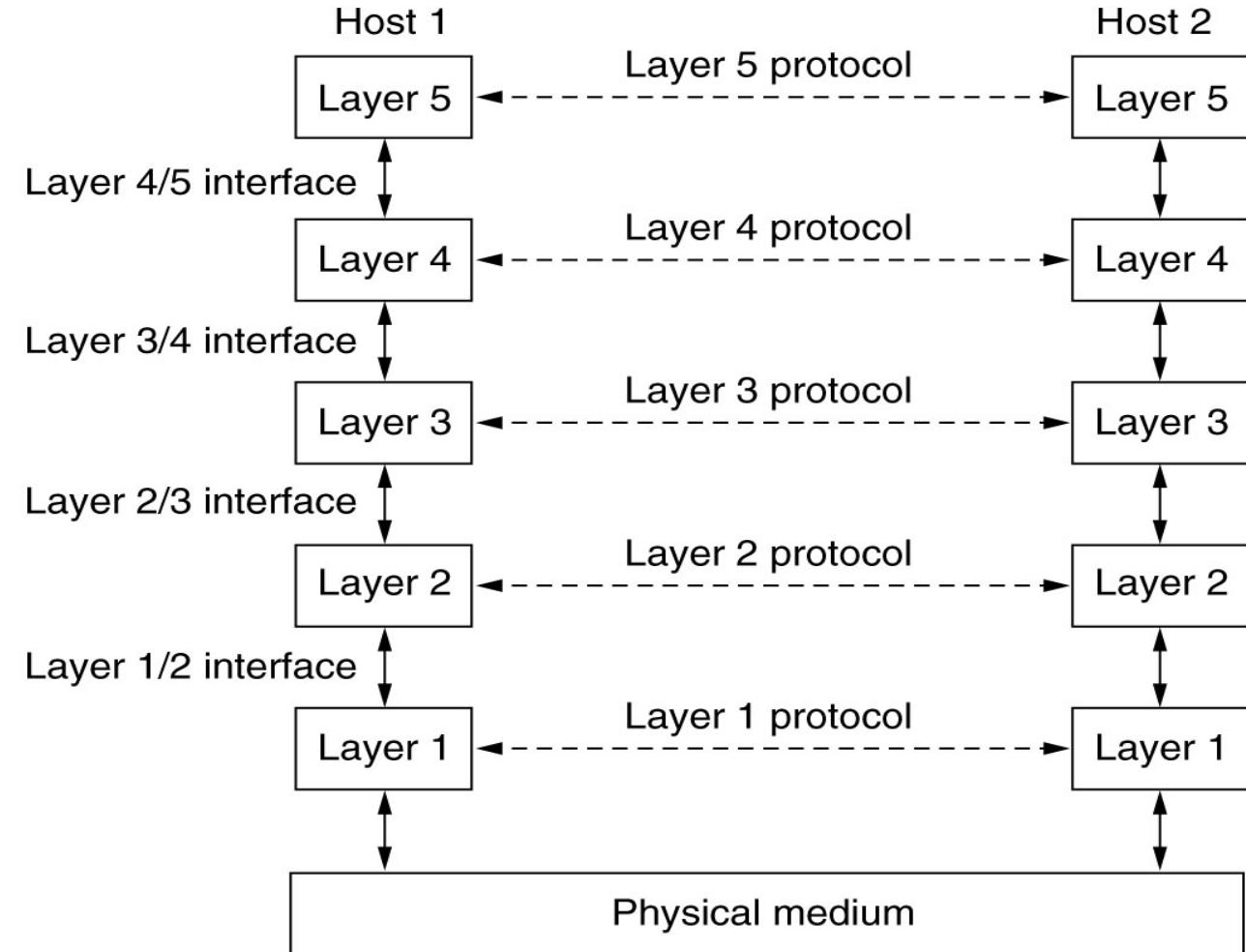
- **Set of rules** and conventions **governs** peer-to-peer **communication**

- **Network Architecture**

- The set of layers of protocols

- Protocol has specific **syntax**

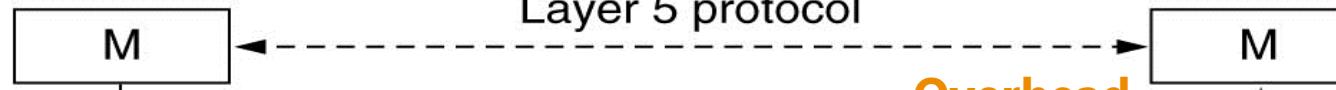
- ✓ Format of the data block
 - ✓ Uses control information for coordination
 - ✓ Sequencing
 - ✓ Speed matching



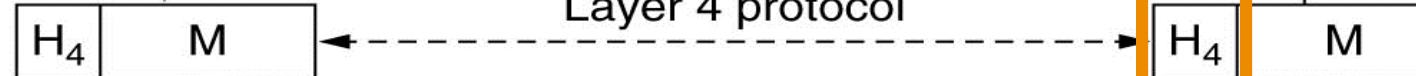
Layer Protocols – cont.

Layer

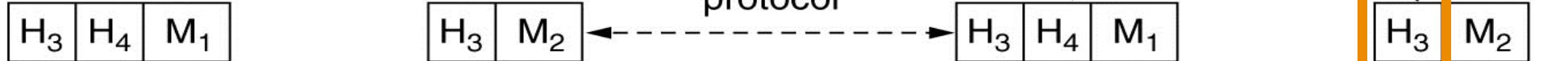
5



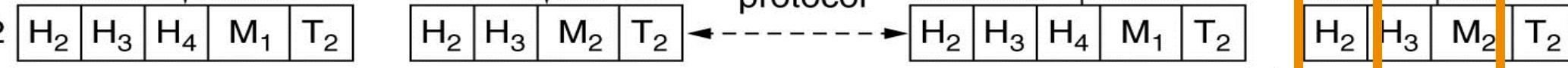
4



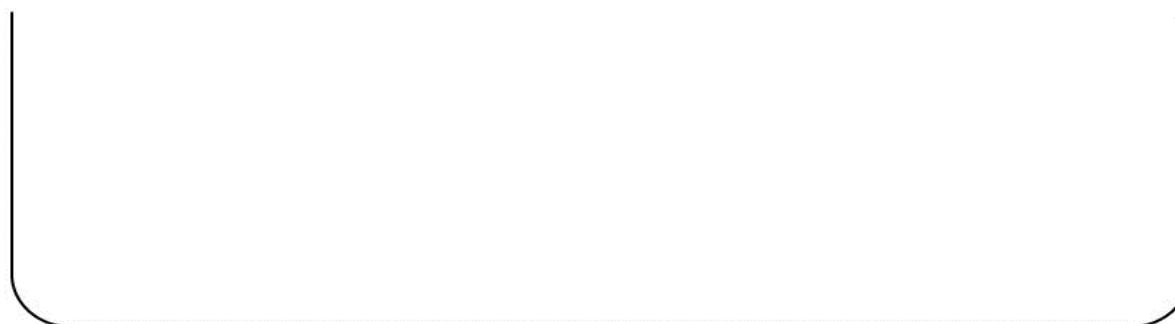
3



2



1

**Overhead / Header**

OSI Model

▪ Open System Interconnect

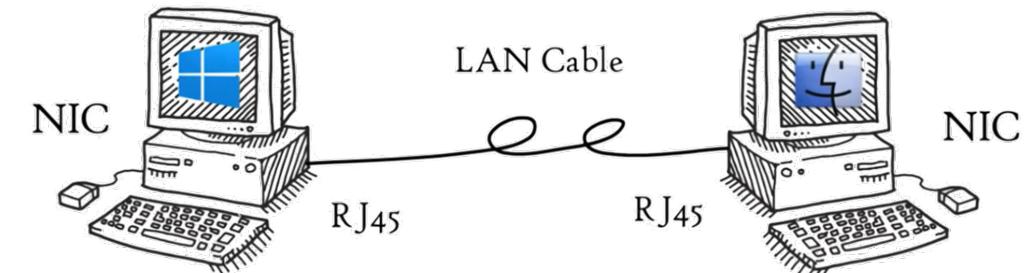
- Introduced by ISO in 1984

▪ Each layer:

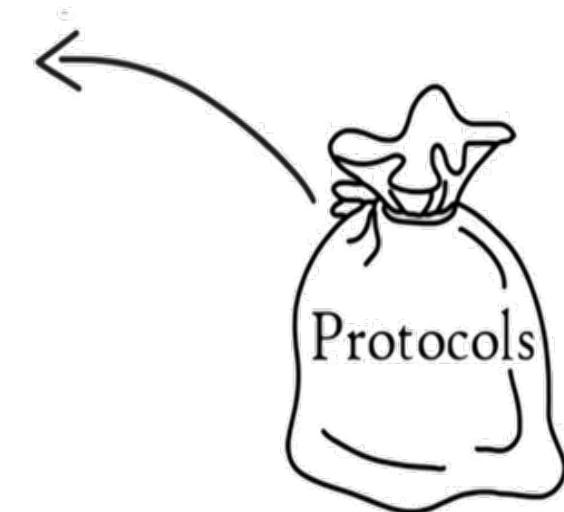
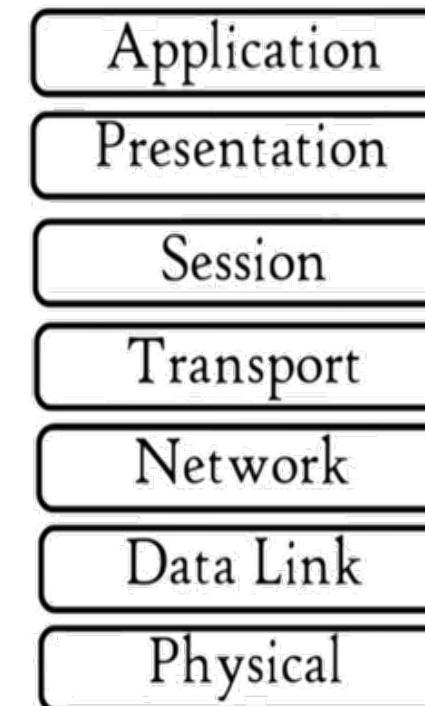
- Package of protocols

▪ 7 Layers

- Please – Physical
- Do - Data Link
- Not – Network
- Throw – Transport
- Sausage – Session
- Pizza – Presentation
- Away – Application

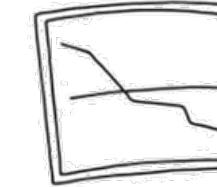
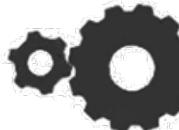


OSI Model



APPLICATION LAYER

Network Applications



HTTP HTTPS FTP
NFS FFTP DHCP
SNMP TELNET
POP3 IRC NNTP



Virtual
Terminals



File Transfer



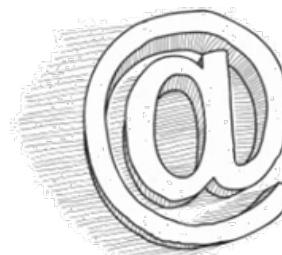
FTP

Web Surfing



HTTP/S

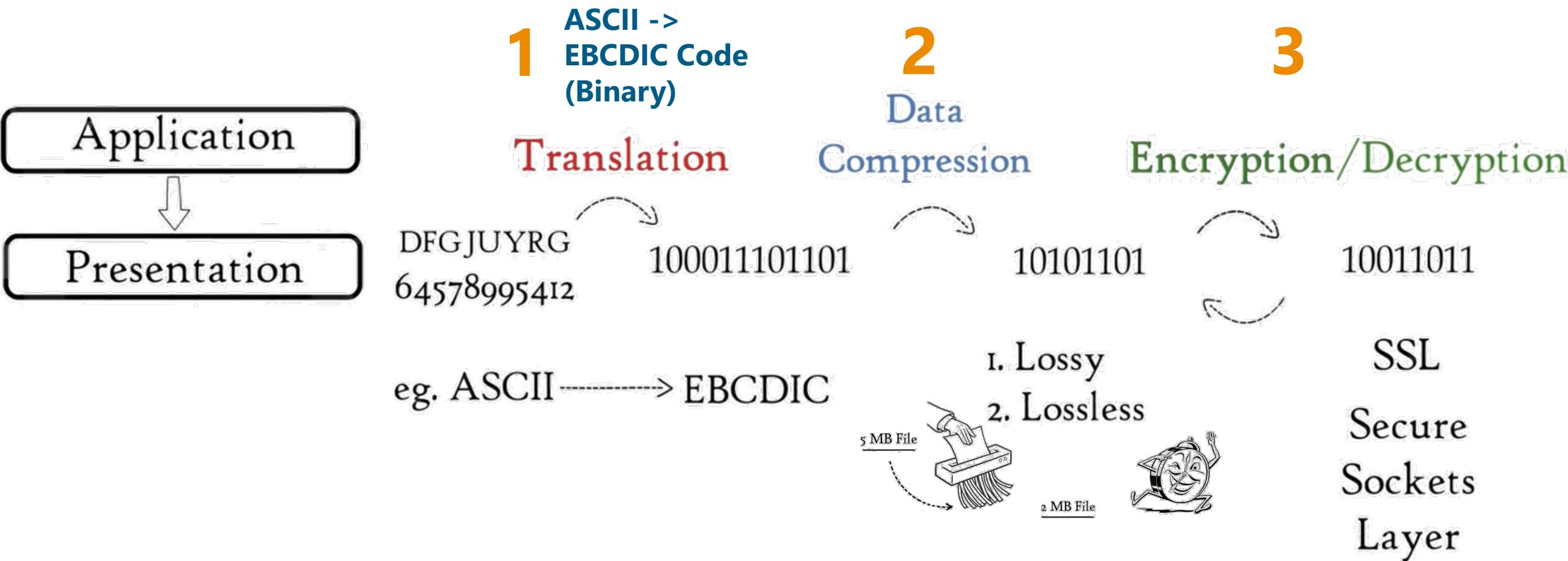
Emails



Telnet

Presentation Layer

- Used by Network Applications



Session Layer

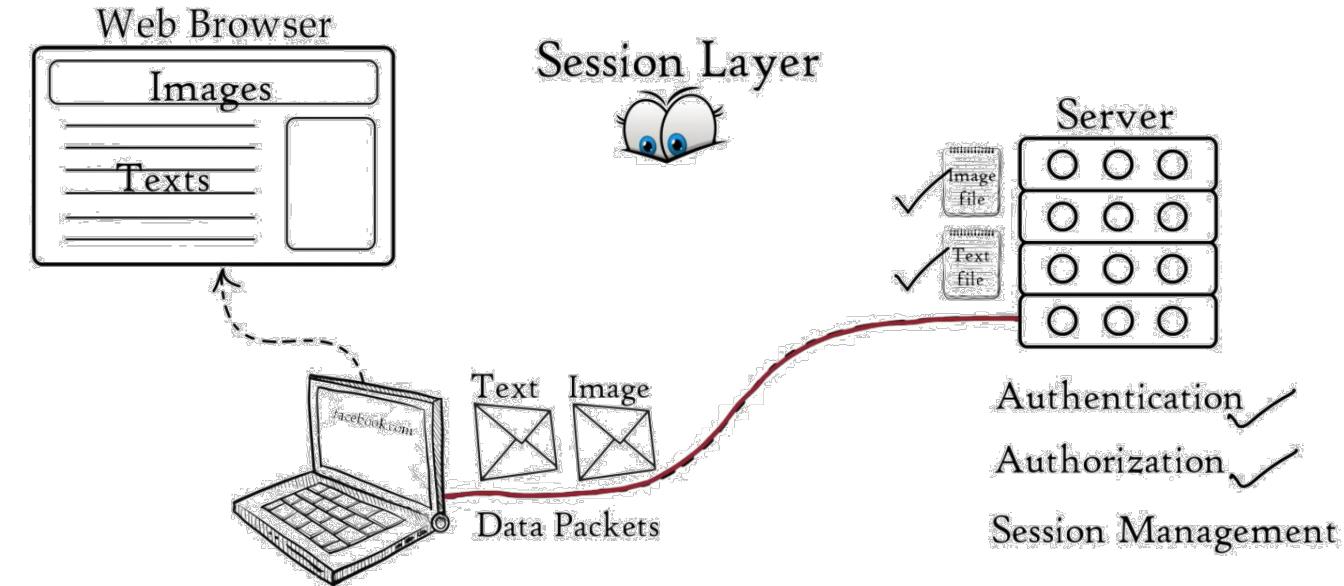
- Setting up and managing connections
- Session Layer Helpers: APIs, NETBIOS
 - Helps applications to communicate with each other

▪ **Session Management**

- Authentication, Authorization

▪ **For example,**

- ✓ Web browser performs:
 - Session maintenance across different tabs



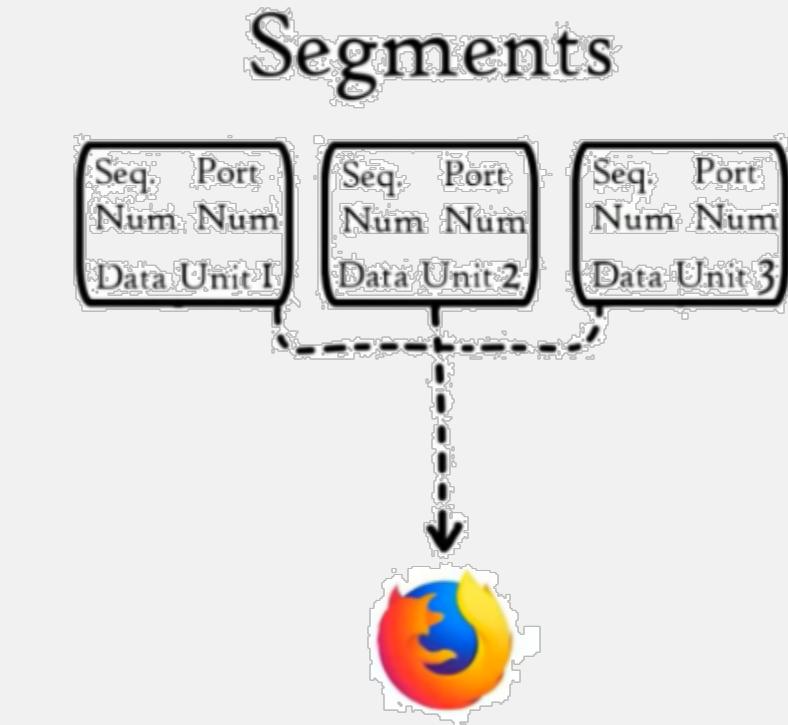
Transport Layer

- Send data from **one application** in a host
to another application in another host



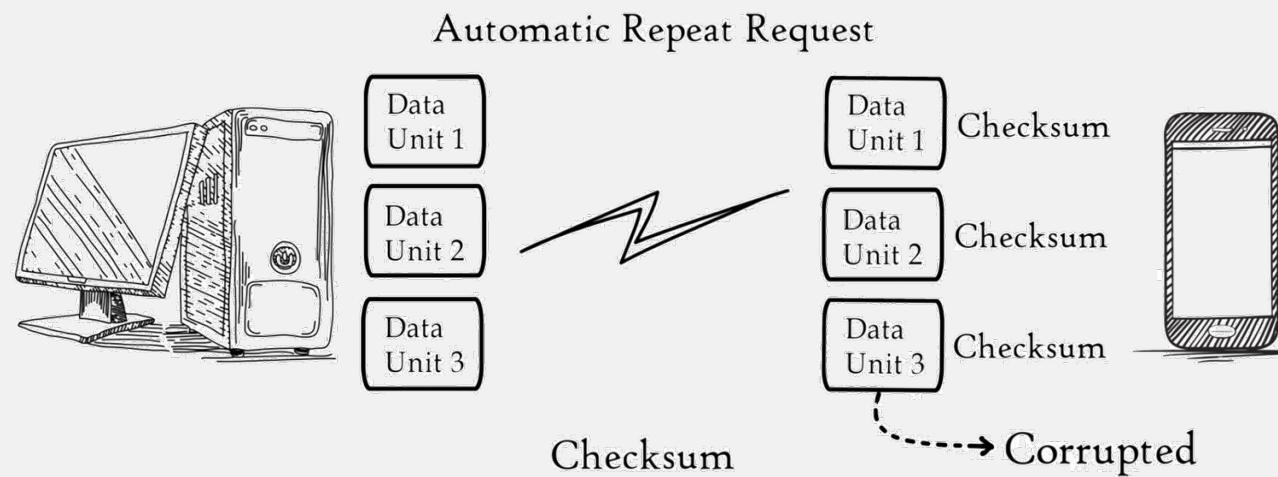
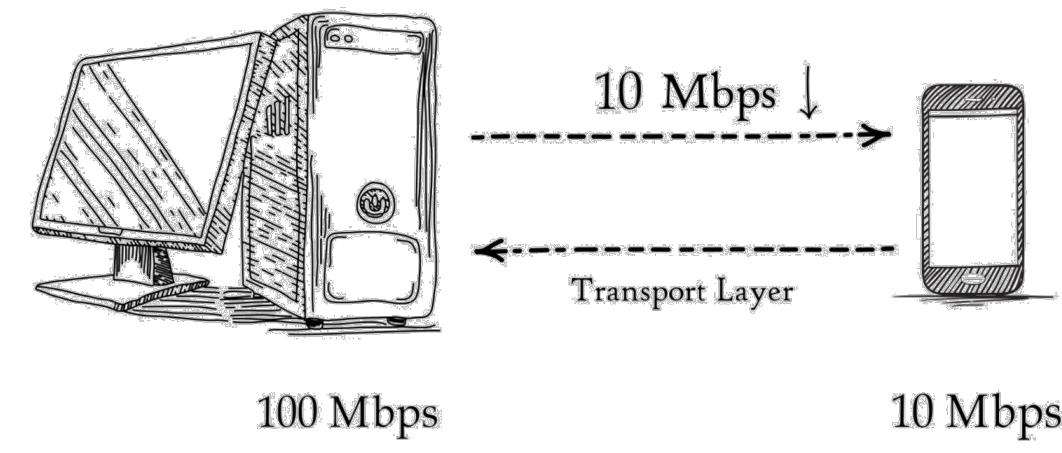
1. Segmentation

- ✓ Data -> segments
- ✓ Segment:
 1. Port number
 2. Seq number
 3. Payload (data)



2. Flow Control

- ✓ Control the amount of data being transmitted

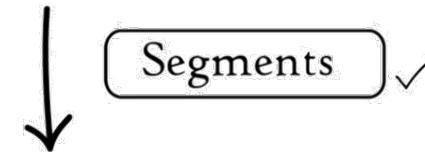


3. Error Control

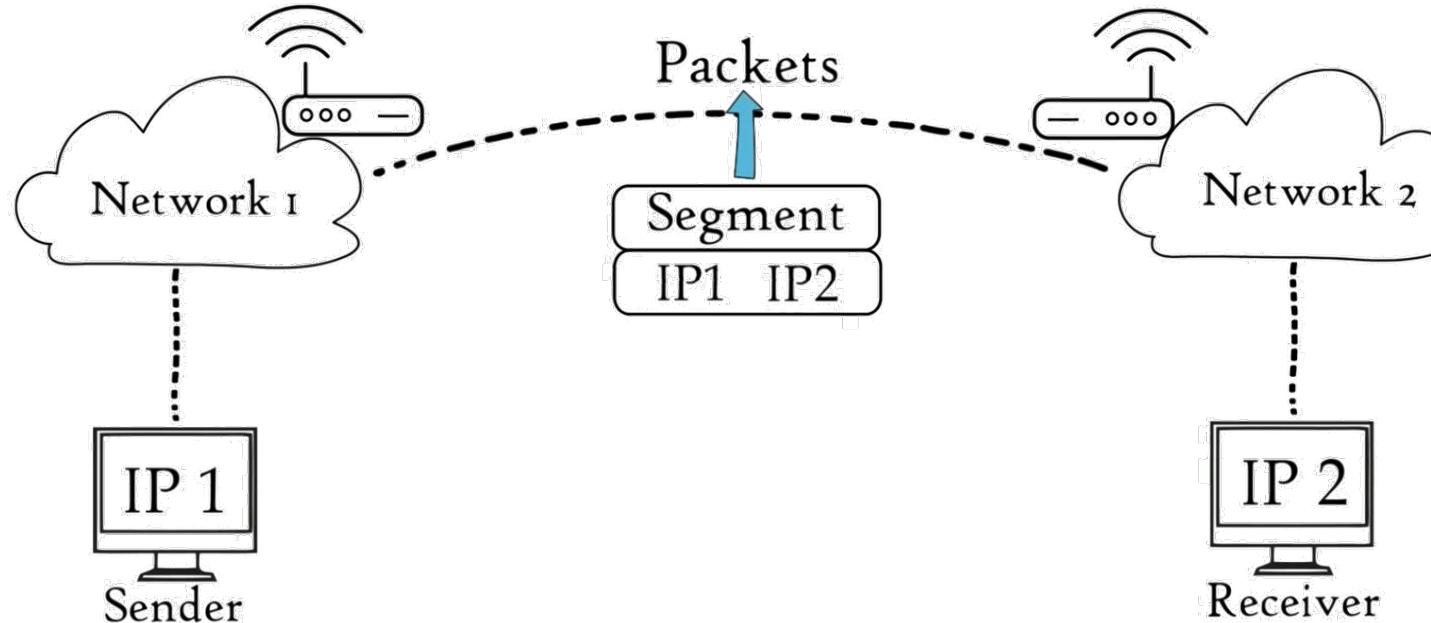
- ✓ Detect errors in data-units
- ✓ Retransmit missing data-units

Network Layer

Transport Layer



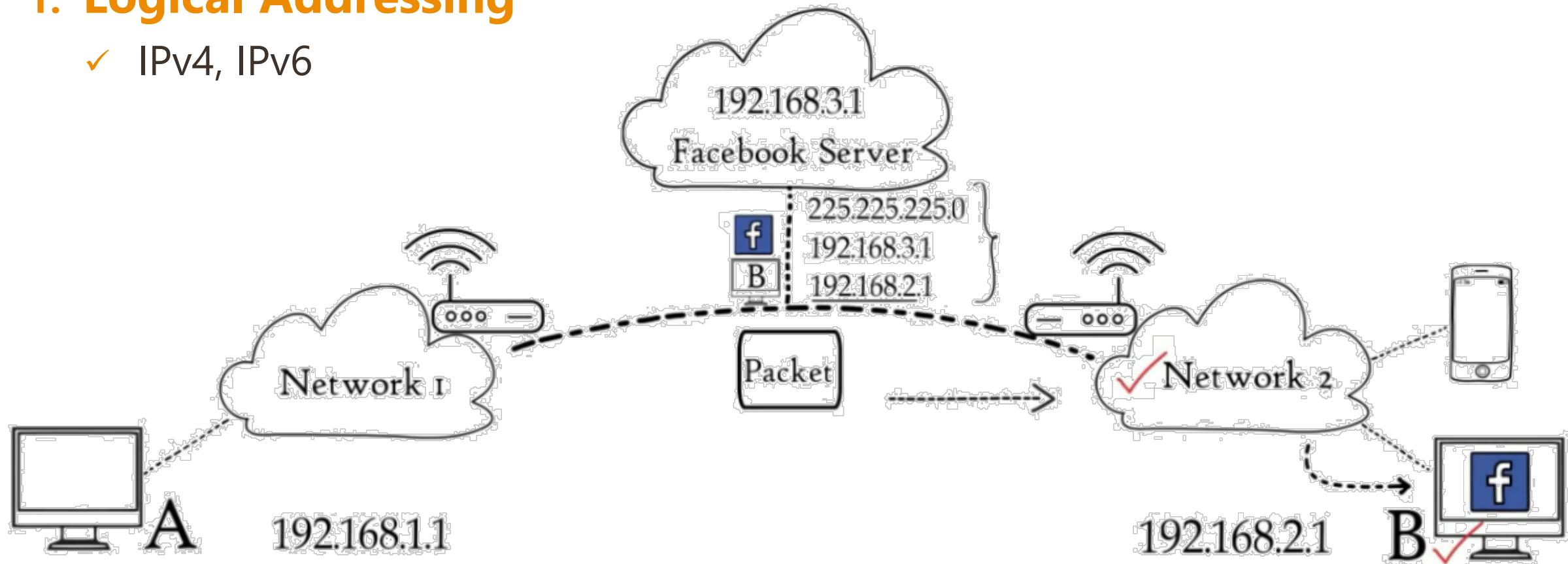
Network Layer



Network Layer – cont.

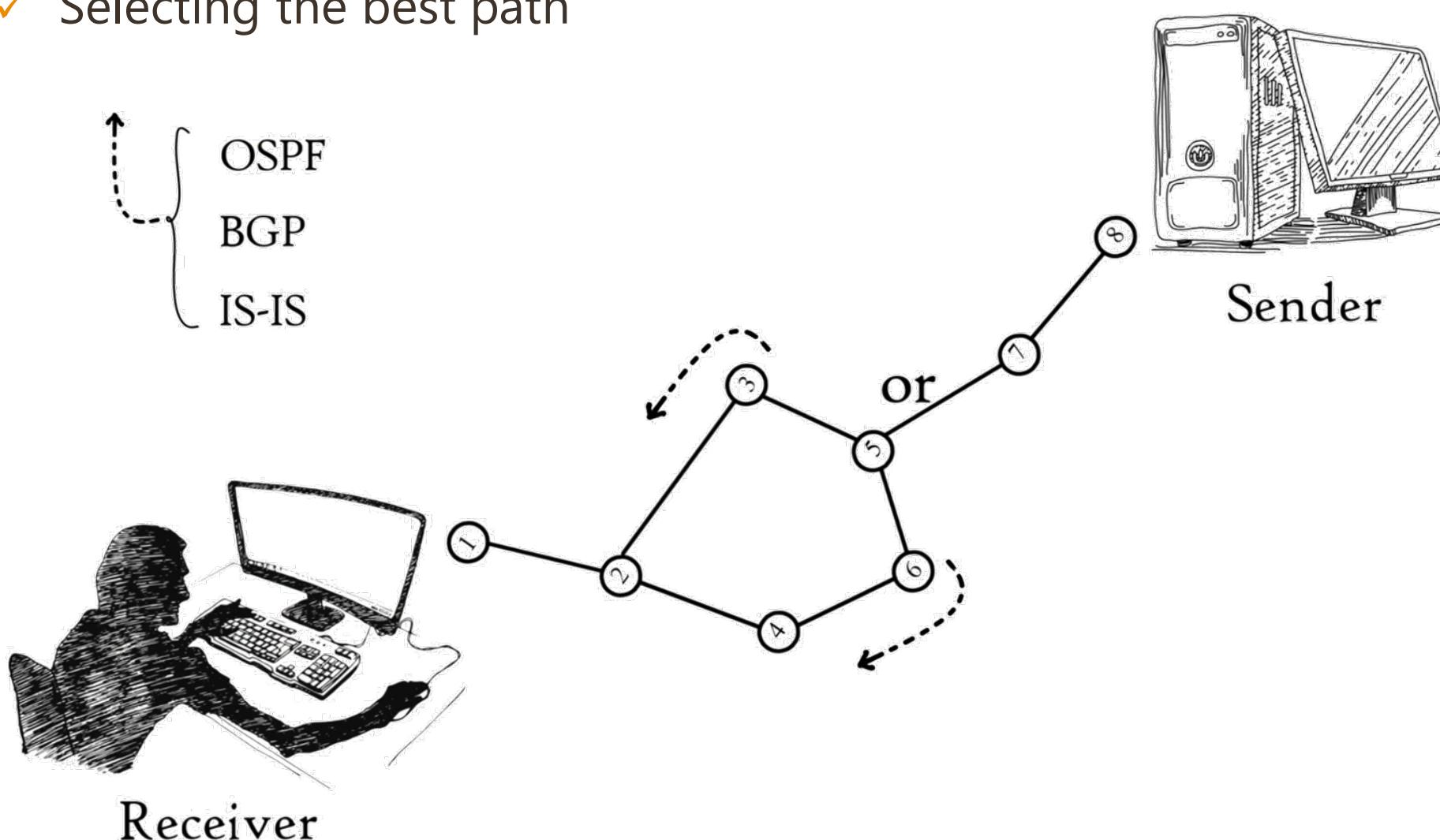
1. Logical Addressing

- ✓ IPv4, IPv6



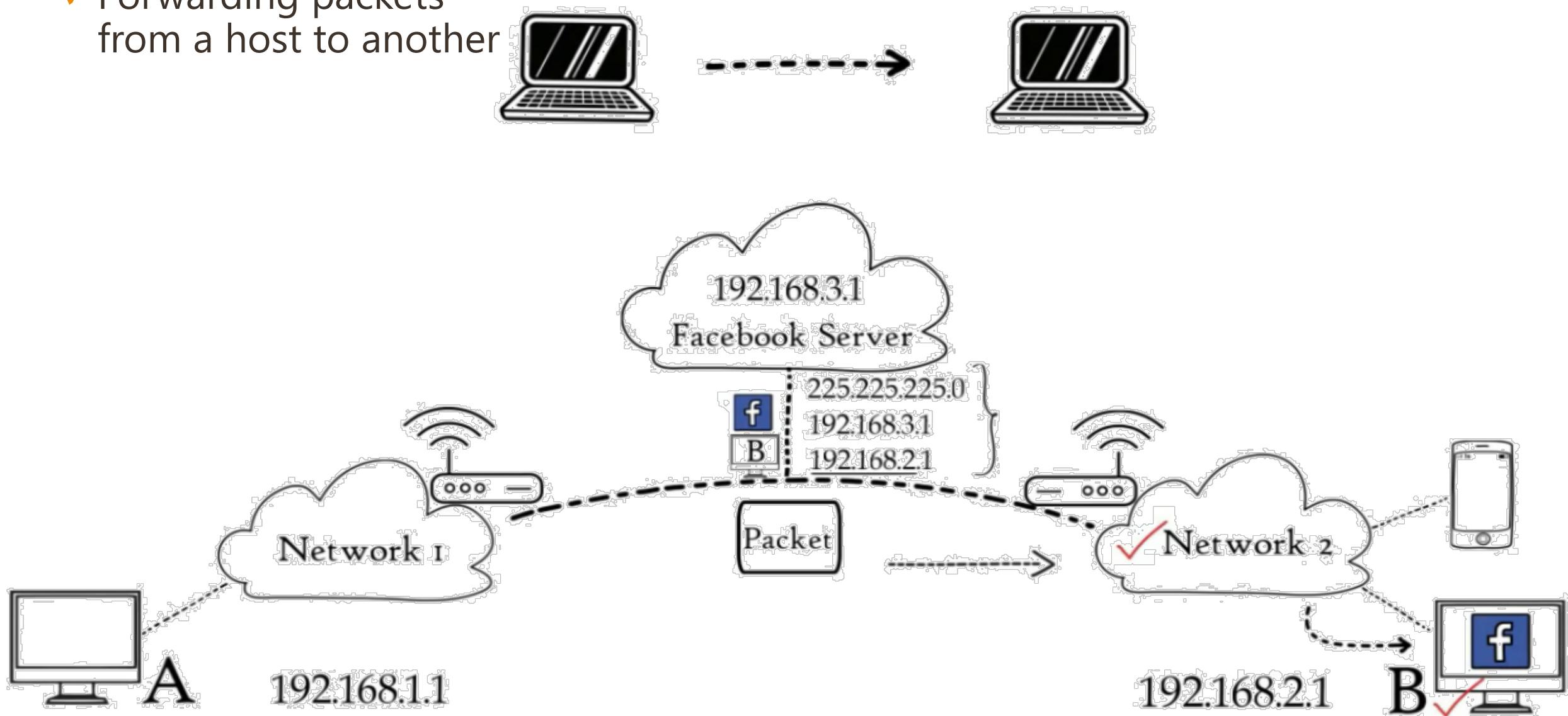
2. Routing

- ✓ Selecting the best path



2. Routing

- ✓ Forwarding packets from a host to another



Data Link Layer

▪ Physical Addressing

- ✓ MAC (Media Access Control) Address
- ✓ Hardcoded by manufacturer to NIC

Two Sub Layers

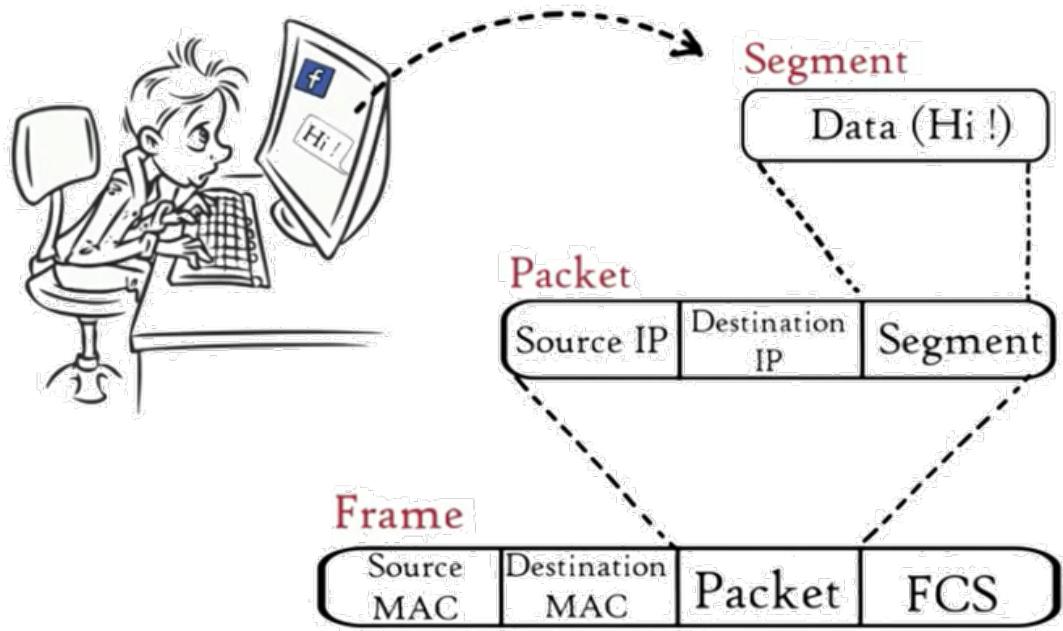
- **Logical Link Control (LLC):**

- ✓ *Error Detection and Correction, etc.*

- **Media Access Control (MAC)**

- ✓ *Framing*
 - ✓ *How the media access is controlled and shared*





TRANSPORT LAYER

NETWORK LAYER

DATA LINK LAYER

APPLICATION
LAYER

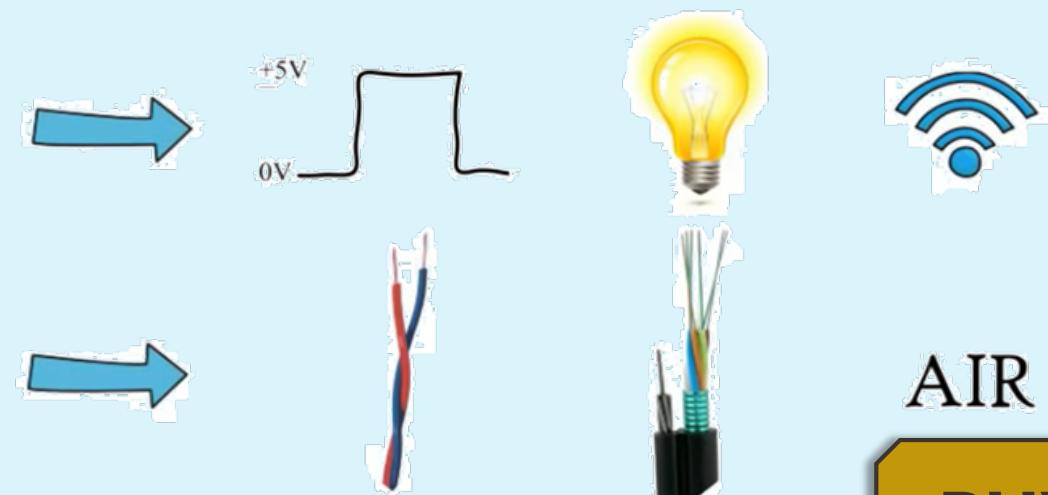
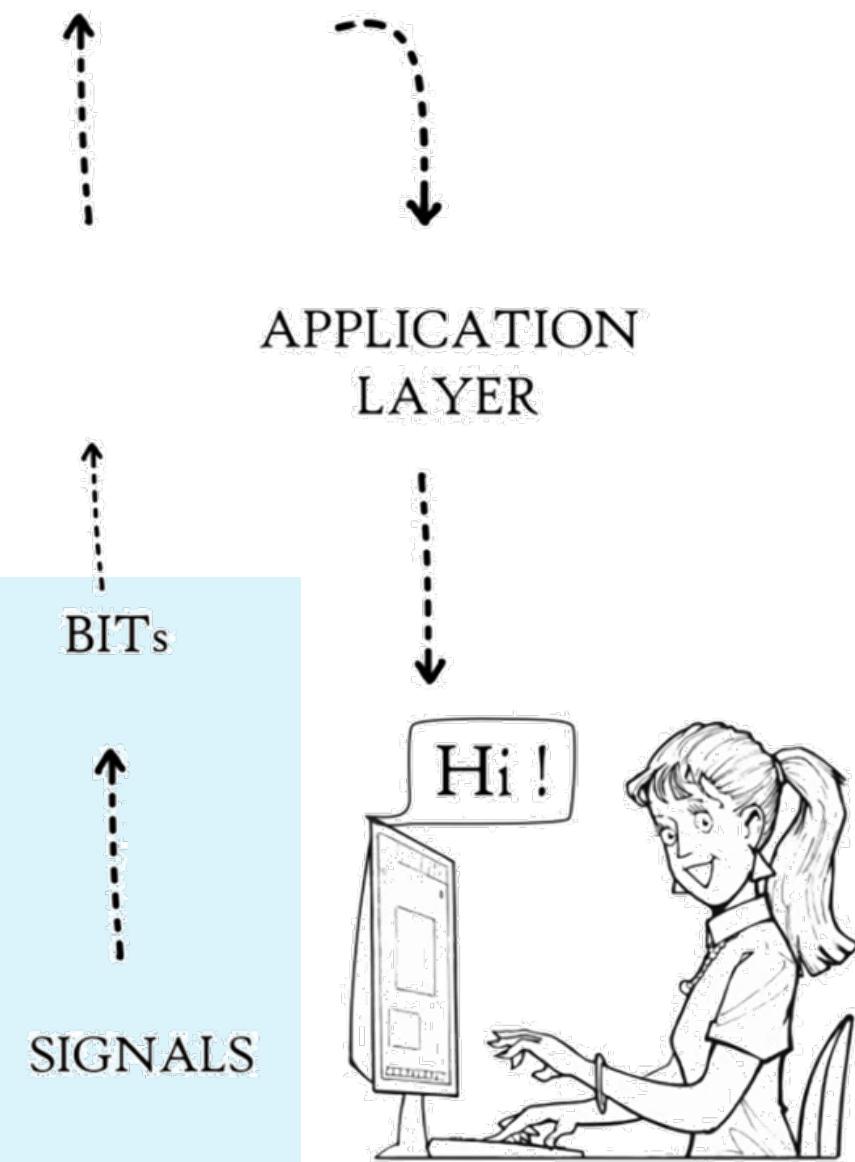
1001011101100101110110111011

BITS

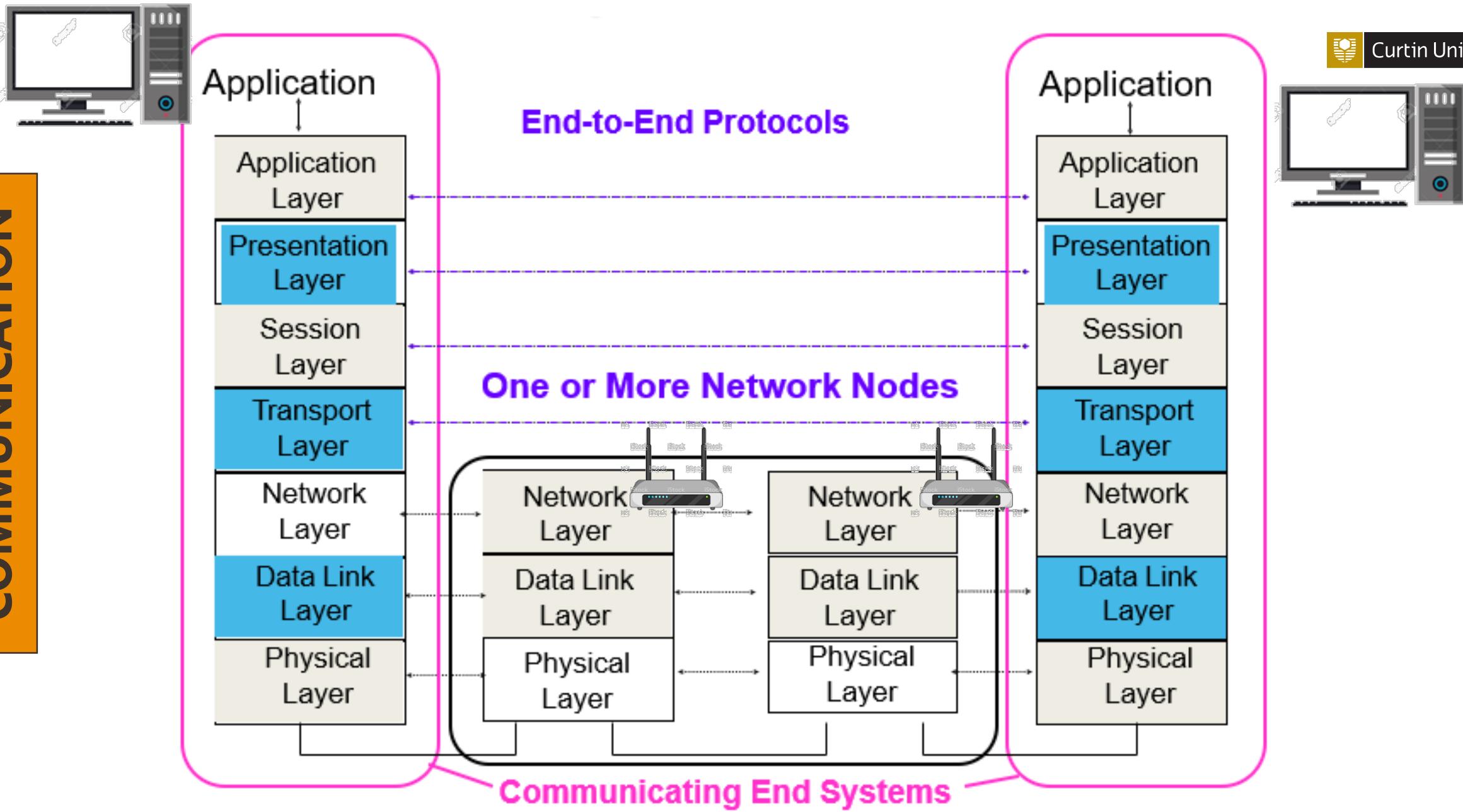
BITS

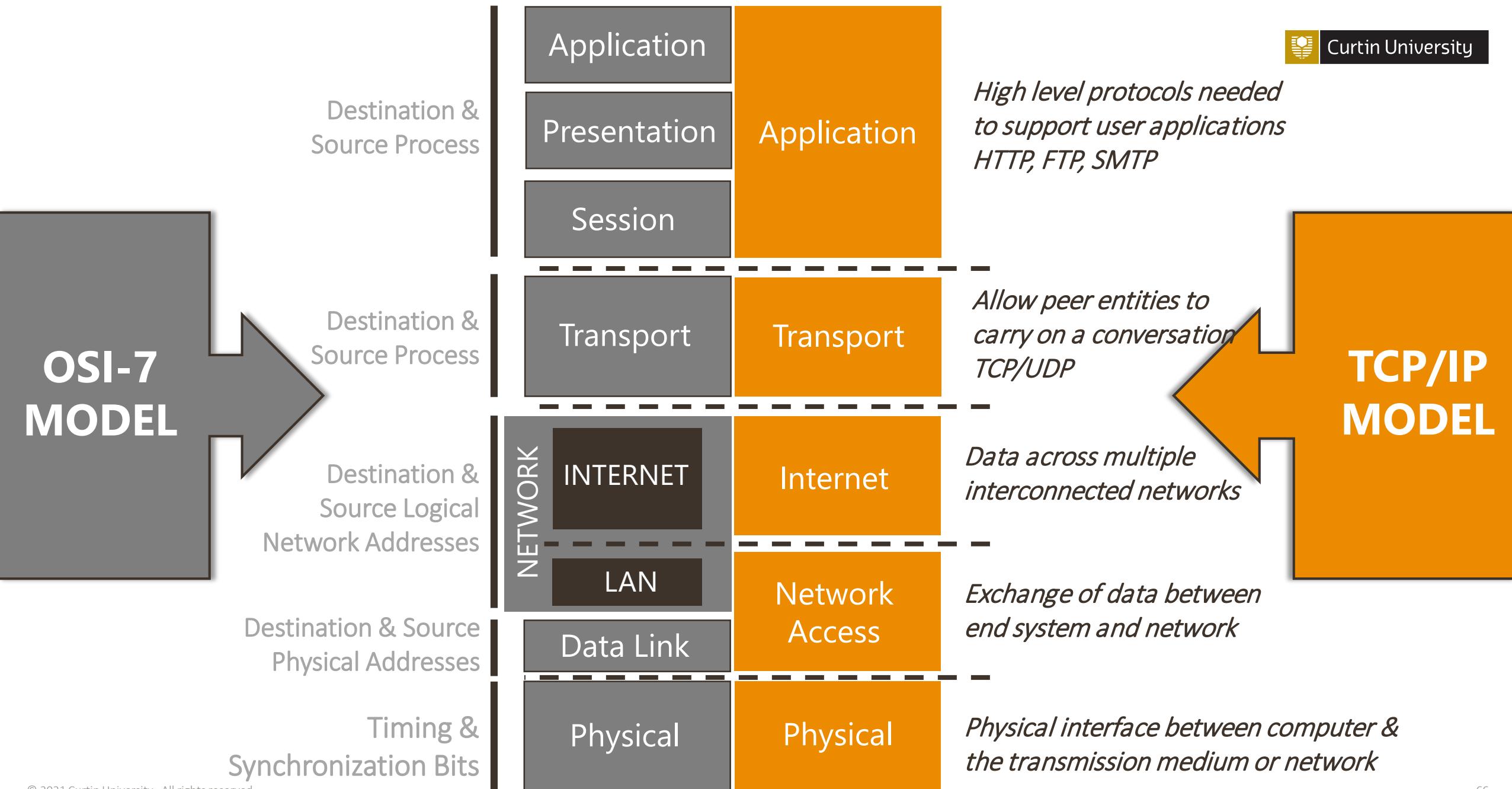
SIGNALS

SIGNALS

**PHYSICAL LAYER**

COMMUNICATION

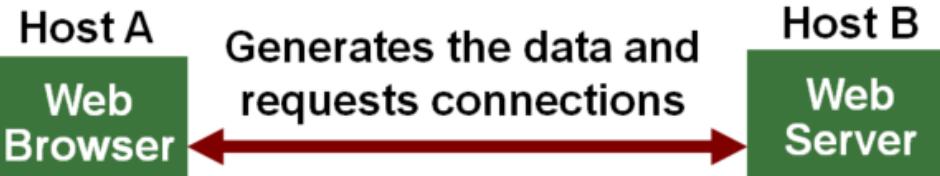






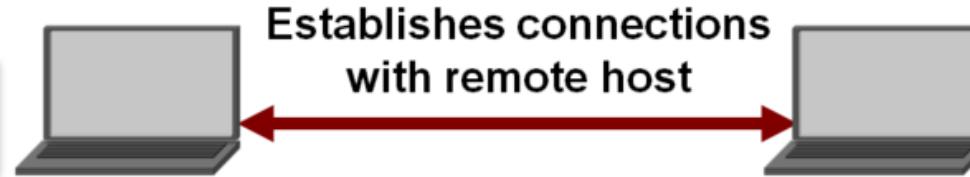
5 Application Layer

The Application layer is the group of applications requiring network communications.



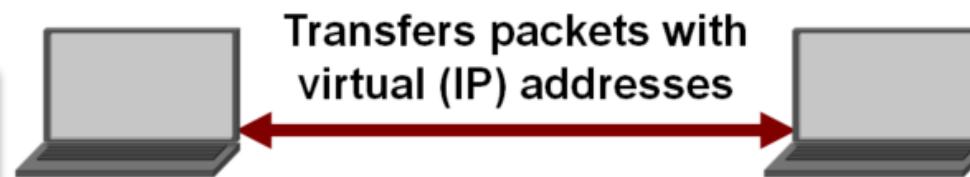
4 Transport Layer (TCP/UDP)

The Transport layer establishes the connection between applications on different hosts.



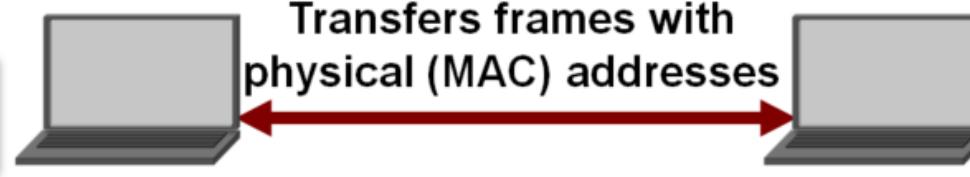
3 Network Layer (IP)

The Network layer is responsible for creating the packets that move across the network.



2 Data Link Layer (MAC)

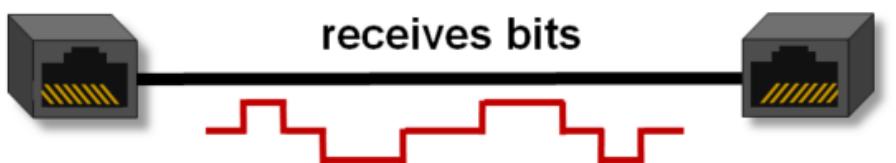
The Data Link layer is responsible for creating the frames that move across the network.



TCP/IP MODEL

1 Physical Layer

The Physical layer is the transceiver that drives the signals on the network.





▪ **Introduction to Networks**

- What is a network?
- Data network elements
- Transmission Technology
 - Point-2-point (unicast)
 - Multi-point (multicast, broadcast)
- Scales of network
 - LAN, MAN, WAN
 - WLAN, WMAN, WWAN, WPAN

▪ **Classification of Networks**

- Circuit-switched networks
- Packet-switched networks
 - Datagram networks
 - Virtual Circuit networks

▪ **Common Networking Technologies**

- Intro. to Telephone Networks (PSTN), Internet, ATM
- Internet

▪ **Layers, Services & Protocols**

- Layering? Services? Protocol?
- Intro to OSI-7-layer model
 - Application Layer Elements / Functions
 - Presentation Layer Elements / Functions
 - Session Layer Elements / Functions
 - Transport Layer Elements / Functions
 - Network Layer Elements / Functions
 - Data Link layer Elements / Functions
 - Physical Layer Elements / Functions
- Intro to TCP/IP 5-layer model
- OSI 7-layer model vs TCP/IP 5-layer model

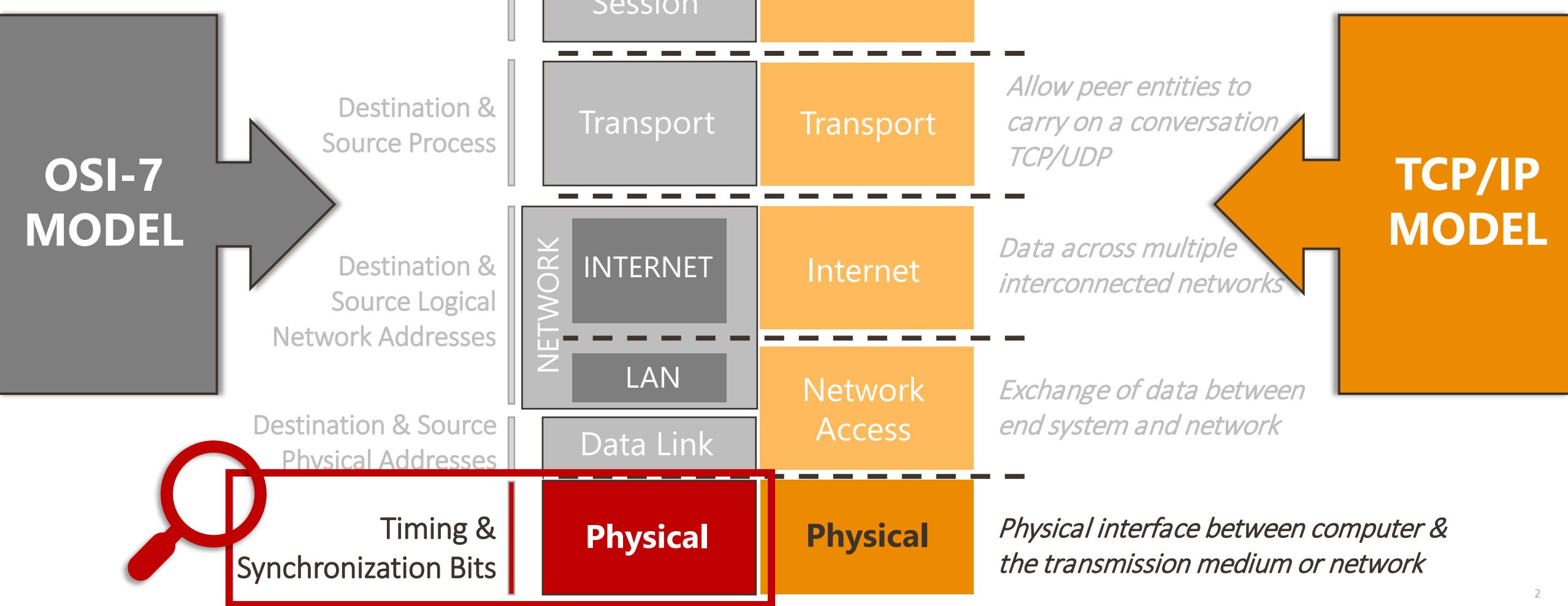
THANK YOU

Make tomorrow better.

Physical Layer

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 02

Semester 1, 2021



Physical Layer

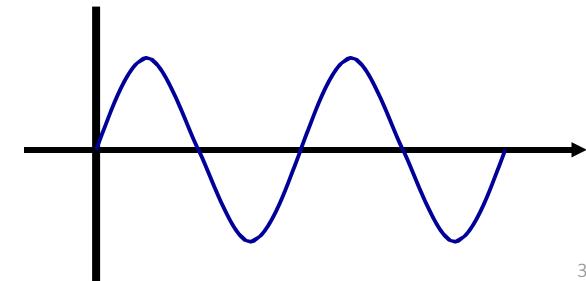


Defines physical characteristics of
interfaces & mediums

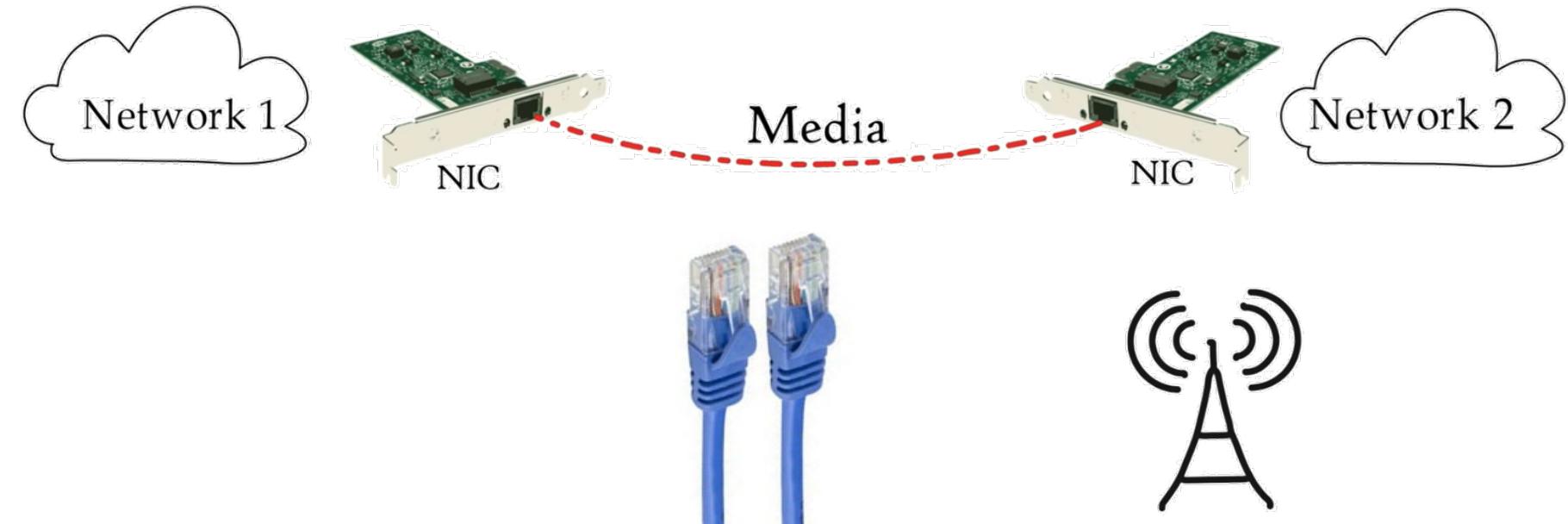
Mechanical/electrical specifications



Move **data** as electromagnetic signals



Physical Layer



- Primarily consists of hardware (unless virtual)
- Provides the basic communication channel that two network devices (e.g. computers) used to send and receive messages
- Provides transmission & reception hardware

Physical Layer: Services

- Transmits bits over a physical link between devices
- Encodes/Decodes **bits into/from** \Leftrightarrow **physical signals**
- Most common **Analog signals** (*Electromagnetic waves*)
 - ✓ Electrical pulses over wire
 - ✓ Radio signals through the air
 - ✓ Pulses of light through fiber optic cable



Signals

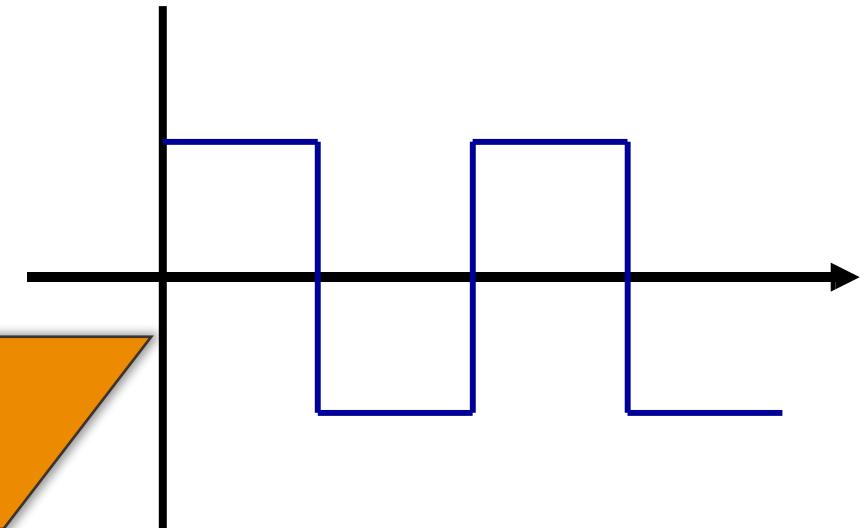
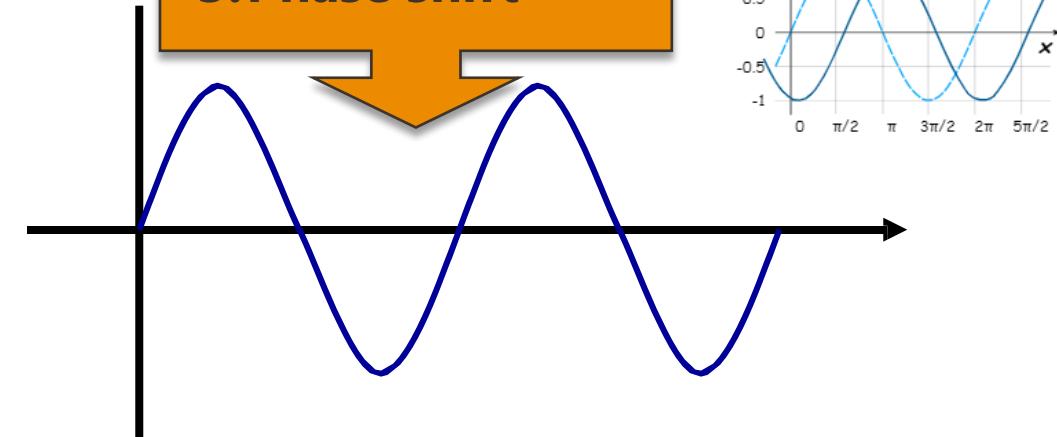
- Analog signals
- Digital signals
- Signal attenuation and amplification

Signals

- Analog
 - ✓ Continuously varying voltage

- Digital
 - ✓ Sequences of specified **voltage levels** (high, low)

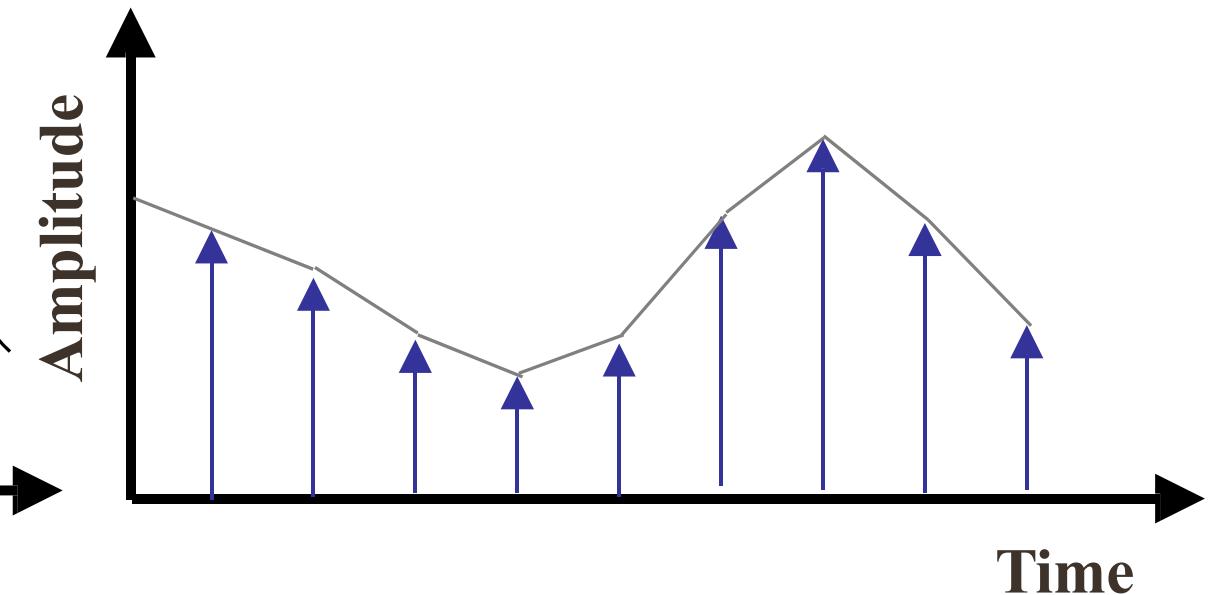
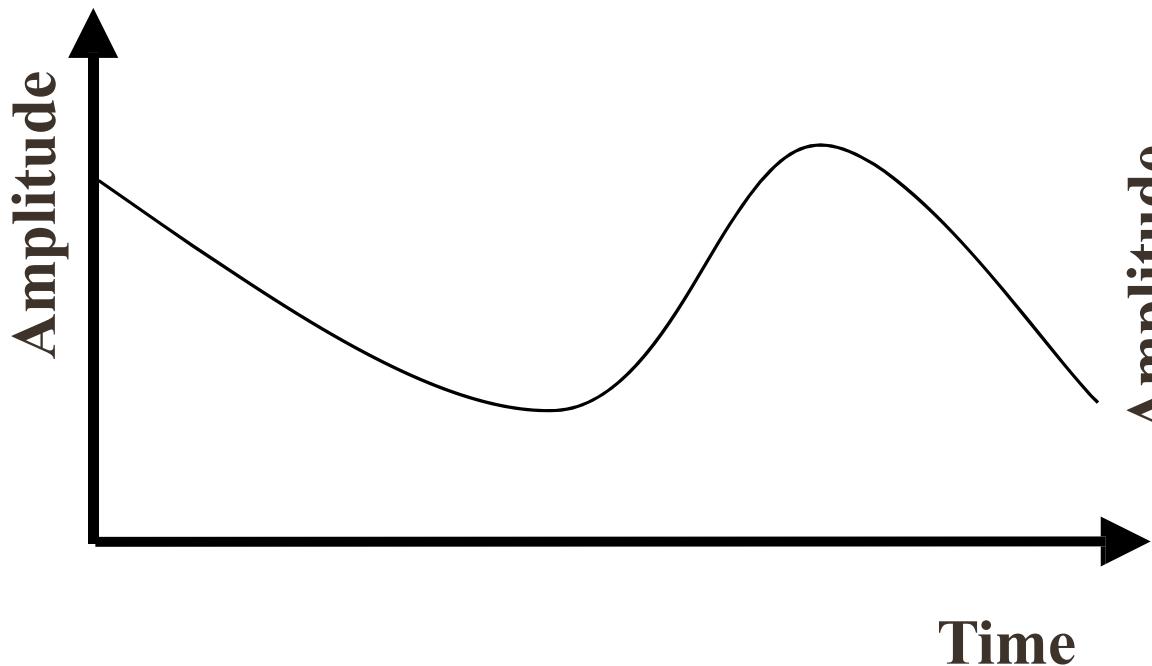
1. Frequency Hertz (cycles/sec)
2. Amplitude
3. Phase shift



Digital Encoding
Digital \leftrightarrow Analogue

Analog to Digital

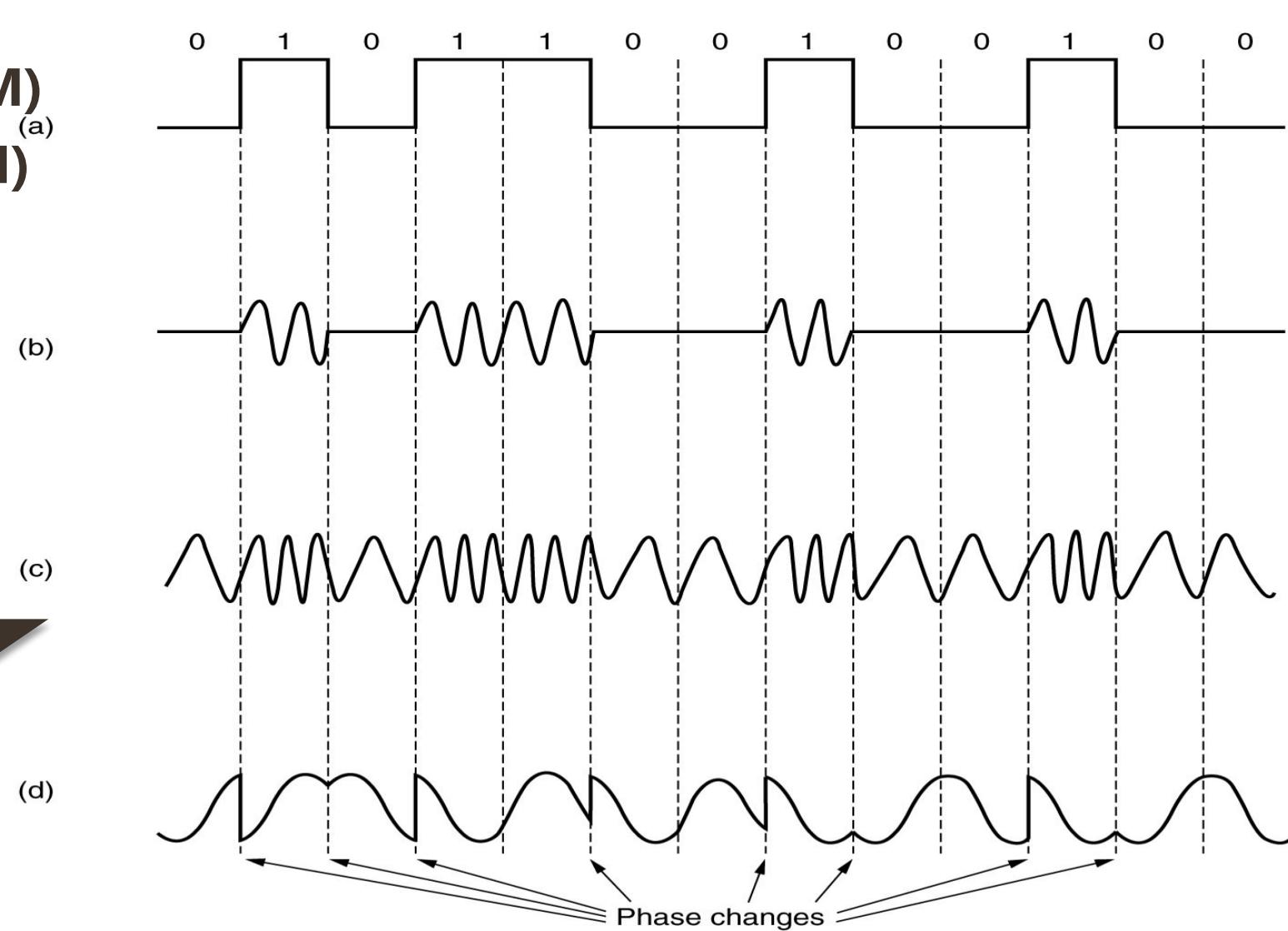
- Sampling



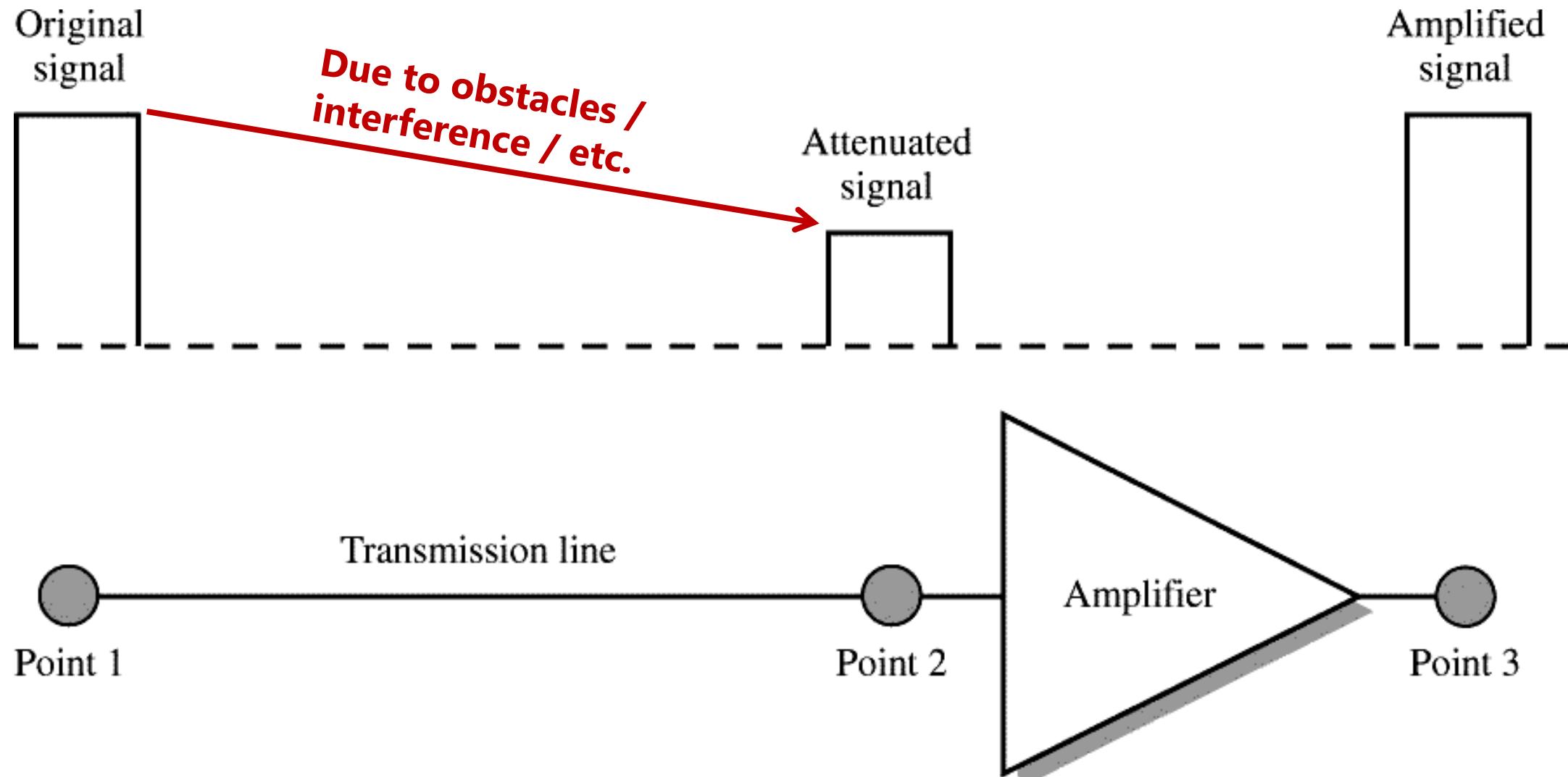
Digital to Analog

- Amplitude Modulation (**AM**)^(a)
- Frequency Modulation (**FM**)
- Phase Modulation (**PM**)

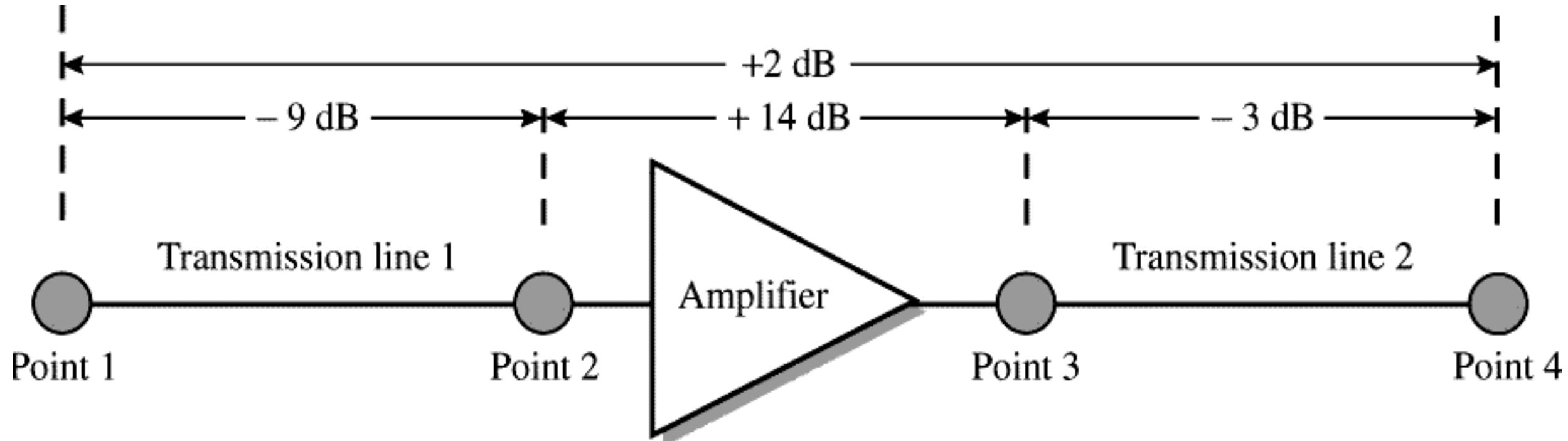
Assign a group of one or more-bit values to a particular analog signal.



Signal Attenuation & Amplification



Signal Attenuation & Amplification



- Positive or negative **dB** represents the **system loss/gain between two points**
- dB level at point 4 => $(-9) + (14) + (-3) = +2\text{dB}$

Signal Attenuation & Amplification

- The decibel (**dB**) is a logarithmic unit used to measure sound level. It is also widely used in electronics, signals and communication.
- The dB is a logarithmic way of describing a **ratio**.
- Suppose we have two signals, the first one with power p_1 , and the second with power p_2 . Using the decibel unit, the **difference in signal power**, between the two signals is defined to be:

$$\text{power_ratio_in_db} = 10 \log^{p_2/P_1}$$

e.g., $p_2 = 5 p_1$, the difference in dB is $10 \log (p_2/p_1) = 10 \log 5 \approx 7$

$p_2 = 0.1 p_1$, the difference in dB is $10 \log (p_2/p_1) = 10 \log 0.1 = -10$



Digital Encoding

- Bit Encoding
- Manchester Encoding
- Differential Manchester Encoding
- MLT-3

Digital Encoding

Bit stream



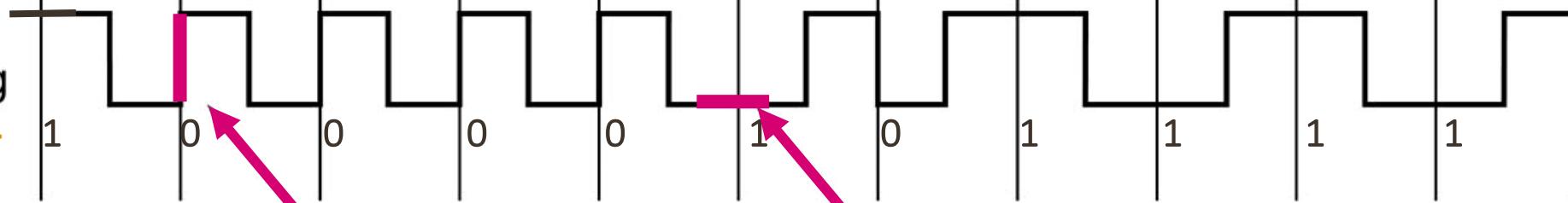
(a) Binary encoding

1: 0:

(b) Manchester encoding

1: 0:

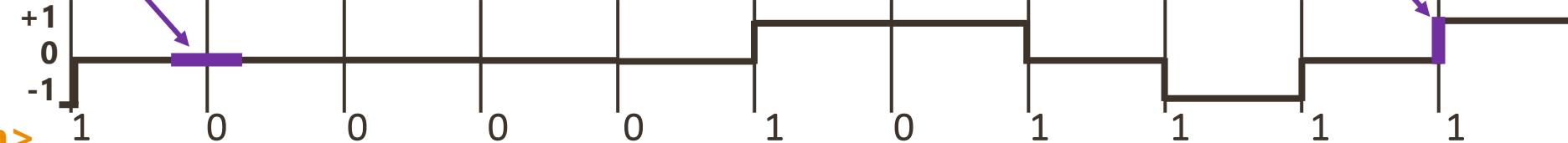
(c) Differential Manchester encoding

1: <no transition>
0: <transition>Lack of transition here
indicates 0Transition here
indicates 0Lack of transition here
indicates 1transition here
indicates 1

(d) MLT-3 encoding

1: <transition>

0: <no transition>



Encoding – cont.

▪ [Differential] Manchester Encoding

- ✓ Self-Ticking Signal (*no need to synchronize the sender/receiver clocks*)
- ✓ Robust to noise

▪ MLT-3

- ✓ 3 Levels of power (+1, 0, -1)
- ✓ Emits less electromagnetic interference
- ✓ Requires less bandwidth than other encoding techniques
- ✓ Used in **Fast Ethernet**





Medium Capacity

- Bandwidth, Speed, Lag, Throughput
- Multiplexing: FDM
- Multiplexing: TDM

Medium Capacity

- Measured in bits per seconds (**bps**)

- Even a perfect channel has a finite transmission capacity

- **Bandwidth:**

- Maximum amount of data transfer per second (**capacity**)

Gigabit Ethernet – Bandwidth: 1Gbps

- The range of frequencies used to transmit signals without being strongly attenuated (**range**)

Bandwidth (capacity) for single mode fiber 10Gbps with a **Bandwidth (range)** of 20Ghz

Medium Capacity – cont.

- No transmission facility can transmit without some **degradation**...

Loss of some frequency components

- Usually, frequencies from $0-f_{\max}$ are transmitted, and **above f_{\max}** are **attenuated**

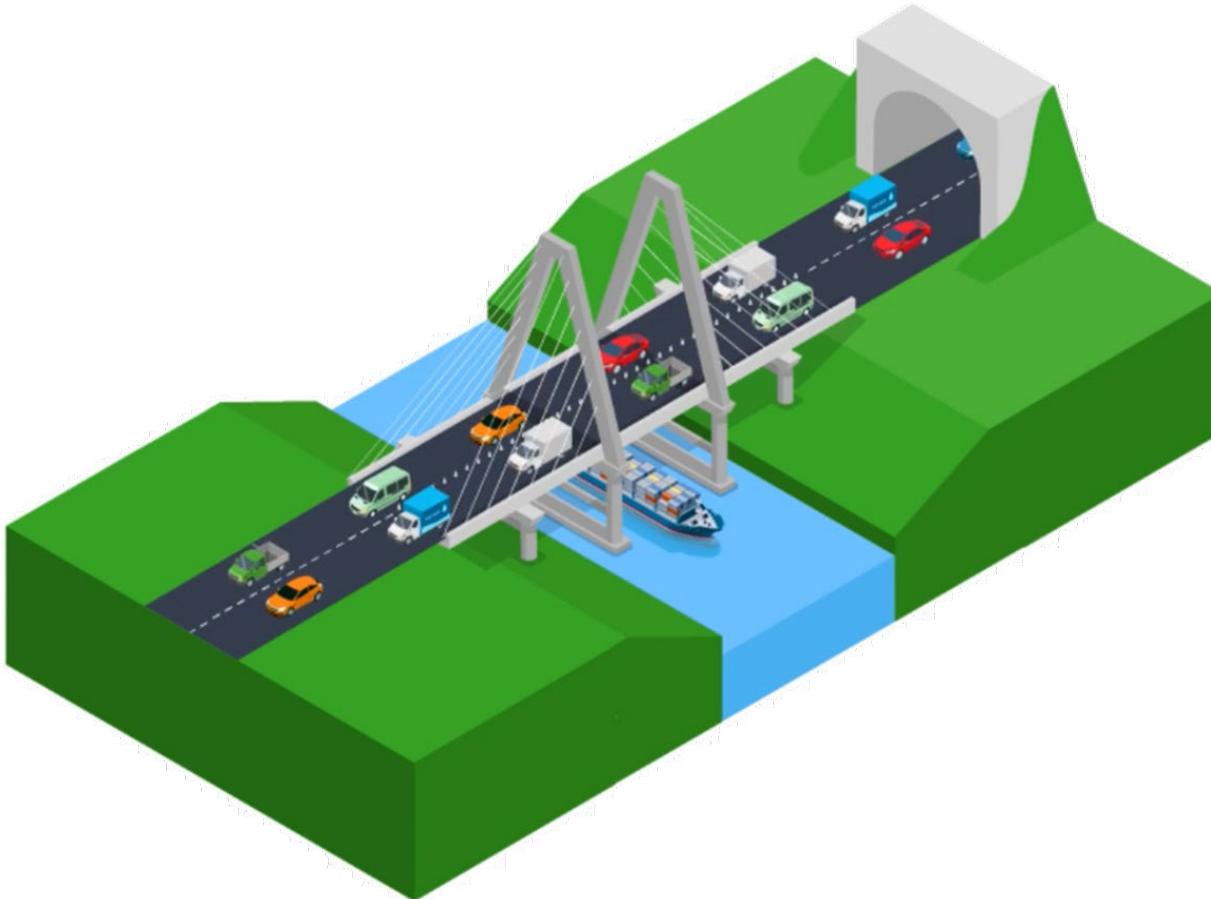
Analog voice signal: 300-3000Hz

Speed $\propto 1/\text{Distance}$

- **Wi-Fi 2.4GHz**
 - ✓ long distance
 - ✓ low speed
- **Wi-Fi 5GHz**
 - ✓ shorter distance
 - ✓ high speed

Bandwidth vs Speed

- **Bandwidth:** max amount of vehicles passing bridge/hour



- **Speed:** End-to-end flow

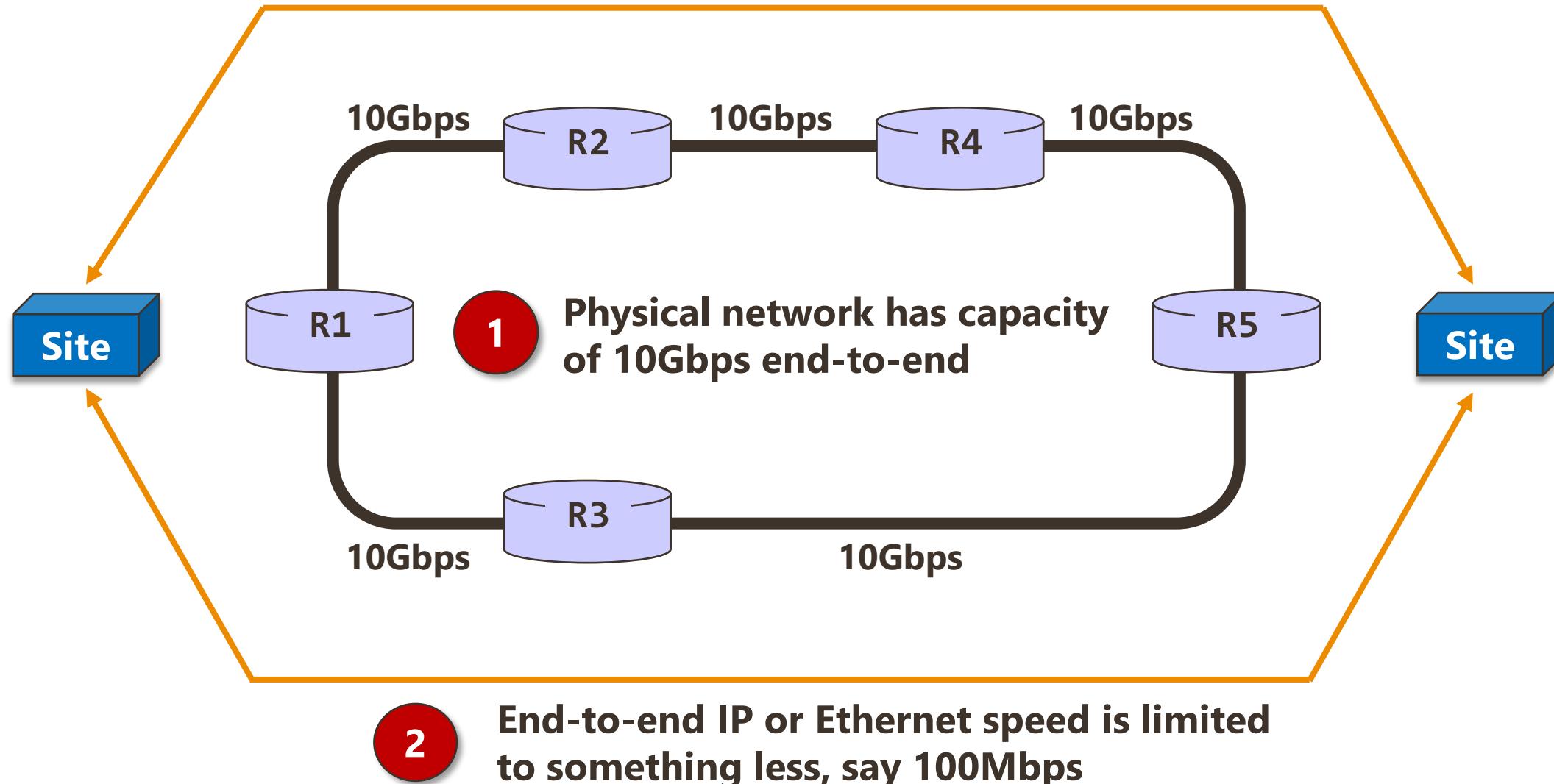


Bandwidth vs Latency

- Latency is sometimes referred to as delay or ping rate.
- It's the **lag** you experience while waiting for something to load.
- If bandwidth is the amount of information sent per second, **latency is the amount of time** it takes that information to get from its source to destination



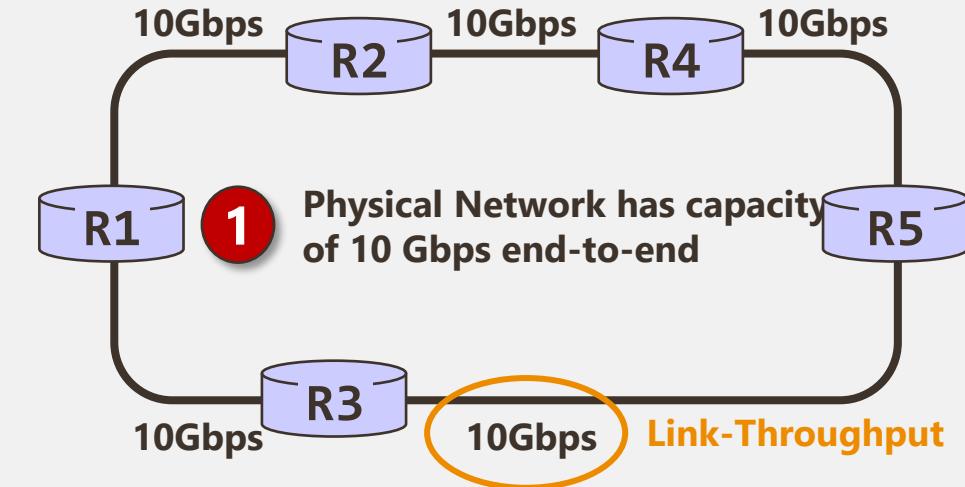
Physical network can have less speed than the total bandwidth/capacity



Bandwidth vs Throughput

Actual amount of data transfer per second

- ✓ Taking latency, network speed, packet loss and other factors into account



Bandwidth: max amount of vehicles passing bridge/hour



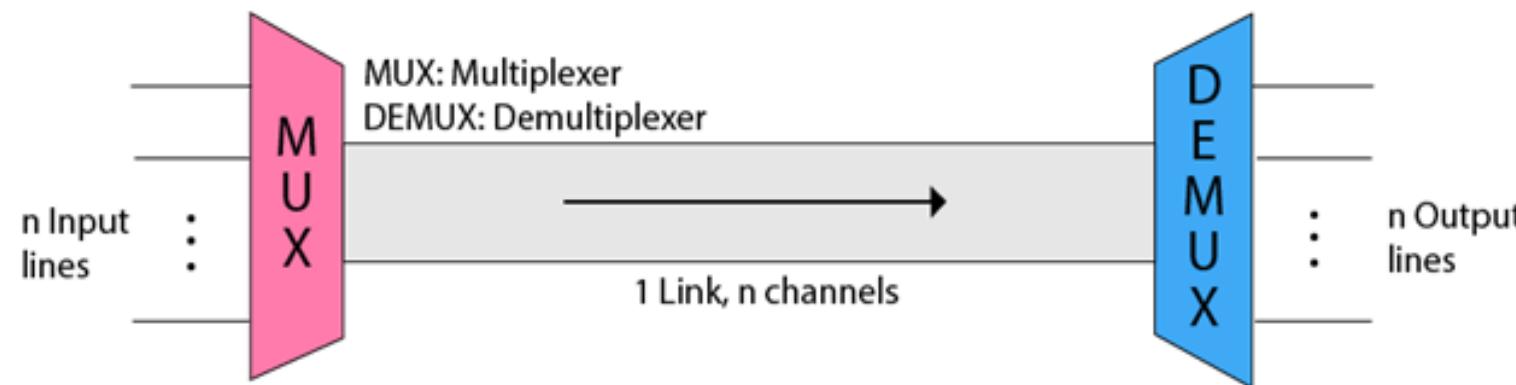
Throughput: Actual number of vehicles passing the bridge/hour



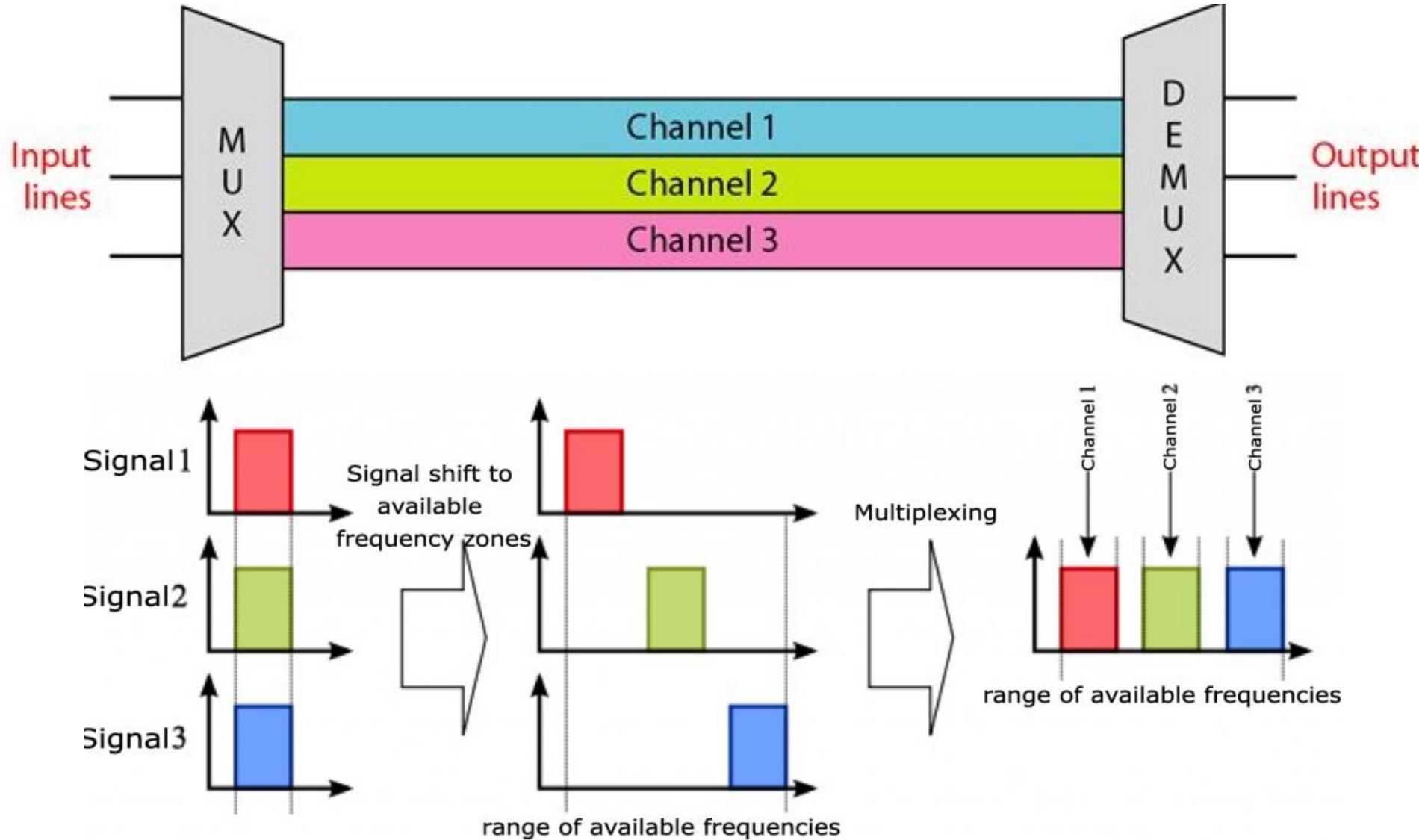
Throughput: Node to Node (link) | Speed: End-To-End

Multiplexing

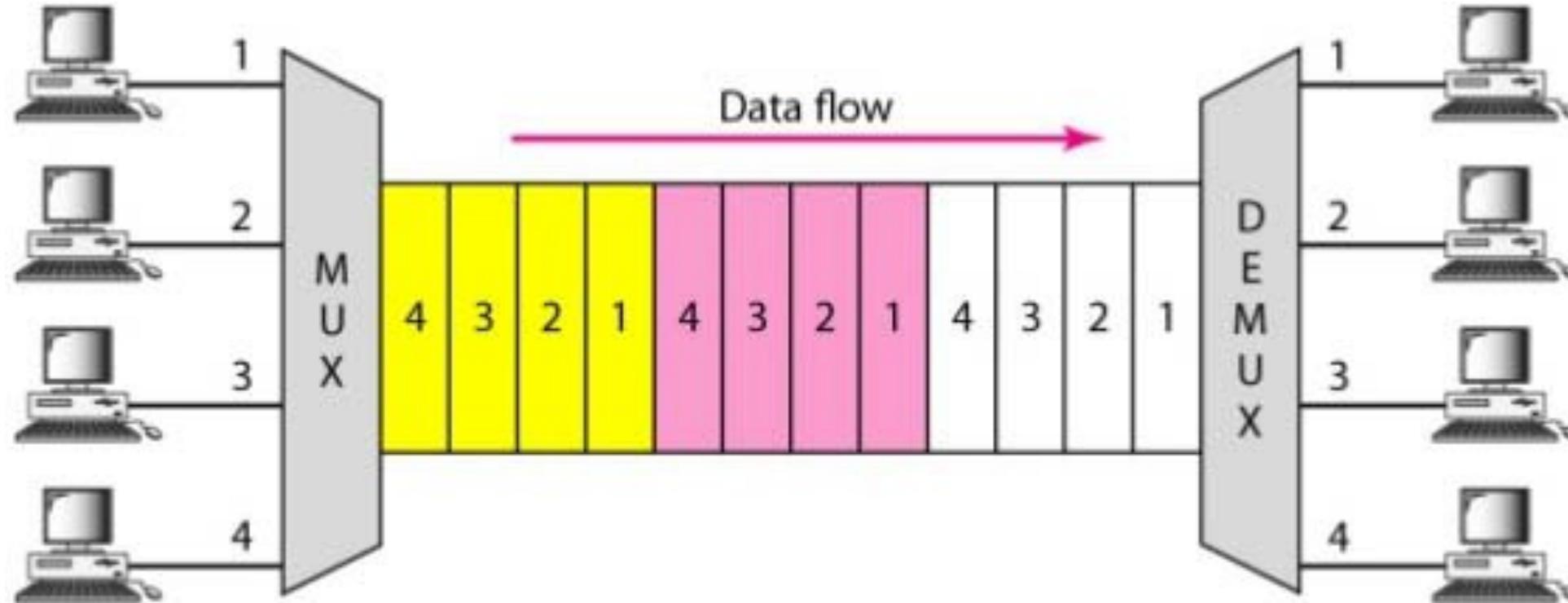
- A technique used to combine and send the multiple data streams over a single medium.
- The process of combining the data streams is known as multiplexing
- hardware used for multiplexing is known as a multiplexer.



Frequency Division Multiplexing (FDM)



Time Division Multiplexing (TDM)





Network Topologies

- Physical Topology
- Logical Topology
- Hybrid Topology

Network Topology

- Refers to **how network devices are connected**

- 3 Aspects

- **Physical Topology**

- How network devices are physically connected
 - Layout of the network

- **Logical Topology**

- The way signal passes through the network
 - *i.e. In this class there are always students physically sitting in the classroom, but logically they are far far away ☺*

- **Hybrid Topology**

- Combination of physical and logical



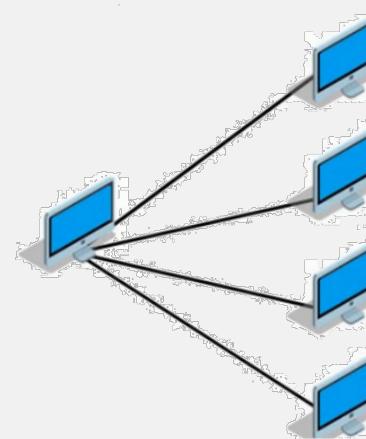


Point-to-Point (PTP, P2P)

- Foundation of all other topologies

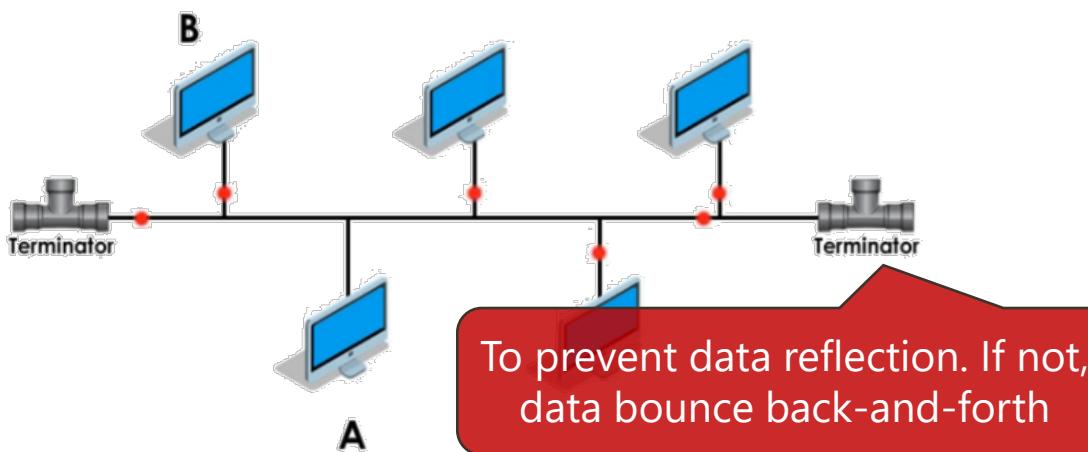
Data from central base station is broadcasted;

Data from subscribers only received by central base station



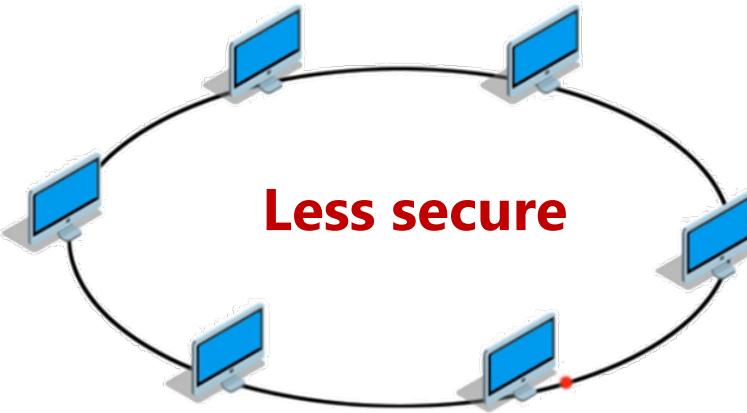
Point-to-Multipoint (PTMP, P2MP)

- Central **base station**, supports subscriber stations
- i.e. *Wireless LAN*



Bus (daisy chain topology)

- No central base station
- Nodes attached to **one shared bus cable**
- One single point failure=whole network failure
- Less secure

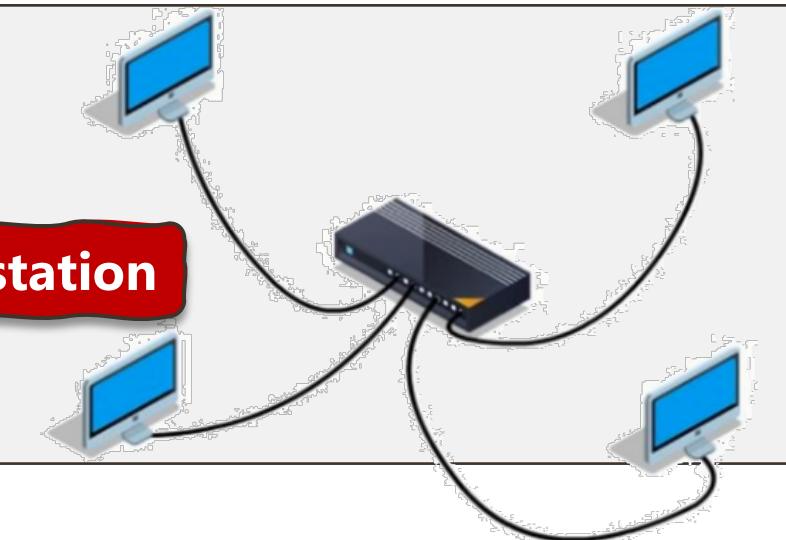


Ring Topology

- Nodes attached to a cable forming a closed loop
- No need for Terminators
- **Data** travels from **one node to another** via a token passing
- One single point failure=whole network failure

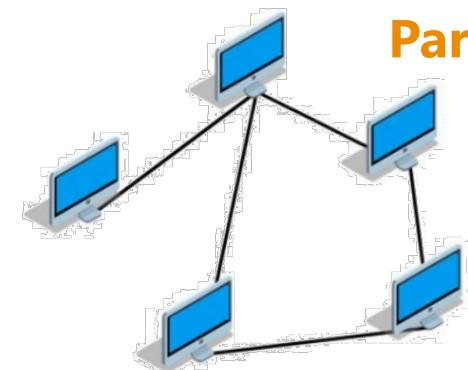
Star Topology

- Nodes communicate with other Nodes via a **central device**
- Fault Tolerant (if one node goes down)
- Scalable, Easy Maintenance
- **-VE: Central Point of Failure**



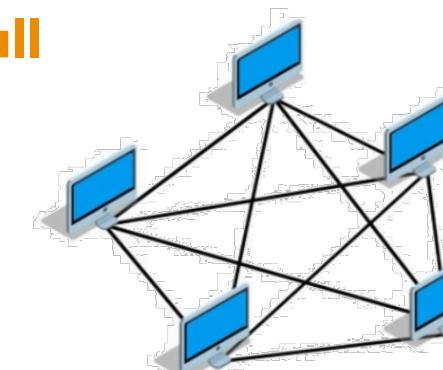
Mesh Topology

- Partially Meshed
- Fully Meshed



Partial

Full

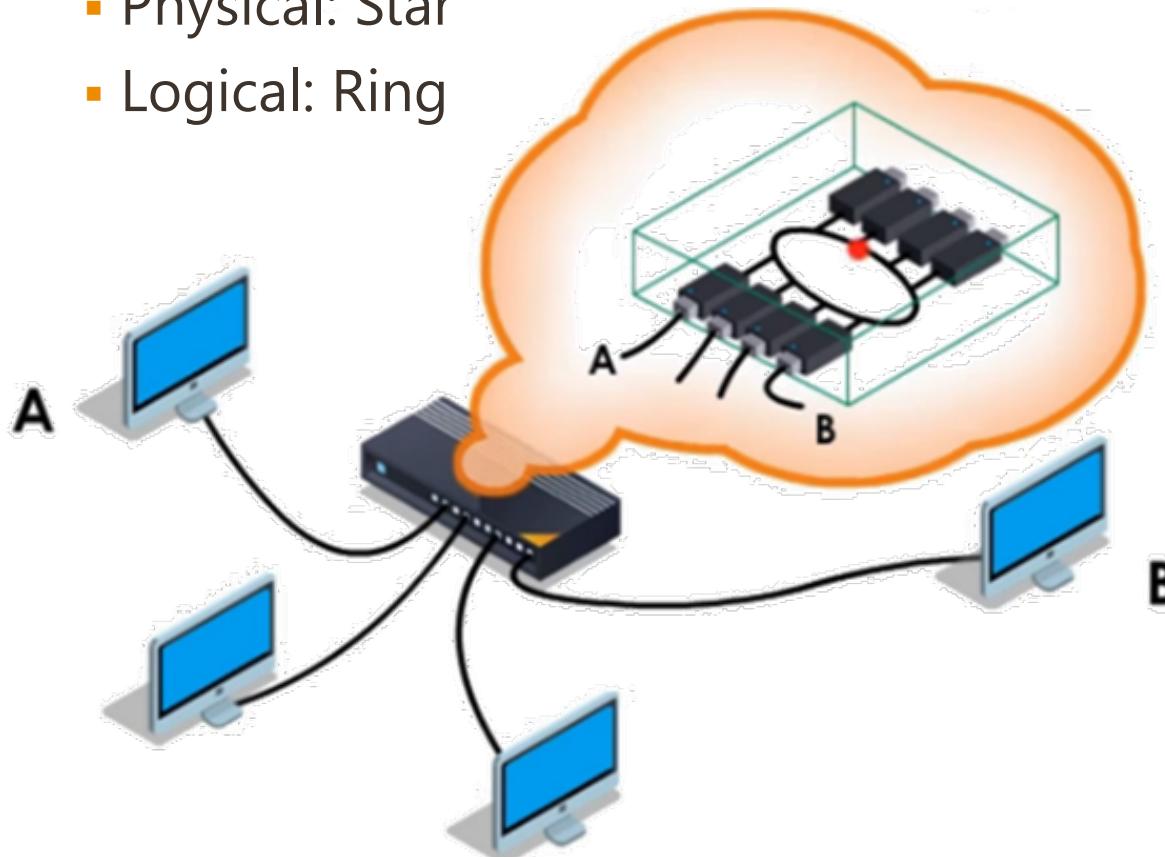


i.e. ad hoc wireless LAN

Hybrid Topology

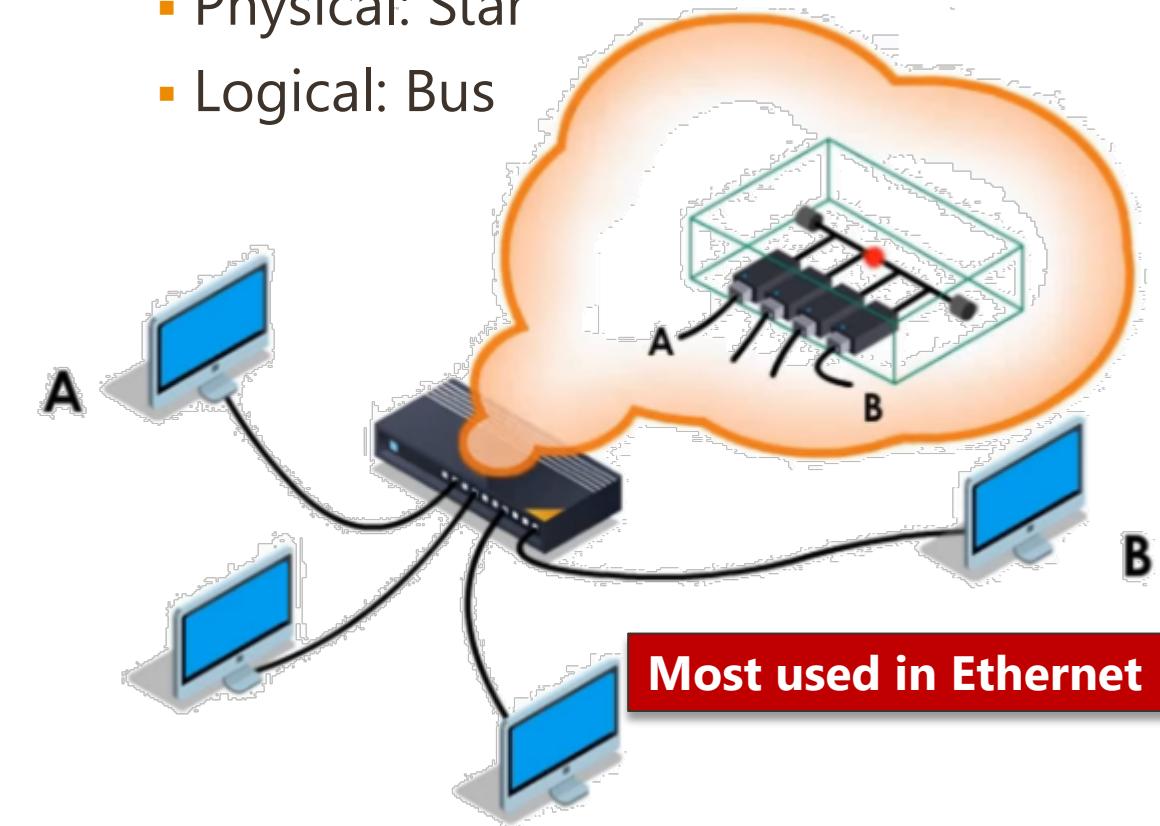
Star-ring Topology

- Physical: Star
- Logical: Ring



Star-bus Topology

- Physical: Star
- Logical: Bus





Transmission Media

- Transmission Modes
- Guided Media
- Unguided Media
- Wireless Signals
- Wireless Signal Attenuation

Transmission Modes

- **Simplex Mode:** communication is unidirectional **Only 1-way**

- ✓ Sender -> Receiver | **not Receiver -> Sender**
- ✓ Entire bandwidth can be used
 - E.g. A keyboard -> monitor
 - E.g. Radio station -> audience

- **Half Duplex Mode:** communication is two-directional **1-way at a time**

- ✓ Sender -> Receiver | Receiver -> Sender (one at a time)

- **Full Duplex Mode:** communication is bi-directional **2-way, same time**

- ✓ Sender -> Receiver | Receiver -> Sender (at the same time)

Transmission Modes – cont.

Comparison	Simplex	Half Duplex	Full Duplex
Direction of Communication	Unidirectional One-way	Two-directional, one at a time	Two-directional, simultaneously
Send / Receive	Sender can only send data	Sender can send, receive data, but one at a time	Sender can send and receive data simultaneously
Performance	Worst performance	Better than Simplex	Best performing mode of transmission
Example	Keyboard and monitor	Walkie-talkie	Telephone

Transmission Media

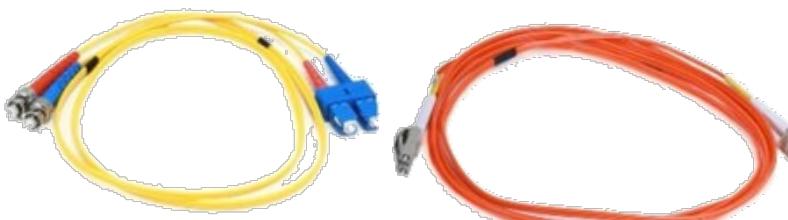
■ Guided:

- ✓ Twisted Pair
- ✓ Coaxial Cable
- ✓ Fiber Optics



Unshielded Twisted Pairs (UTP)

Shielded Twisted Pairs (STP)



Fiber Optic Cable

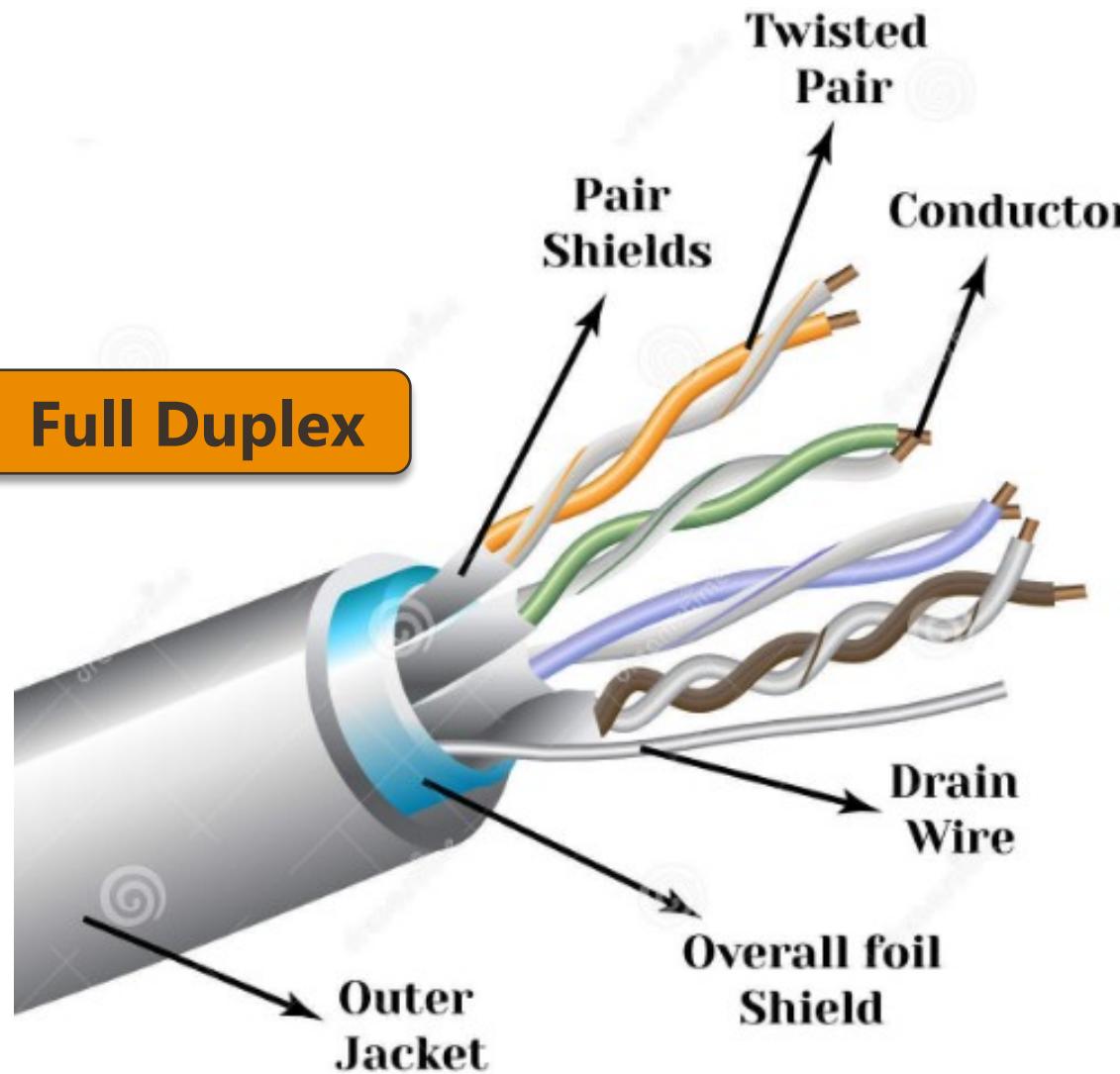
■ Unguided:

- ✓ Radio
- ✓ Microwave
- ✓ Lightwave
- ✓ Infra-red



Coaxial Cable

Twisted Pair



- Oldest and still **common** transmission media
- Can transmit **analog or digital** signals
- Most common is **UTP** (Unshielded Twisted Pair)
- Uses **repeaters** for long distance connection
- Less expensive

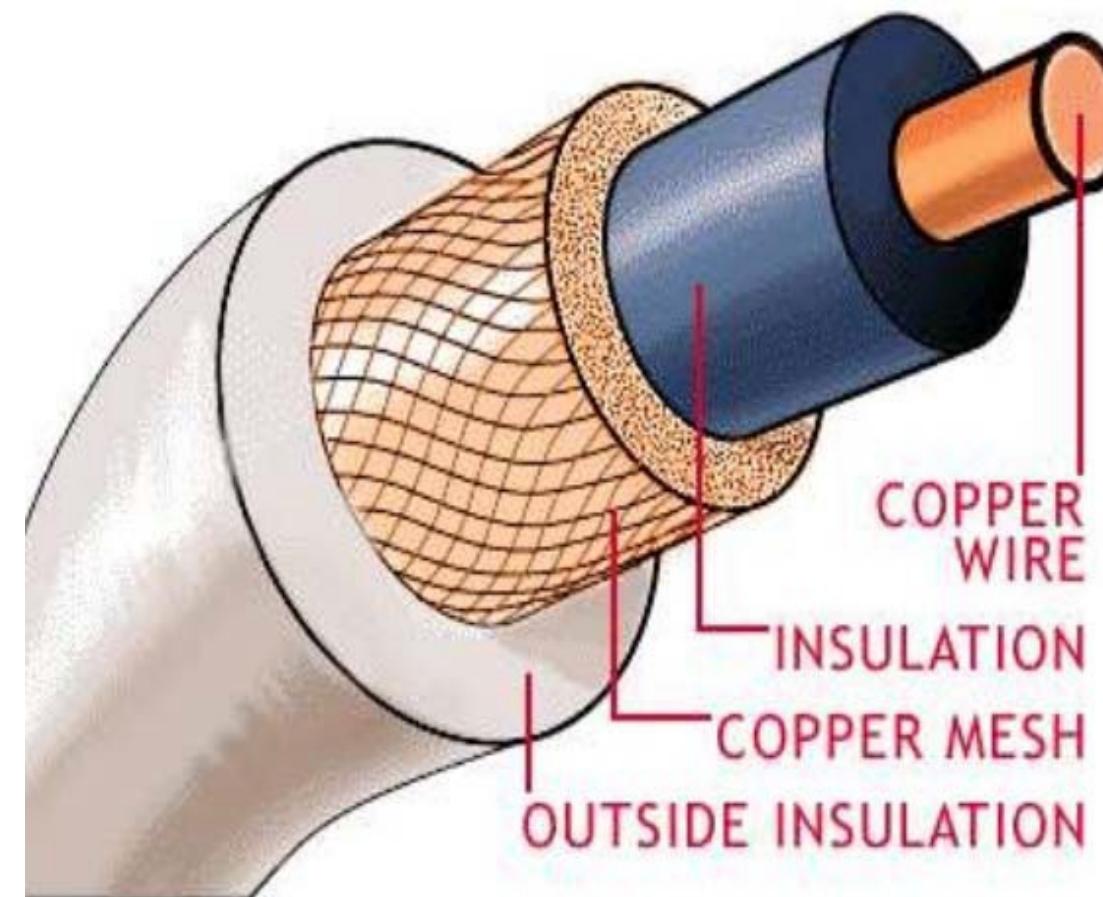
Coaxial Cable

- **Better shielding** than twisted pairs

- **Not flexible**

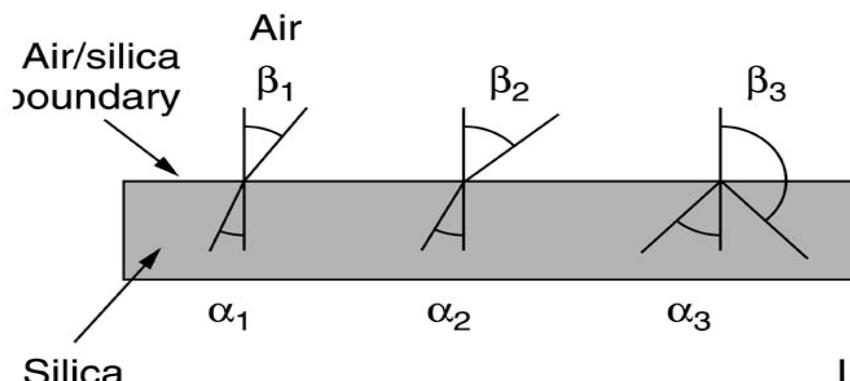
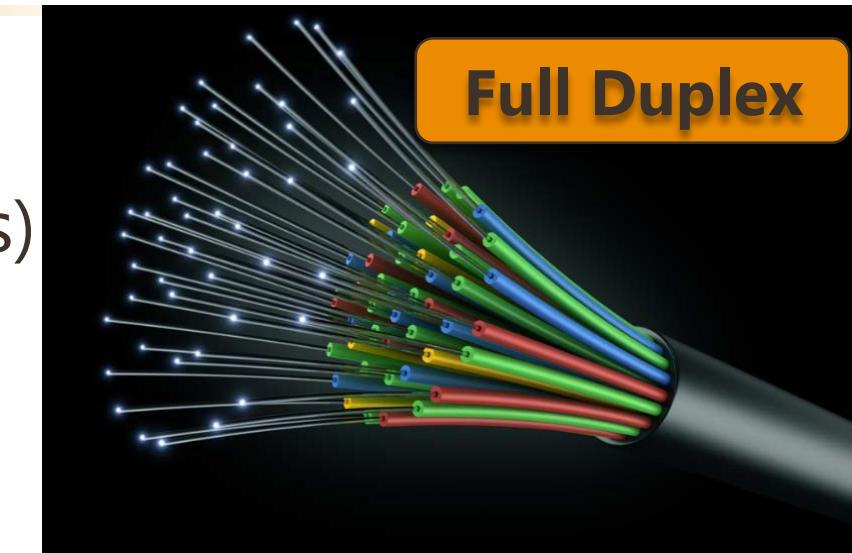
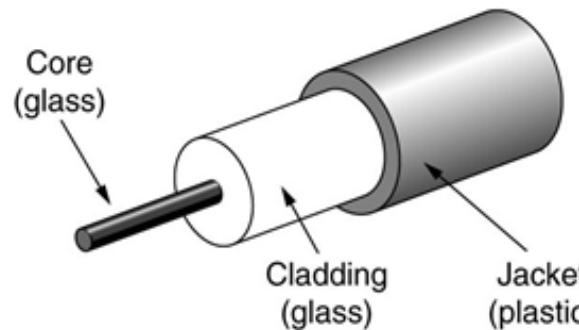
- Two kinds:
 - ✓ 50-ohm cable (digital transmission)
 - ✓ 76-ohm cable (analog transmission & cable television)

Half Duplex

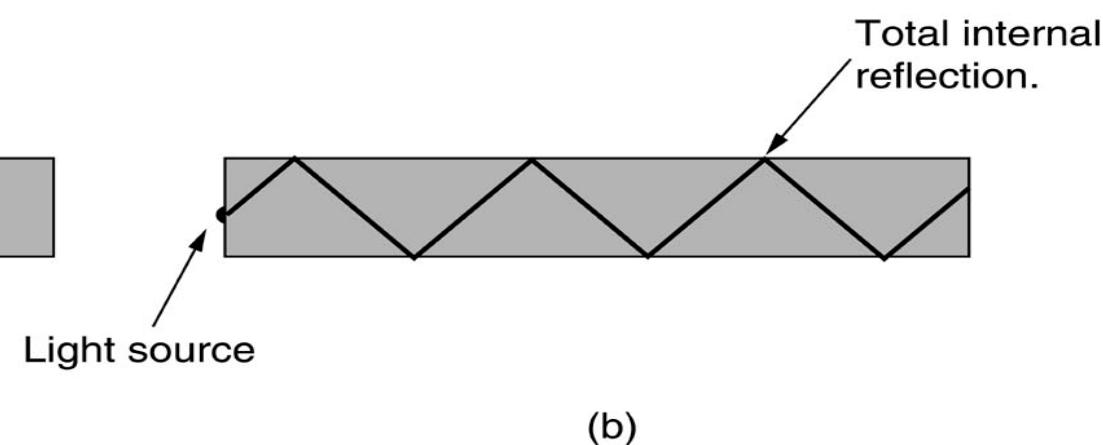


Fiber Optic

- Light source (speed of light)
- Transmission medium (ultra-thin fiber of glass)
- Detectors

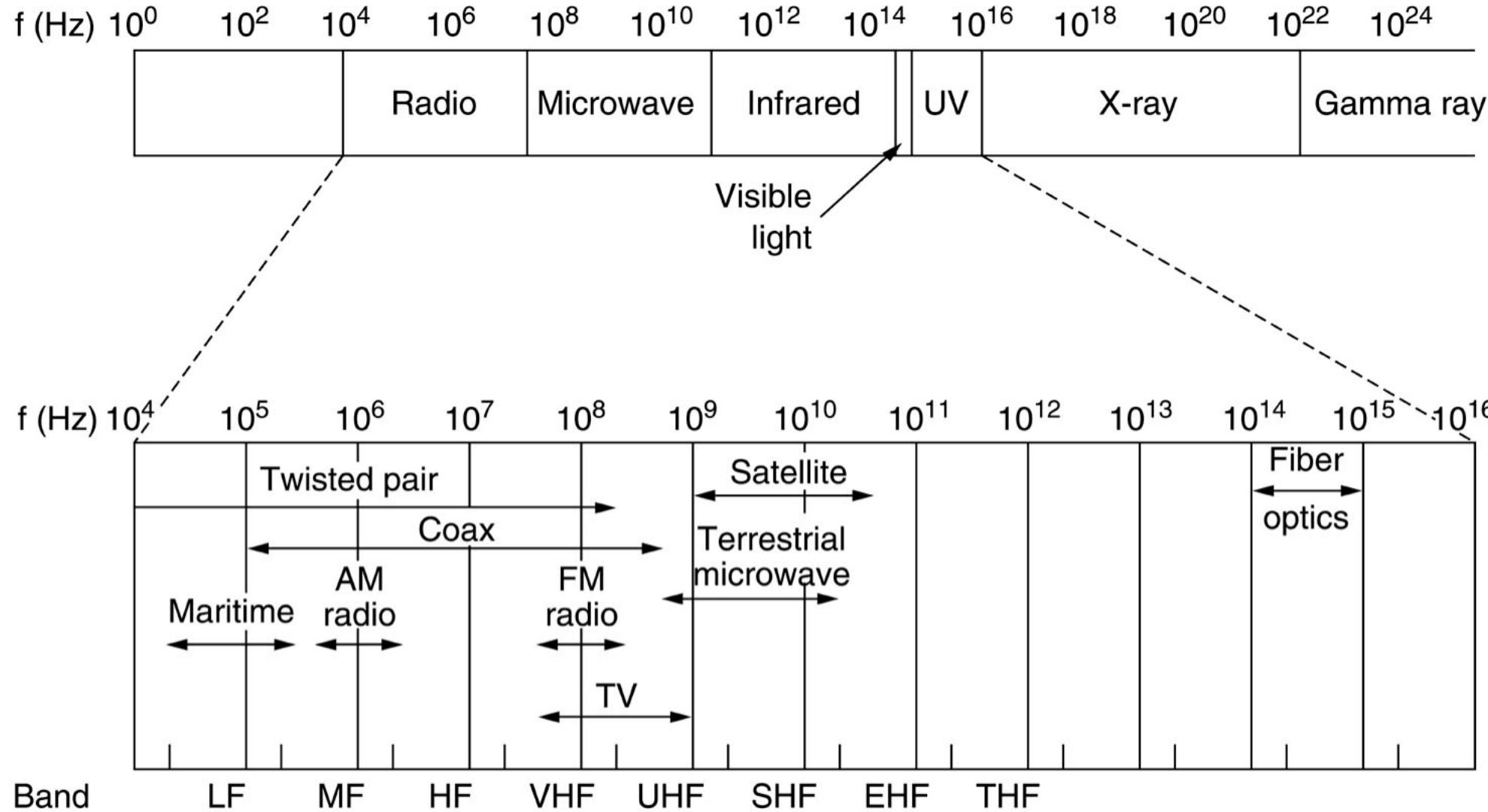


(a)



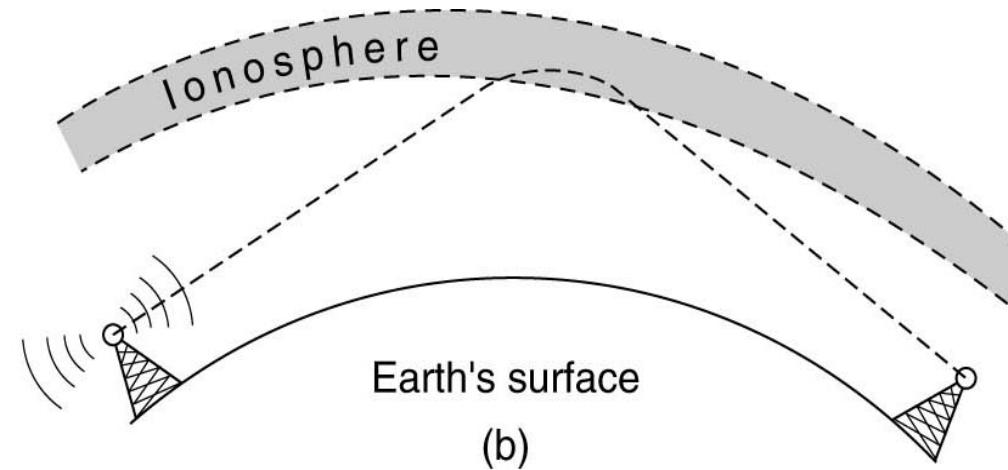
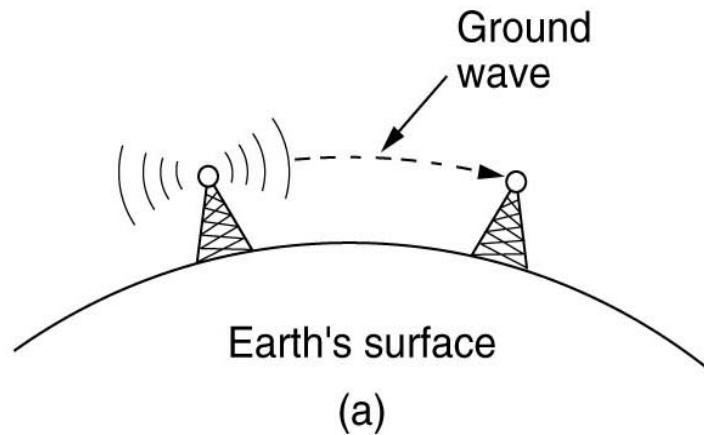
(b)

Unguided: Electromagnetic Spectrum

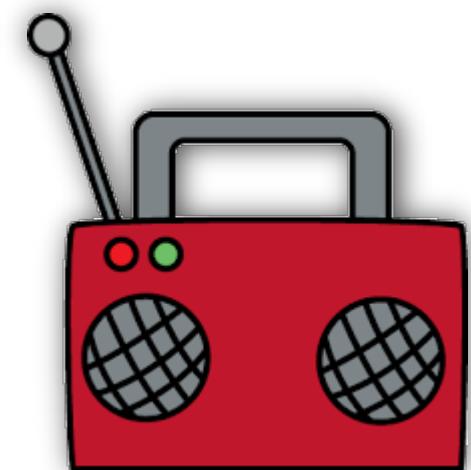


Radio Transmission

- Easy to generate & travel long distance
- Omni-directional



- (a) VLF, LF & MF bands, radio waves ***follow the curvature*** of the earth.
(b) HF band ***bounced off*** the ionosphere. (Tanenbaum)



Microwave Transmission

- Travel in **straight line**
- Used for:
 - ✓ Long-distance telephone communication
 - ✓ Television distribution
 - ✓ Etc.



Infrared & Millimeter Waves

- **Short-range** communication
 - Remote controls
- **Cheap** and easy to built
- **Cannot penetrate** solid objects

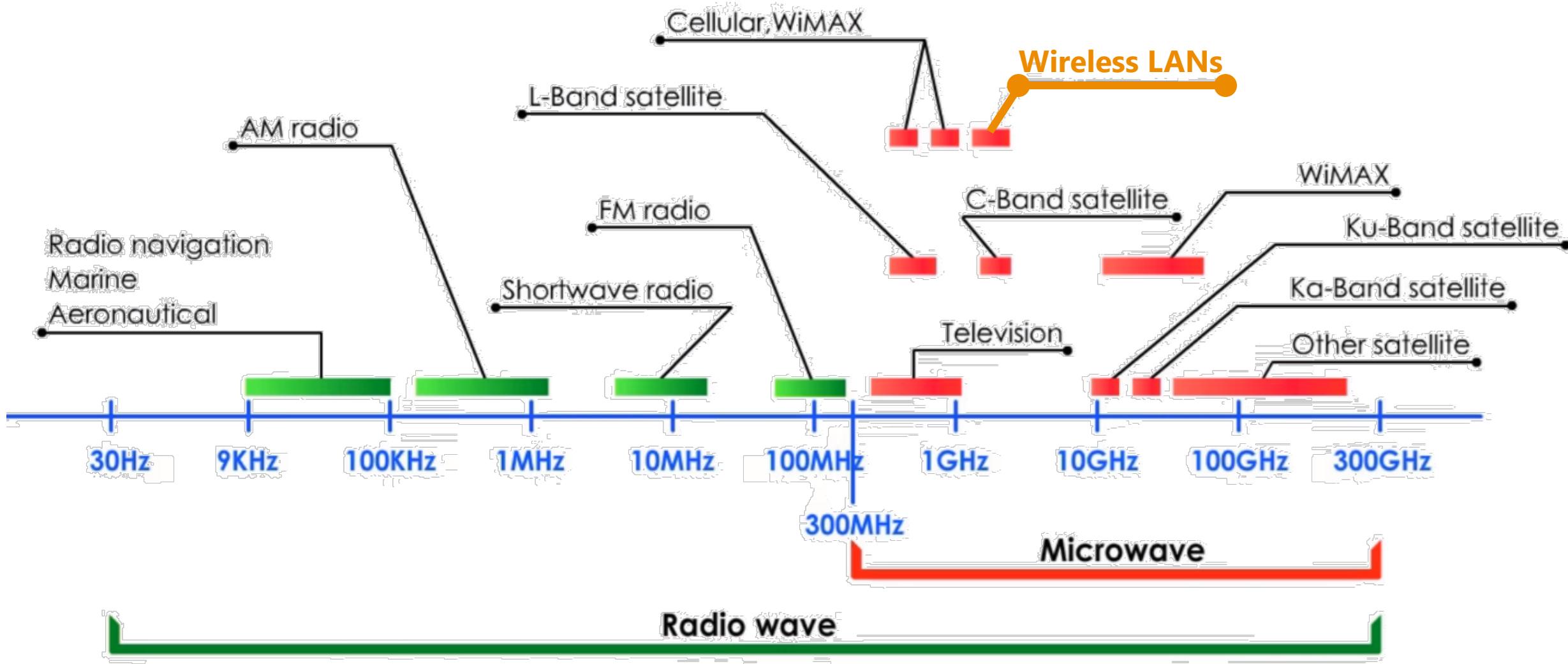


Light Wave Transmission

- **Laser or Light**
- **Unidirectional**
- Offer **high bandwidth** & very low cost
- **Cannot penetrate** rain or thick fog



Wireless Signals



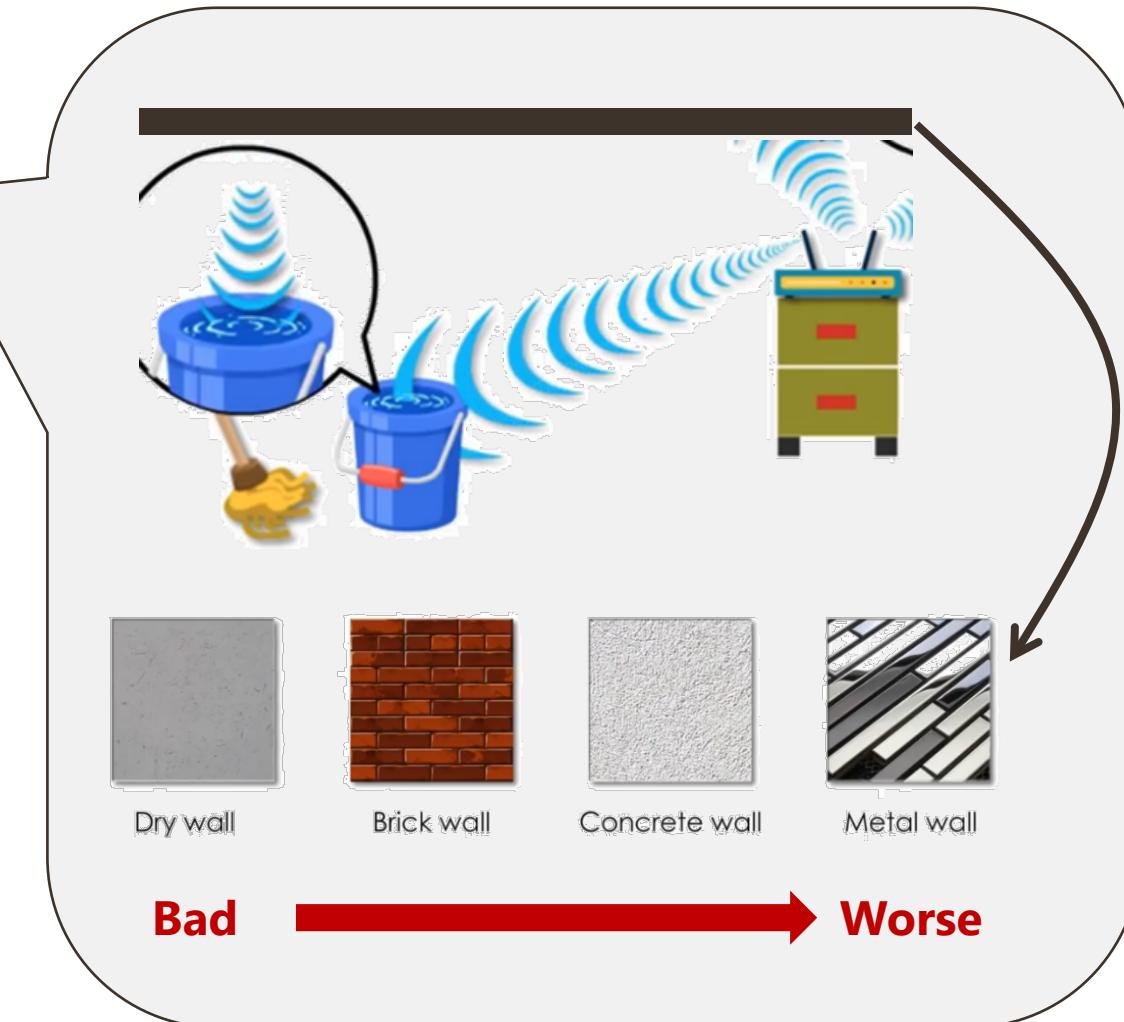
Wireless Signal Attenuation?

▪ Absorption

- Some materials absorb signal's strength

▪ Reflection

- Some absorbed, some reflected
- Not necessarily bad



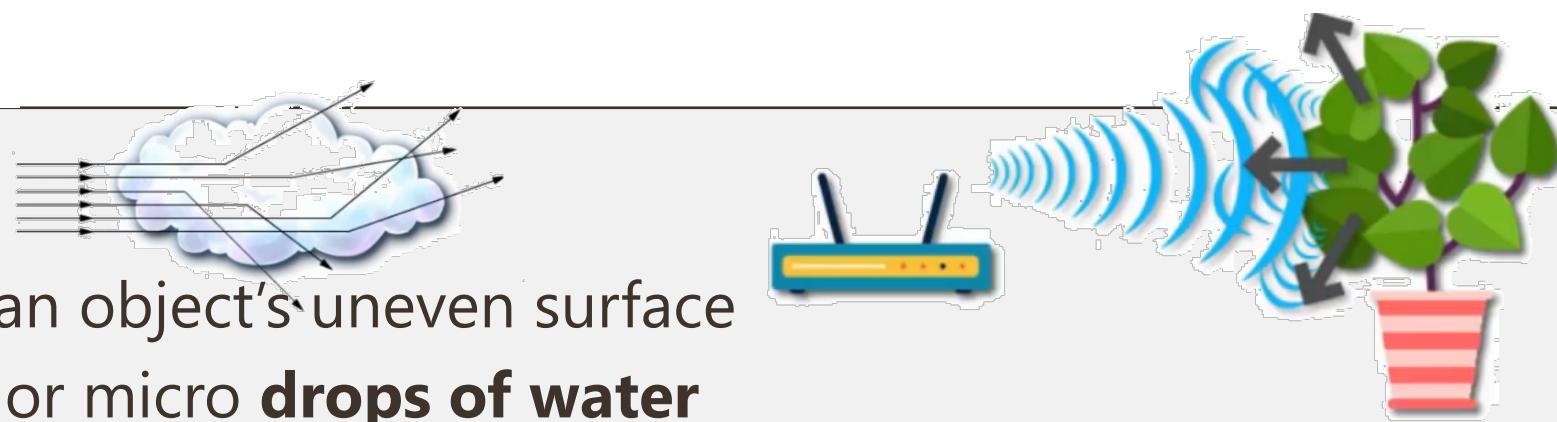


Diffraction

- Sharp edges, corners, etc

Scattering

- Diffusion of wireless signal at an object's uneven surface
- i.e. plant leaves, **dust**, **smoke**, or micro **drops of water**



Interference

- Devices operating on same 2.4Ghz

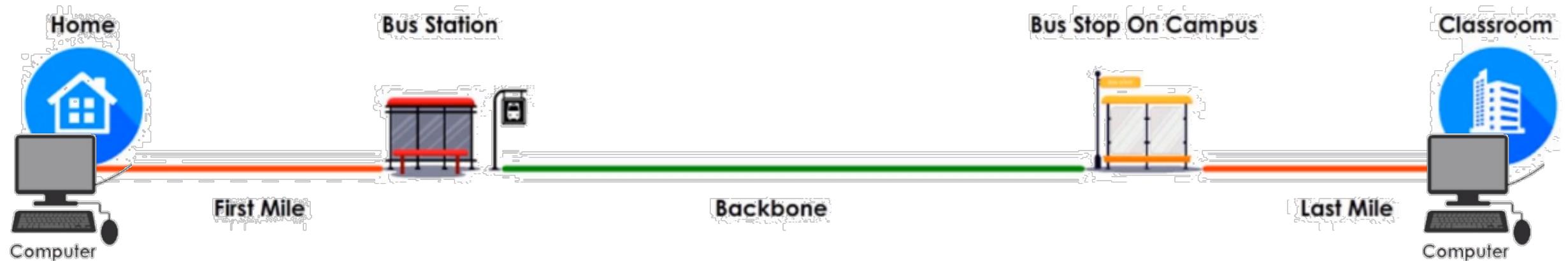


Last Mile Technologies

- Dial-up
- ISDN
- DSL/ADSL
- NBN

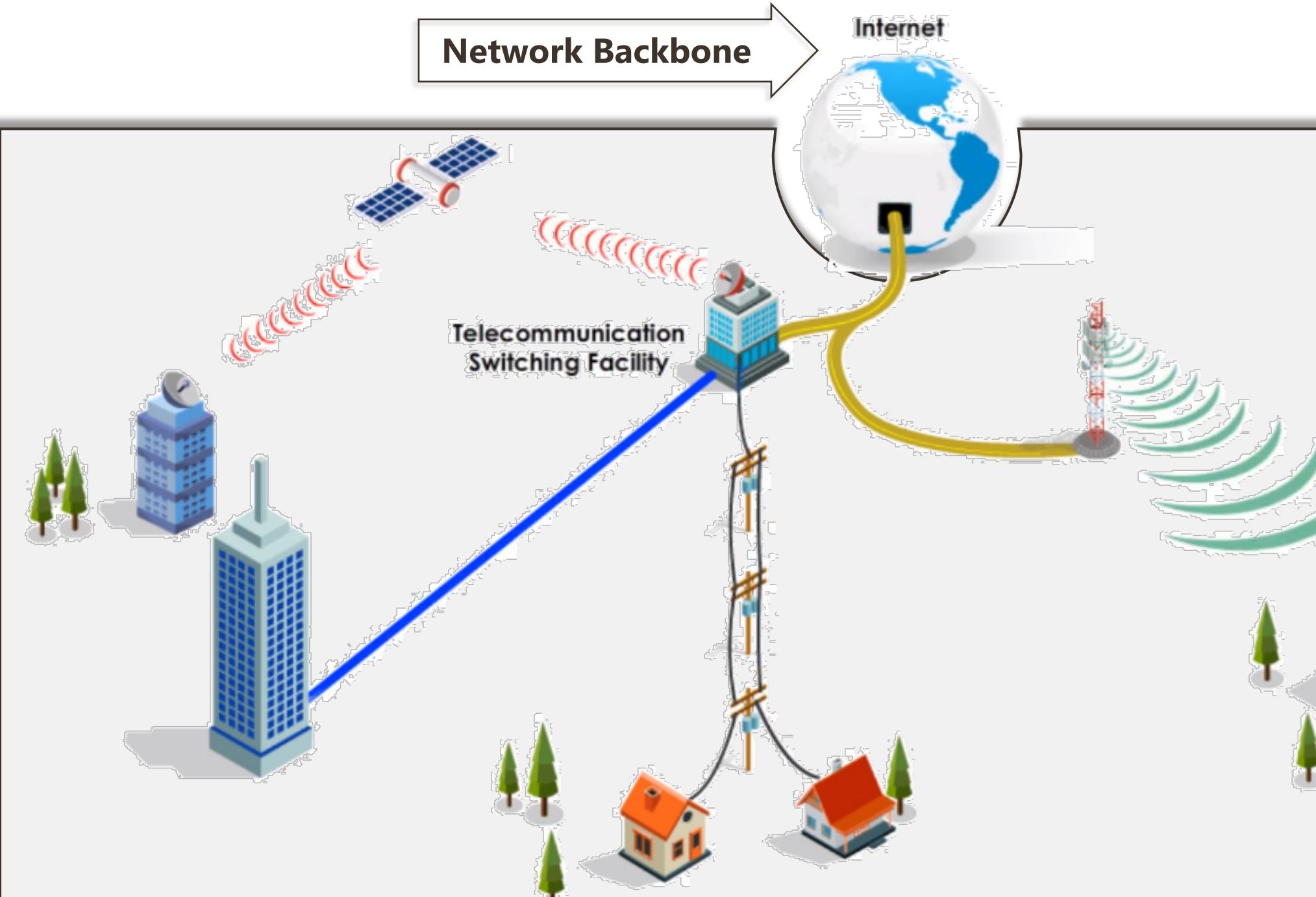
Last Mile Technologies

- Refers to any **telecommunication methods, devices** and **media** that **carry signals over the last/first mile**
- **E.g.**
Network backbone use Fibre Optics but the last mile still use copper wires
- **Typically the speed bottleneck**





Network Backbone



LAST MILE TECHNOLOGY

Dial-up

DSL

Cable
modem

Broadband

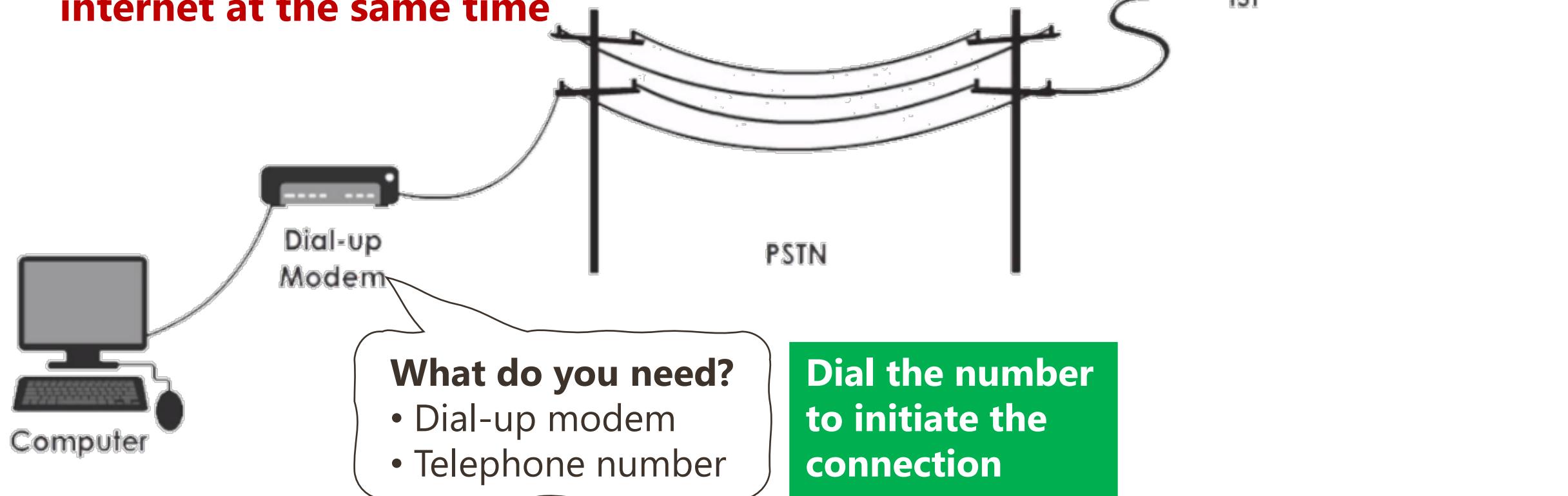
WiMax

Cellular

Fibre Optics
Satelite

Dial-up

- Very old technology (1990s)
- Max speed: 56Kbps
- **Cannot use telephone and internet at the same time**



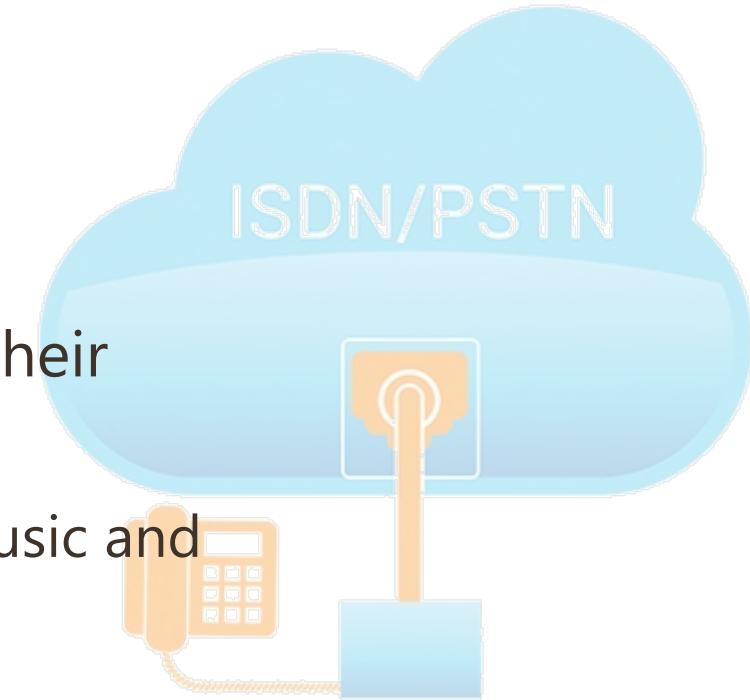
Dial-up: PSTN

- PSTN or Public Switched Telephone Network is most commonly known as a '**telephone line**'
- Users need to use **one line** for one conversation at a time using only one **phone number (=one phone line)**
- Circuit-switched copper phone lines are used to transmit **analogue voice data**
- As a **dedicated service**, a PSTN line cannot be used for any other purpose while a call is being made



ISDN

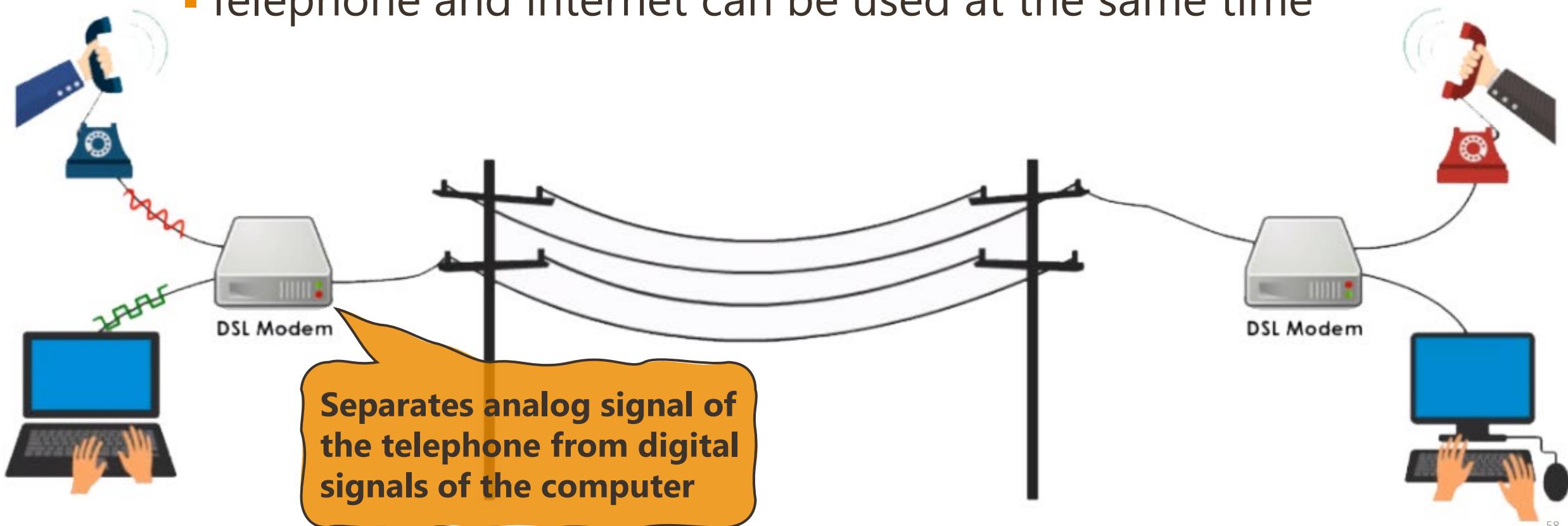
- Integrated Services Digital Network (ISDN) provides **digital transmission of voice and data** services
- ISDN supports **multiples channels** for voice and data.
- Medium to Large Businesses: Option of integrating with their phone systems (PABX) to enable **multiple features**
 - ✓ Like using a 100-number range, groups, queues, on hold music and RVAs, etc.



With **NBN** (National Broadband Network), PSTN and ISDN will be phased out.

Digital Subscriber Line (DSL)

- Faster data transfer over PSTN
- Always on connection (No dialing required)
- Telephone and internet can be used at the same time



DSL

- **ADSL (Asymmetric DSL)**

- Downstream is faster than upstream

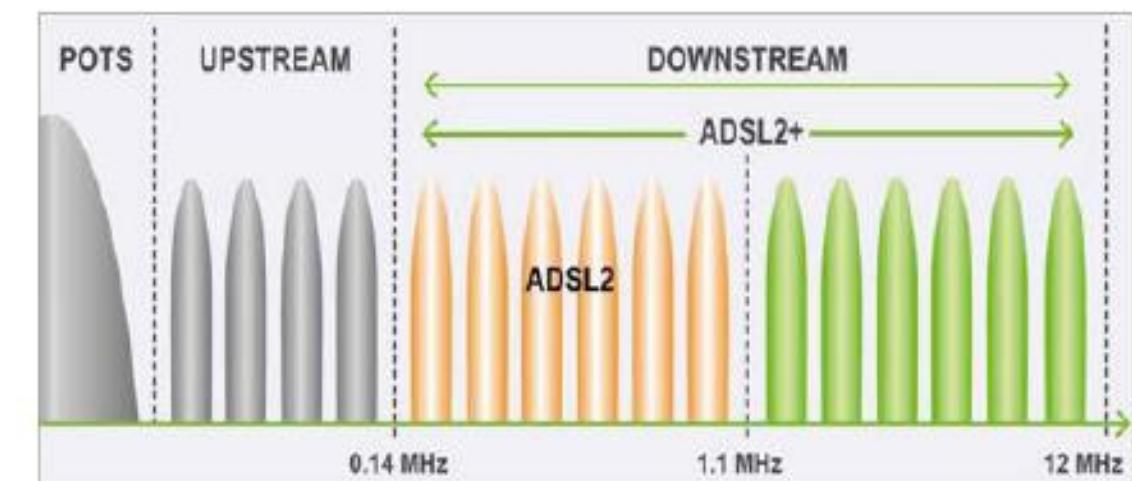
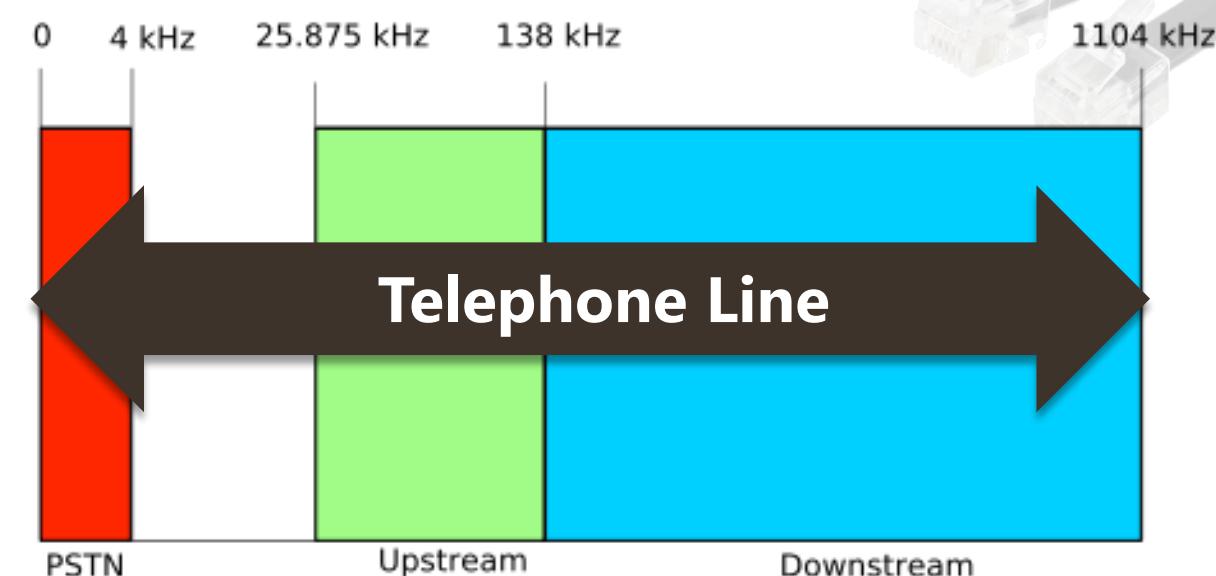
- **For business use:**

- **SDSL (Symmetric DSL)**

- i.e. video conferencing

- **HDSL (High bit-rate DSL)**

- **VDSL (Very high bit-rate DSL)**



National Broadband Network (NBN)

- National network of communication infrastructure currently being built on behalf of federal government
- **For internet and phone (voip)**
- Aims to deliver network speed of **50Mbps – 1Gbps**

- Broadband Network via Fibre Optics, Fixed Wireless, Satellite Technology for home users
- **Always on** connection (*no dialing*)

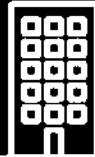
NBN Types



FTTP



FTTC



FTTB



HFC



FTTN



Fixed Wireless

NBN Types

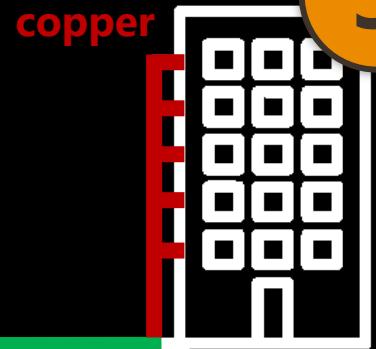


FTTP

Fibre To The Premises

- 1 Fibre is connected all the way from exchange to office/house
 - **GOLD** standard in NBN

$\leq 1 \text{ Gbps}$



FTTB

Fibre To The Building

- 3 Fibre to central point in the building
 - Copper to each socket

$\leq 100 \text{ Mbps}$

2

- Fibre to curb or driveway

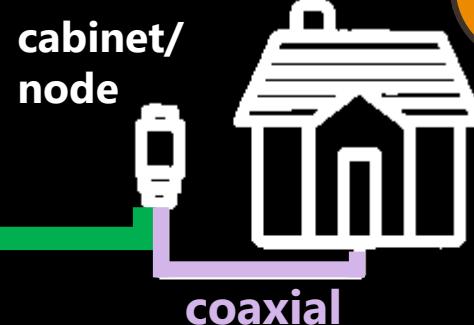


FTTC

Fibre To The Curb

$\leq 100 \text{ Mbps}$

NBN Types



HFC
Hybrid Fibre Coaxial

<= 100 Mbps



Fixed Wireless

12-50 Mbps

4

Fibre to cabinet (a.k.a node) near your premises

- Existing coaxial cable from cabinet to premises

5

Fibre to node near your premises

- Existing copper lines (50m-1.5km) from node to premises



FTTN
Fibre To The Node

25-100 Mbps

6

Remote areas

- Fixed antenna to receive the signal from the tower
- Wireless can be affected by many factors

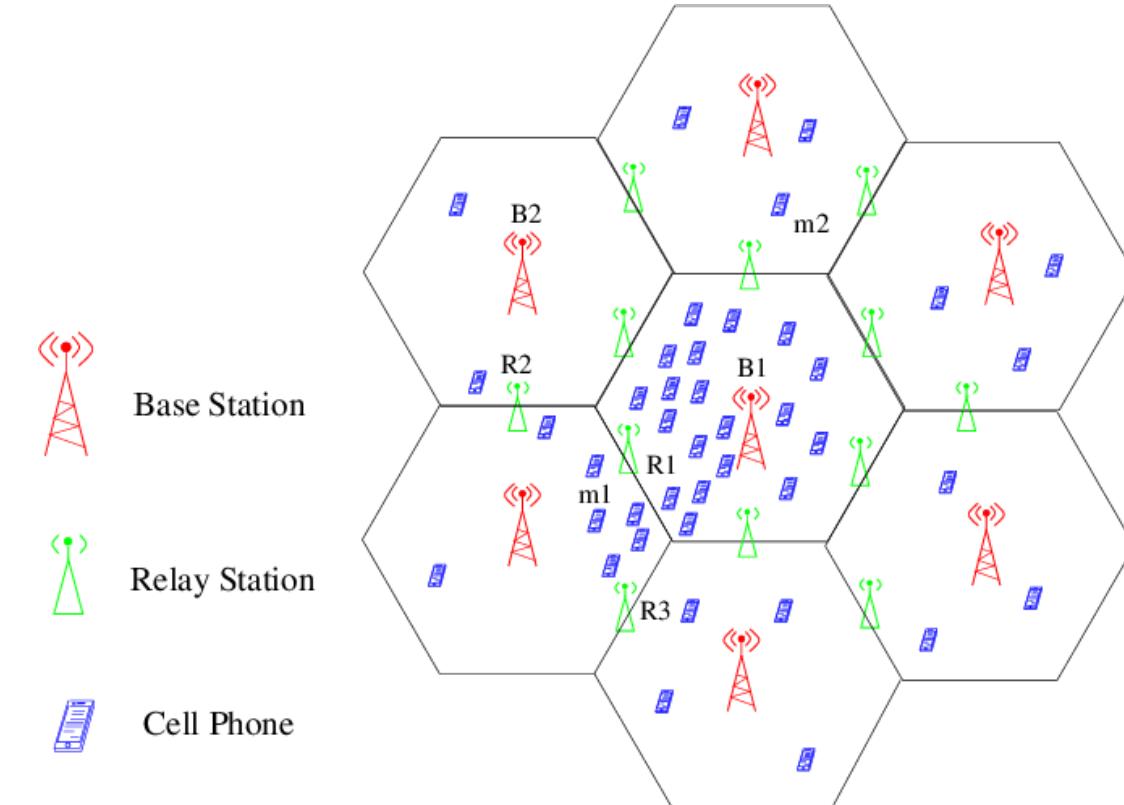


Last Mile Technologies – Cellular Networks

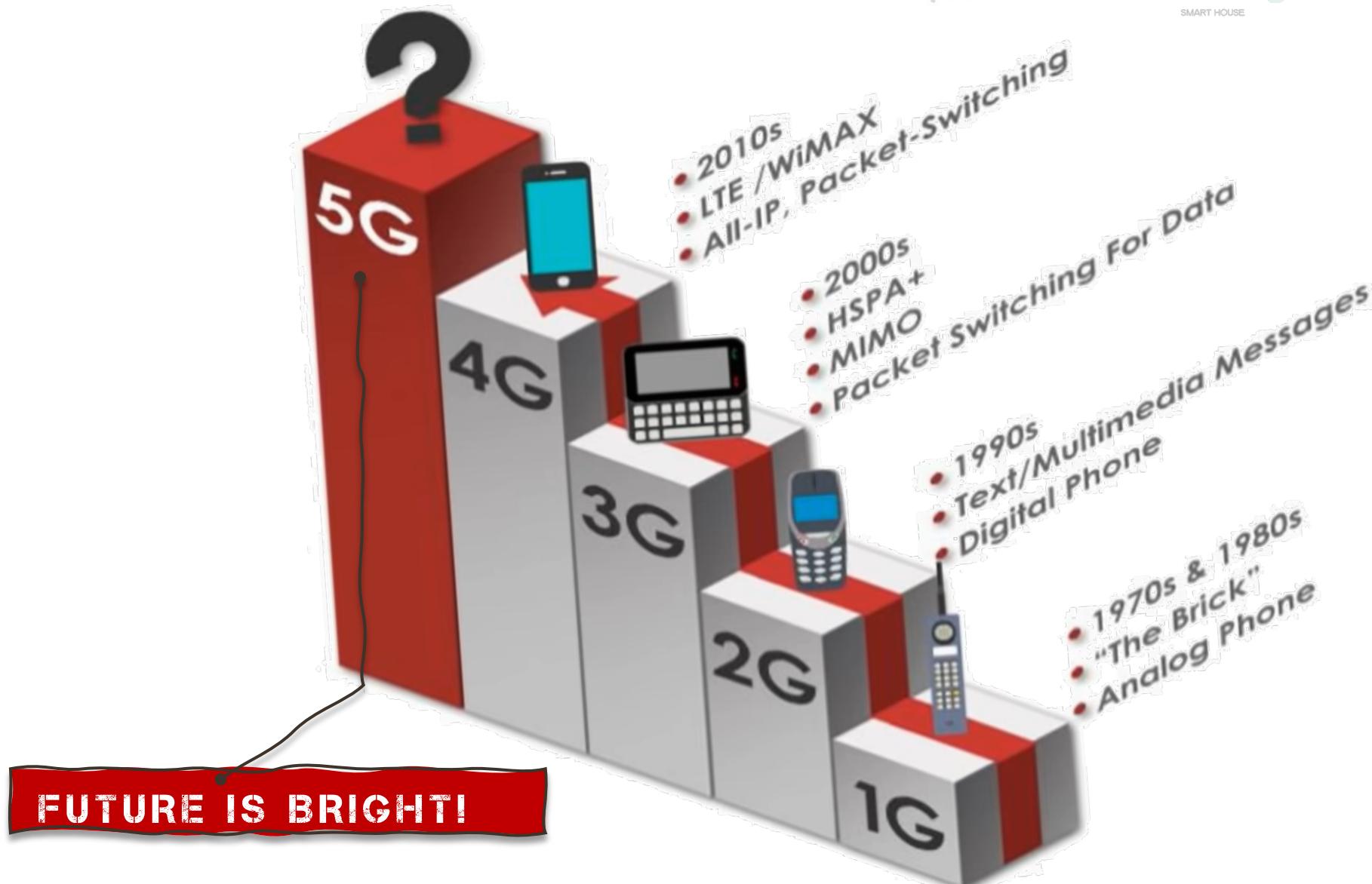
- 1G, 2G, 3G, 4G, **5G**
- 5G Technologies
 - Millimetre Wave
 - Massive MIMO
 - Small Cell
 - Beamforming
 - NOMA (non-orthogonal multiple access)
 - MEC (Mobile Edge Computing)

Cellular Networks

- High-speed, high-capacity voice and data communication networks
- Enhanced multimedia and seamless roaming capabilities for supporting cellular devices.



Cellular Networks



Cellular Networks



- **1G**: Analog signals only: voice calls
- **2G**: Ran on digital signals. SMS (and MMS).
- **3G**: Still in use today. Greater voice and data capacity and speed. Mobile phones are no longer just about making calls, but hubs of social connectivity.
- **4G**: Much faster than 3G (Up to 100Mbps). Offer connectivity for tablets and laptops as well as smartphones.

Cellular Networks

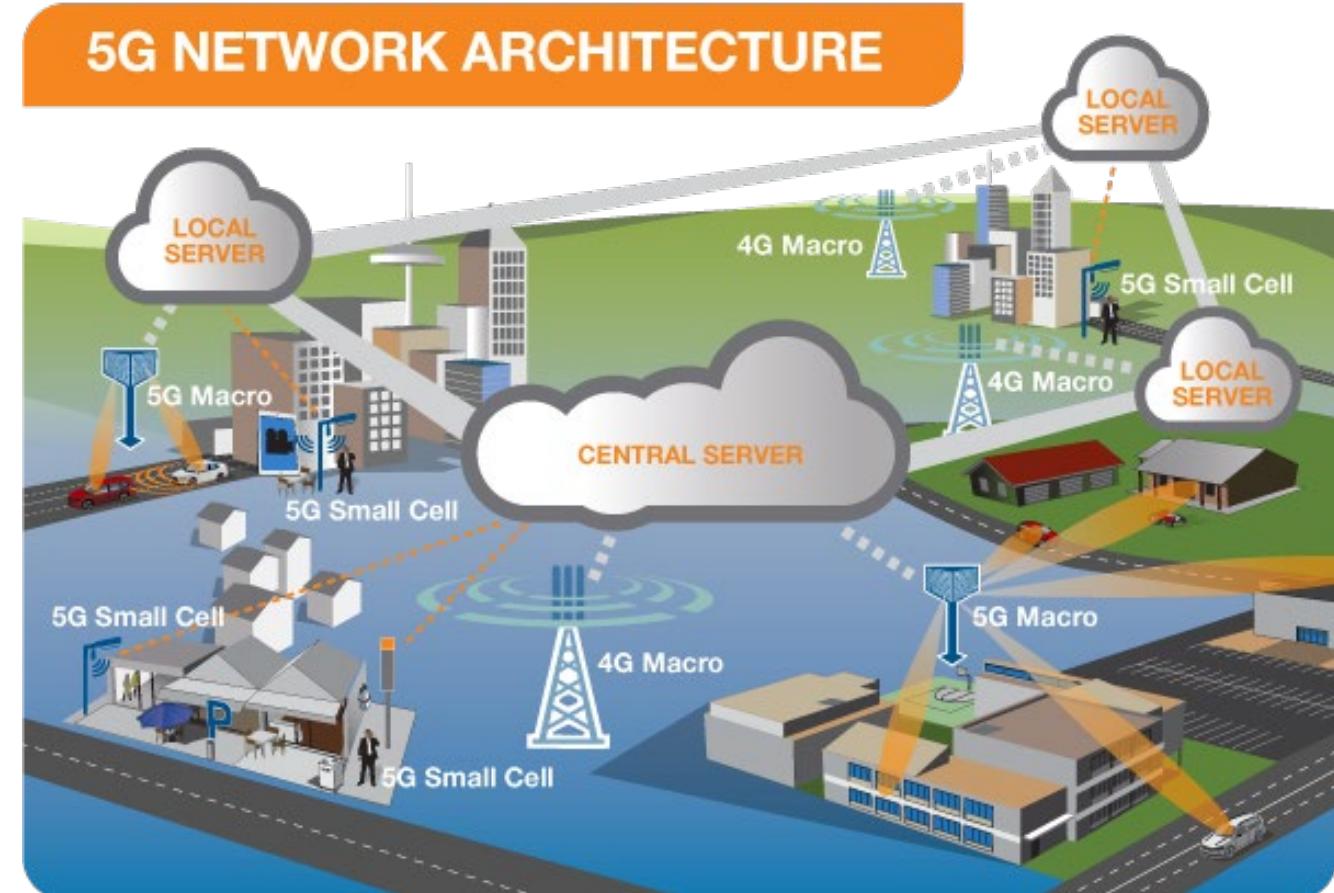
- **2020s: 5G era**
- High speed, low latency, and high connectivity
- 3 major categories:
 1. **Machine-to-machine communication: IoT**
 2. **Ultra-reliable low latency communications**
 3. **Enhanced mobile broadband**



5G Technologies

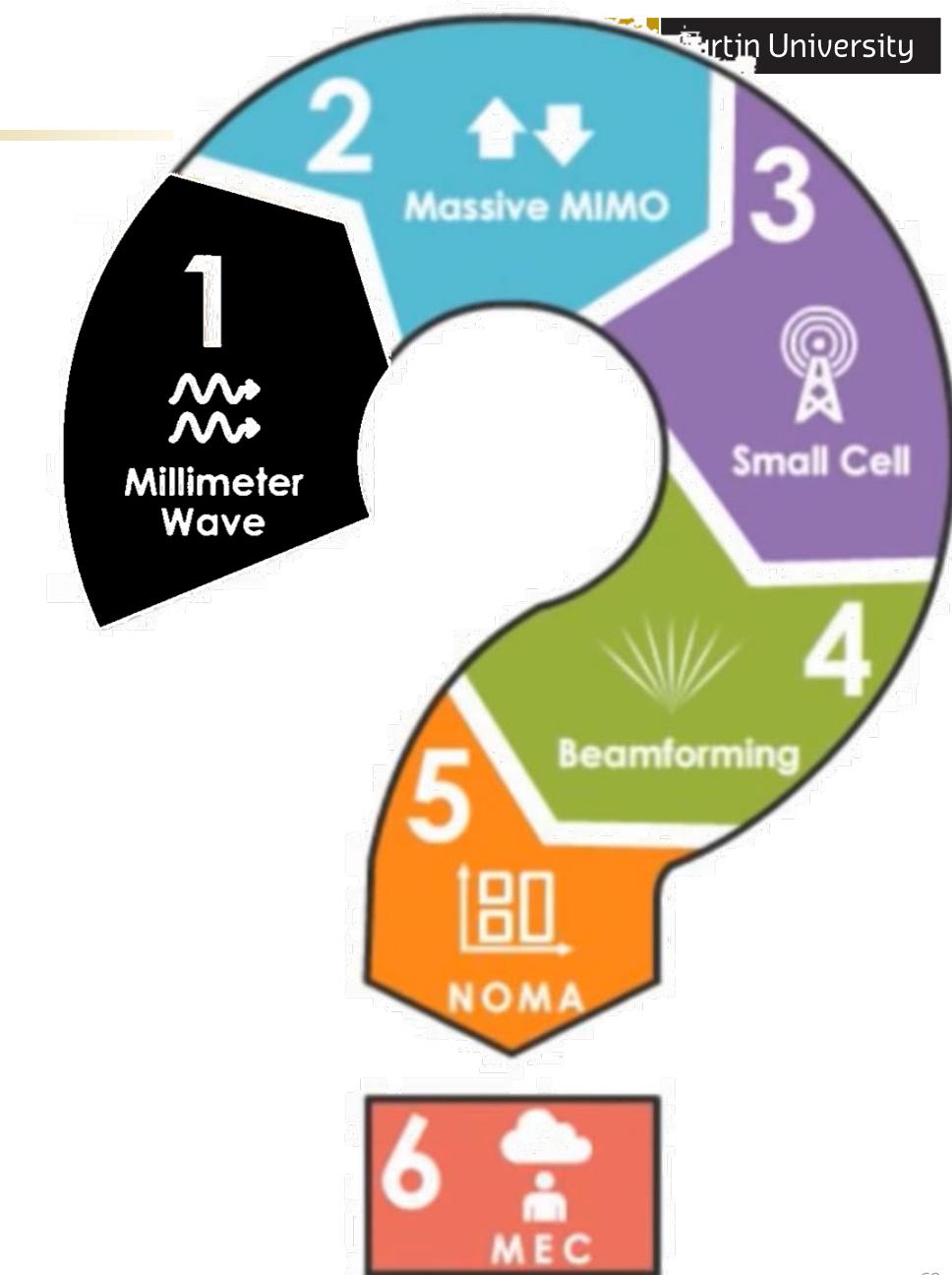
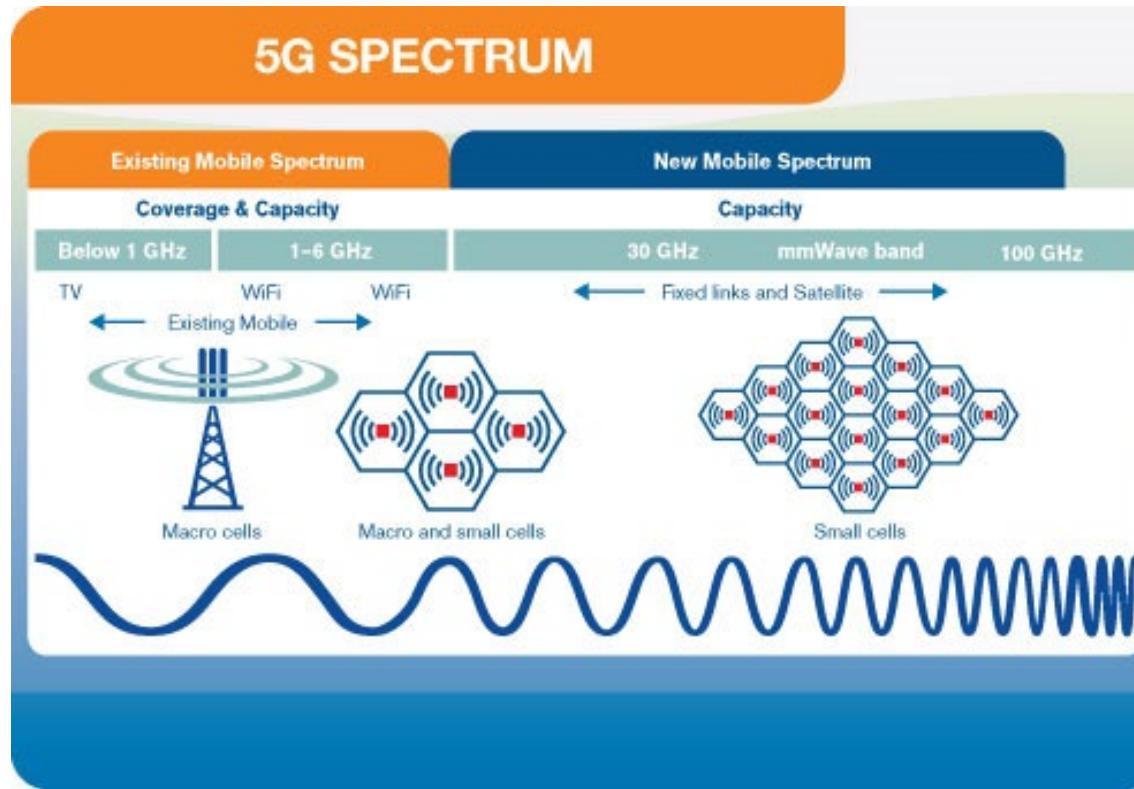


1. Millimeter Wave
2. Massive MIMO
3. Small Cell
4. Beamforming
5. NOMA
6. MEC



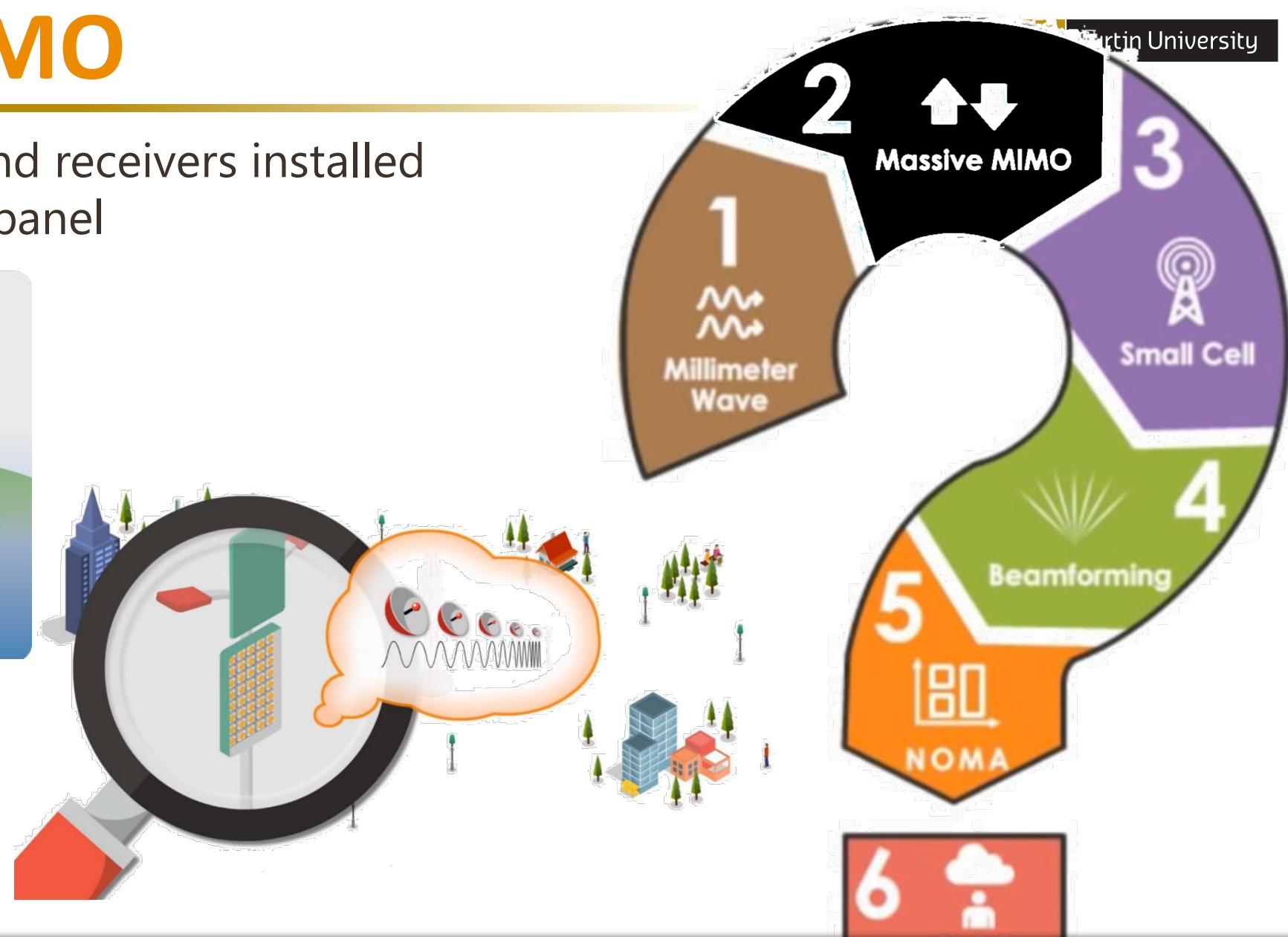
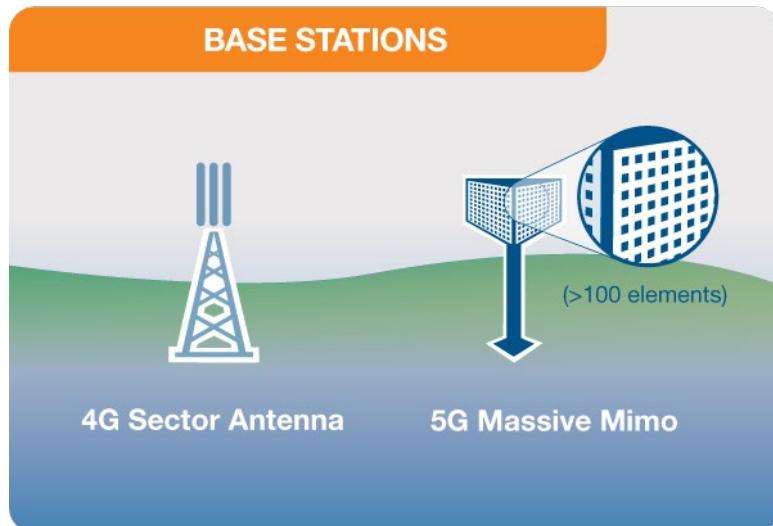
Millimeter Wave

- New and less used band
- Higher frequency wave carry more data
- Supports having a massive MIMO antenna



Massive MIMO

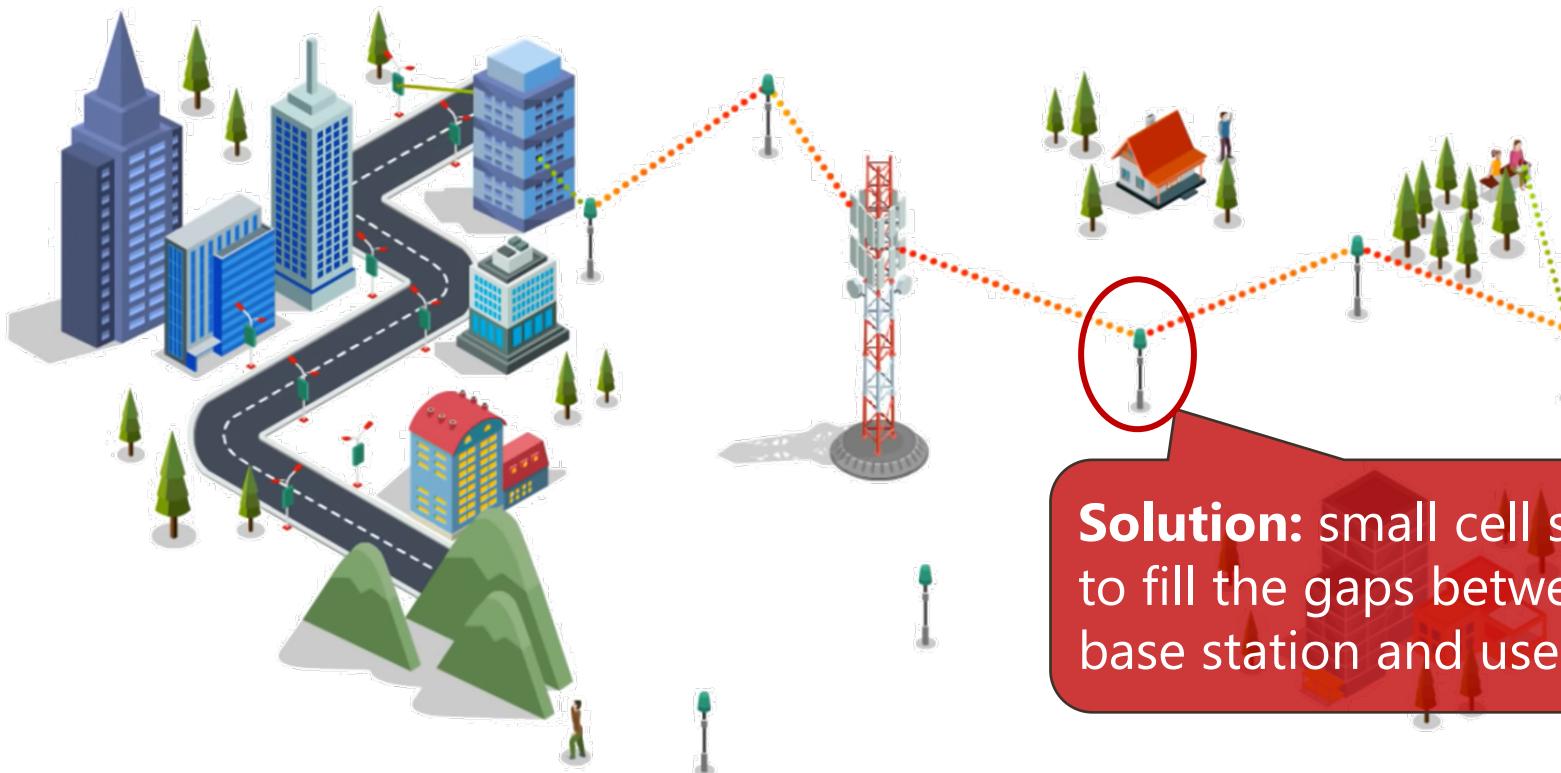
- A lot of transmitters and receivers installed on a small size cell or panel



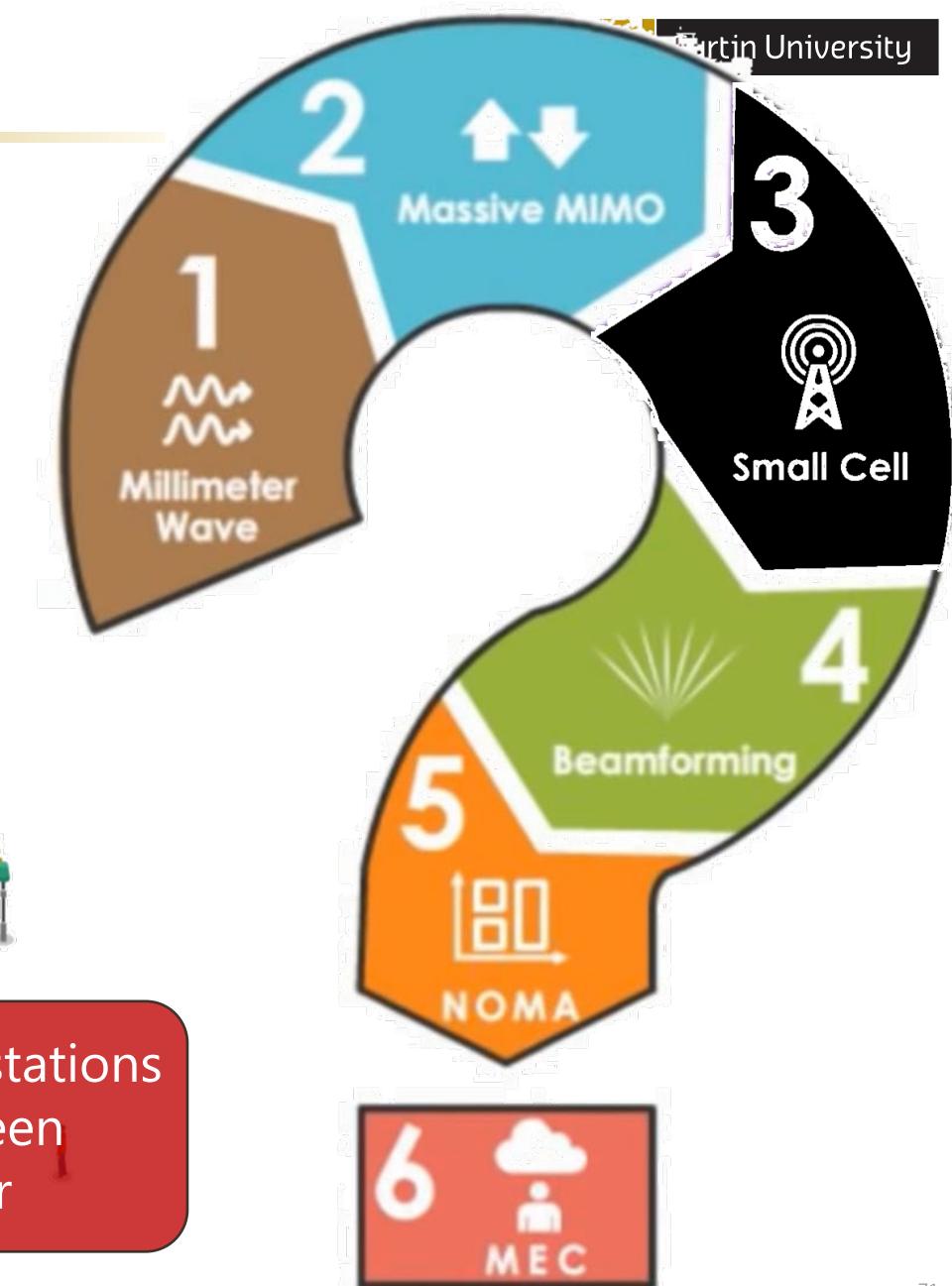
MIMO: Multiple Input Multiple Output

Small Cell

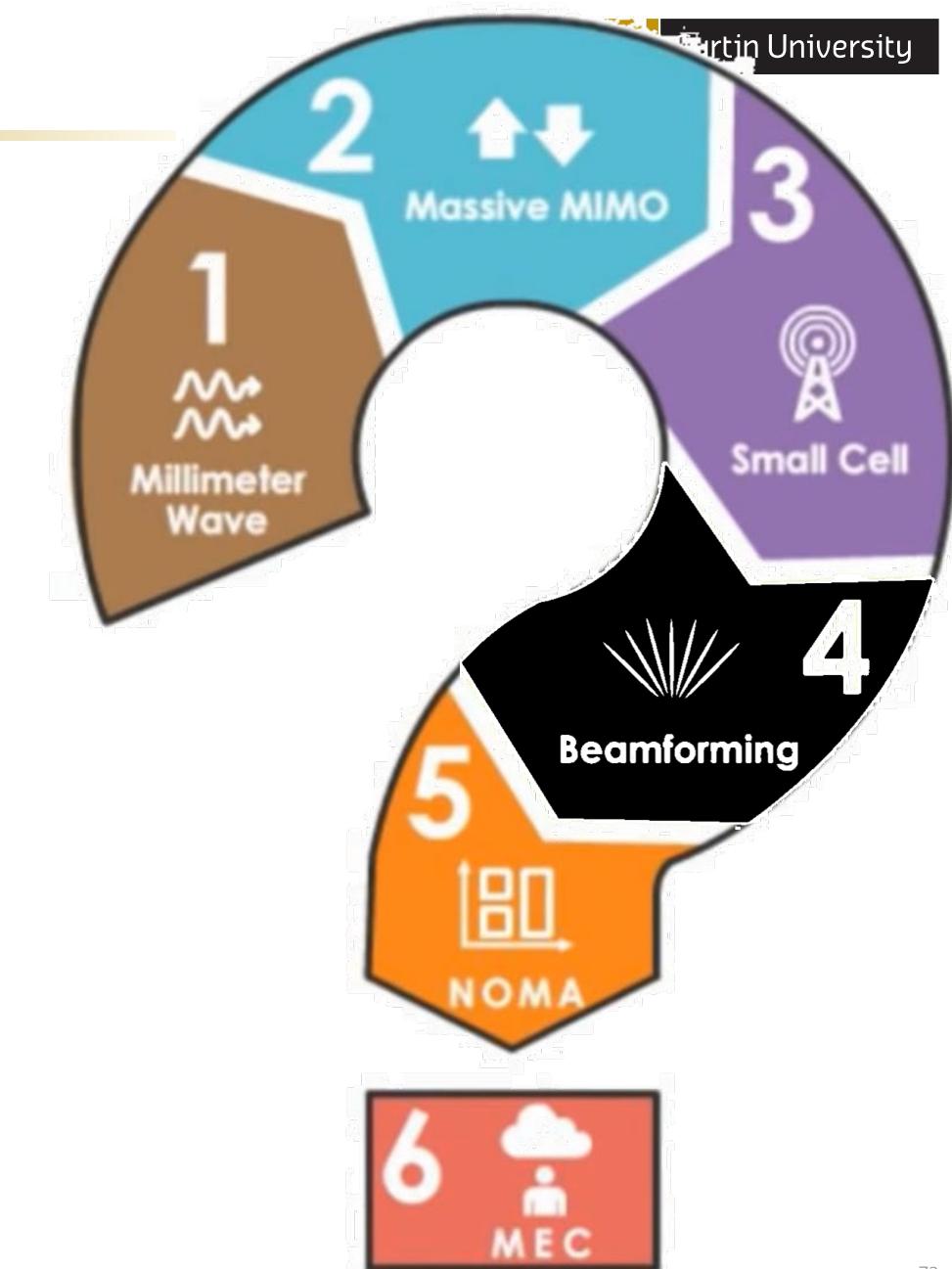
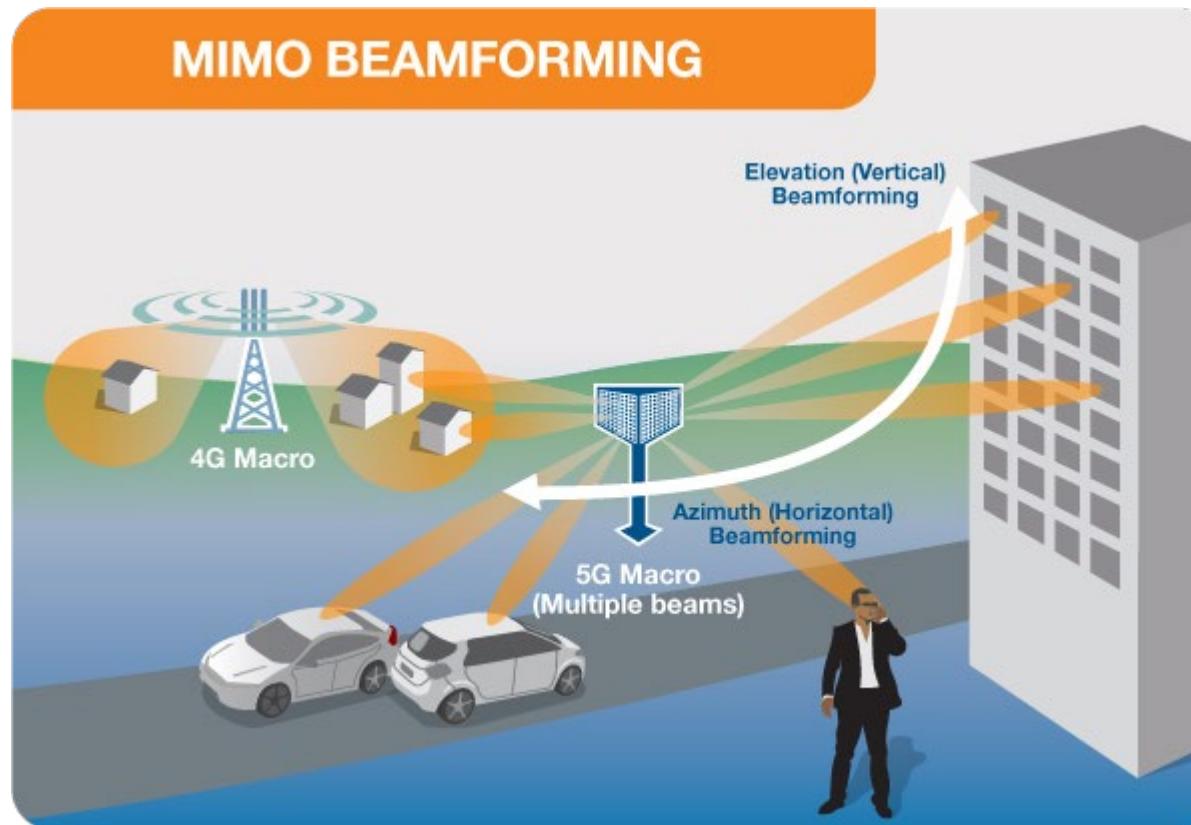
- **High freq. signals** have **more collisions** with obstacles in the air/on the ground
- Thus able to **cover shorter distance**



Solution: small cell stations to fill the gaps between base station and user

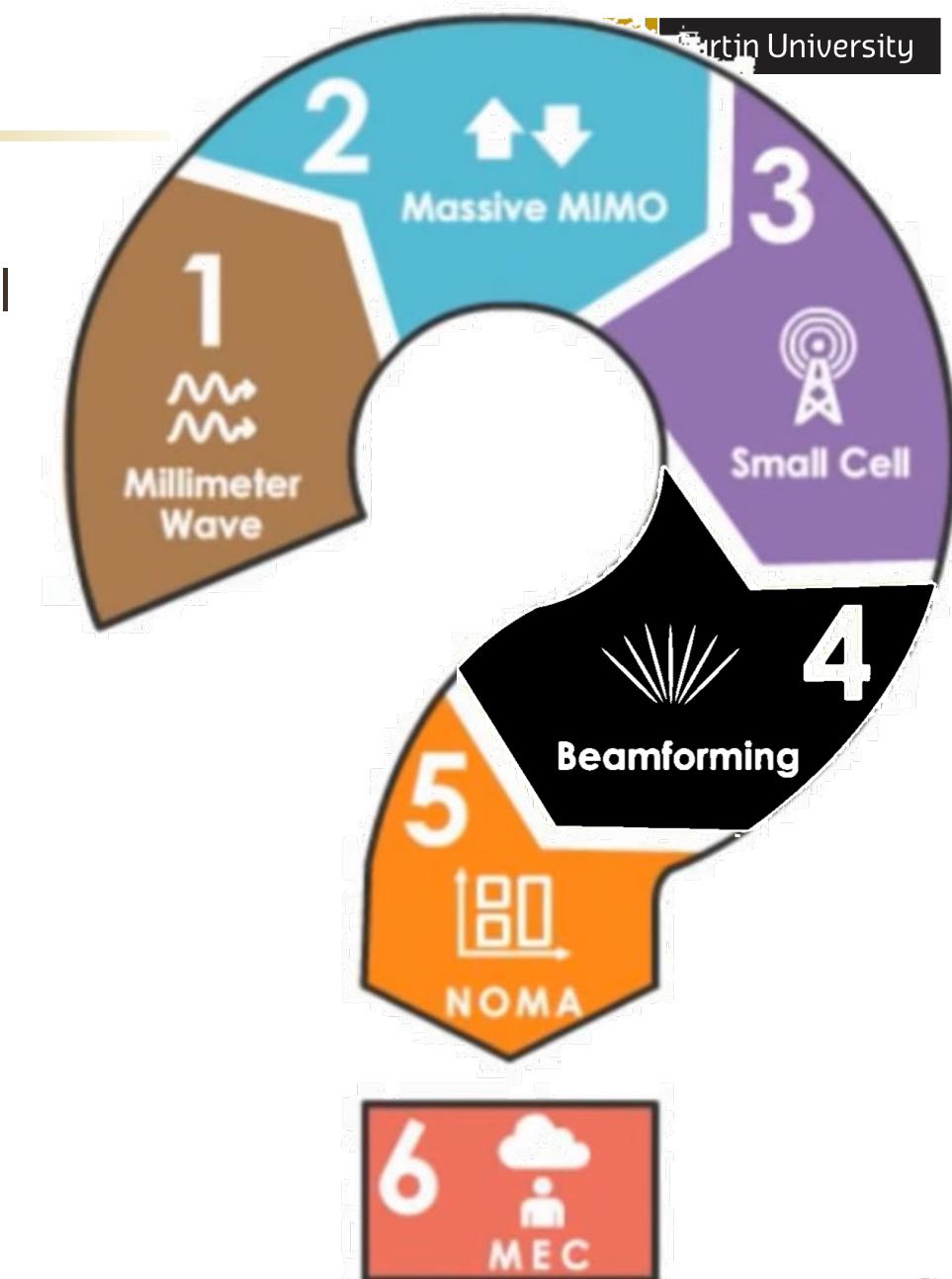
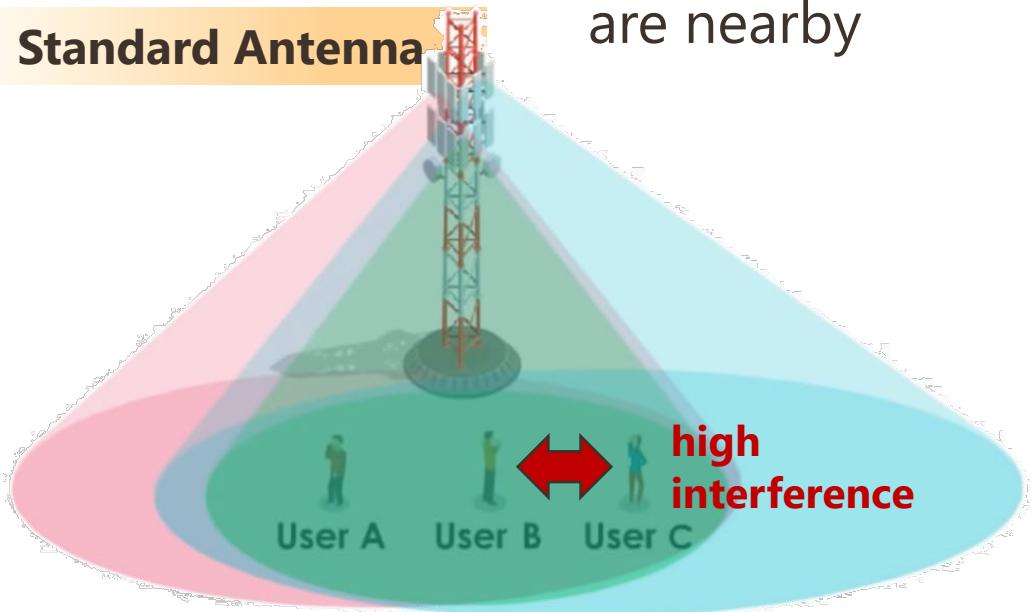


Beamforming

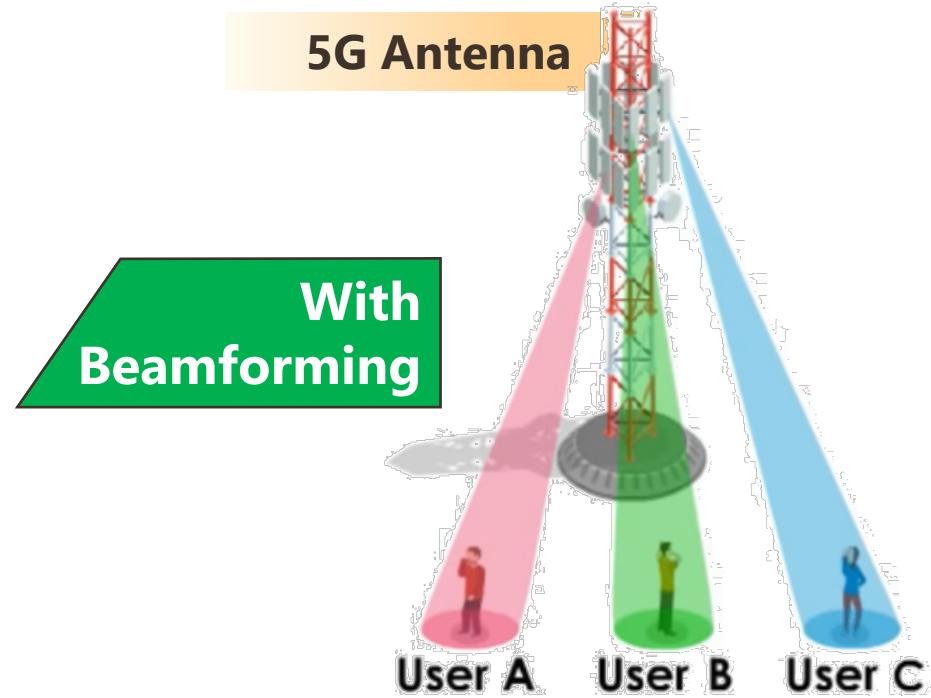


Beamforming

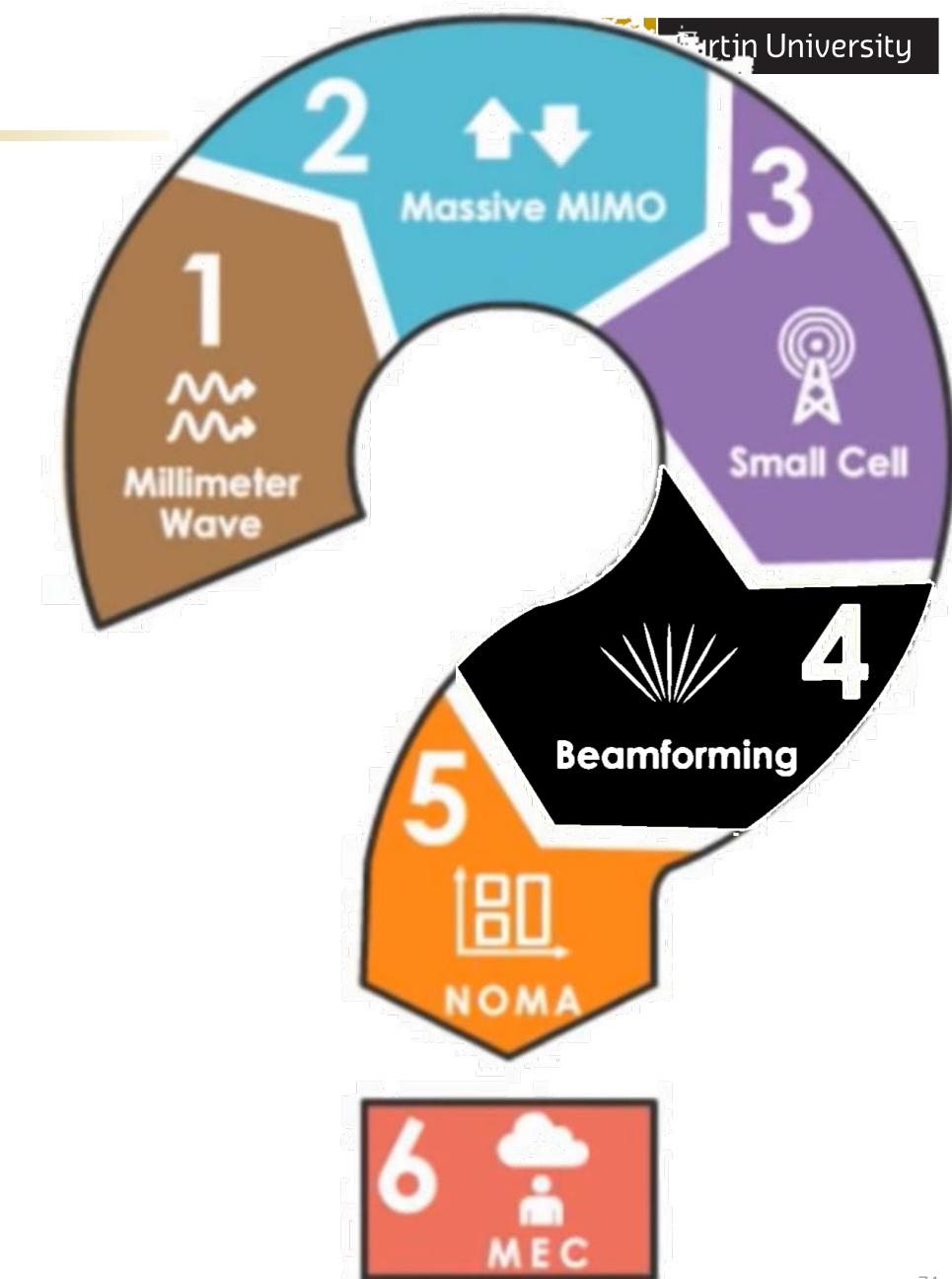
- **4g signals are omni directional**
spreads over large area as they travel
– lose energy
- **Interference is worse** when users are nearby



Beamforming

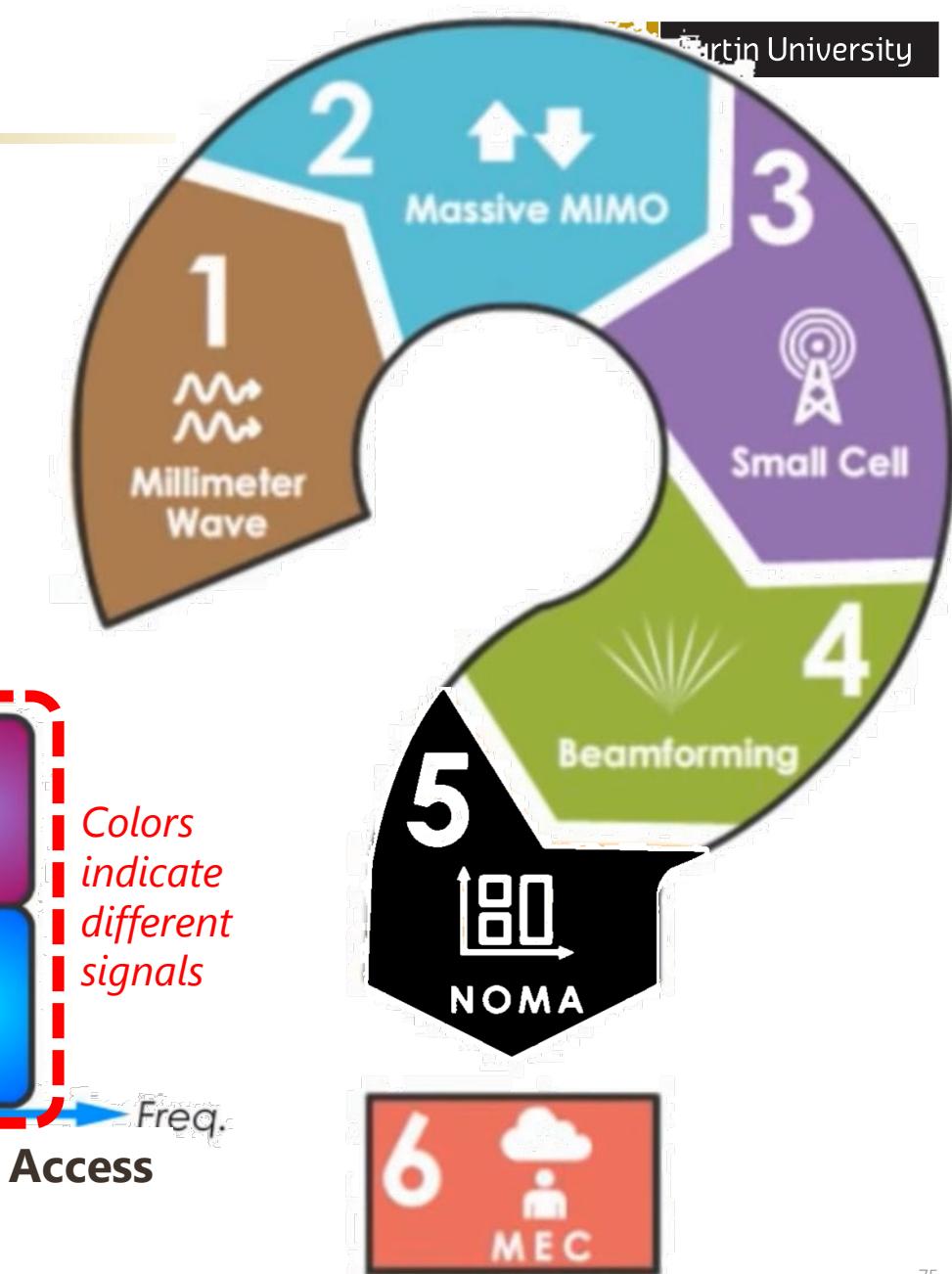
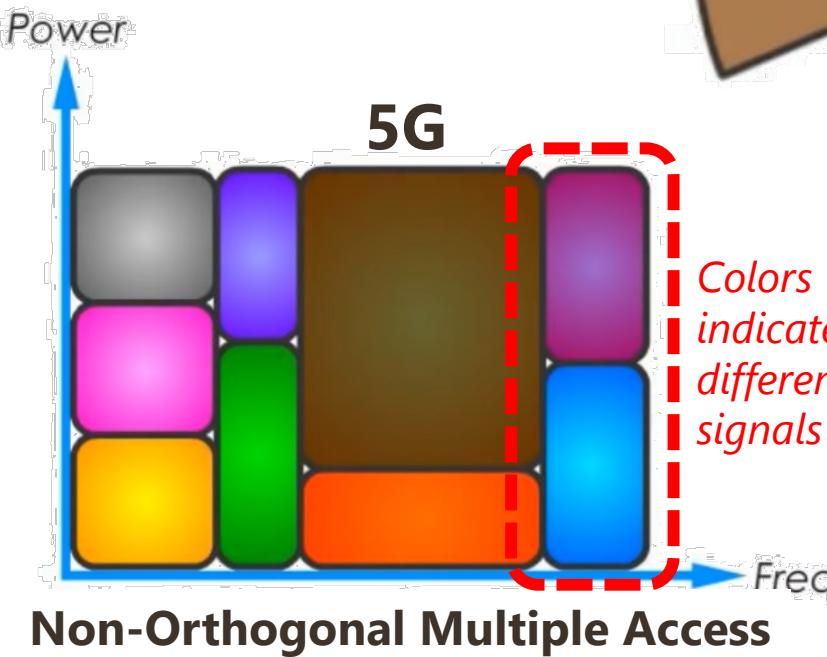
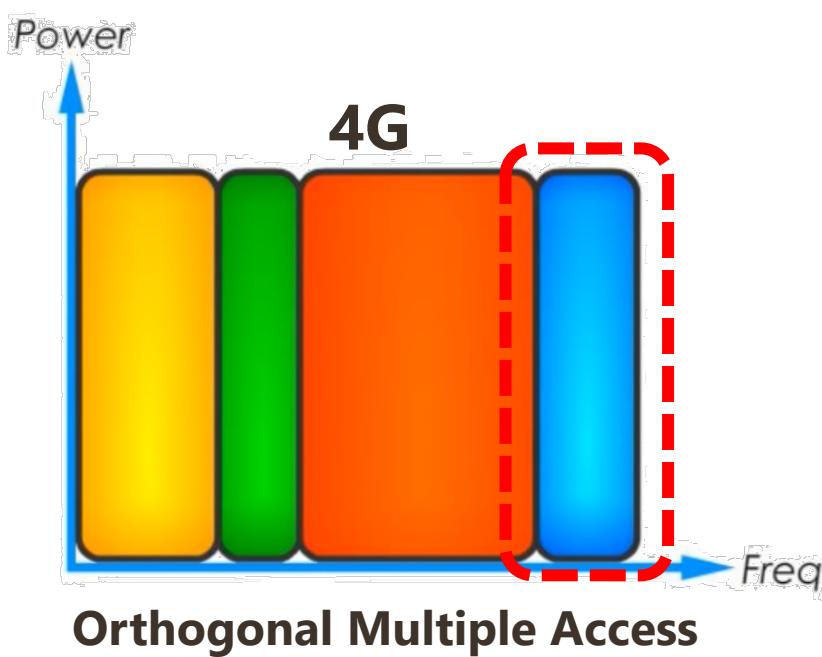


- ✓ Transmission between user and base/cell station **more directional** (i.e. laser beam)
- ✓ Less interference, less energy consumption



NOMA

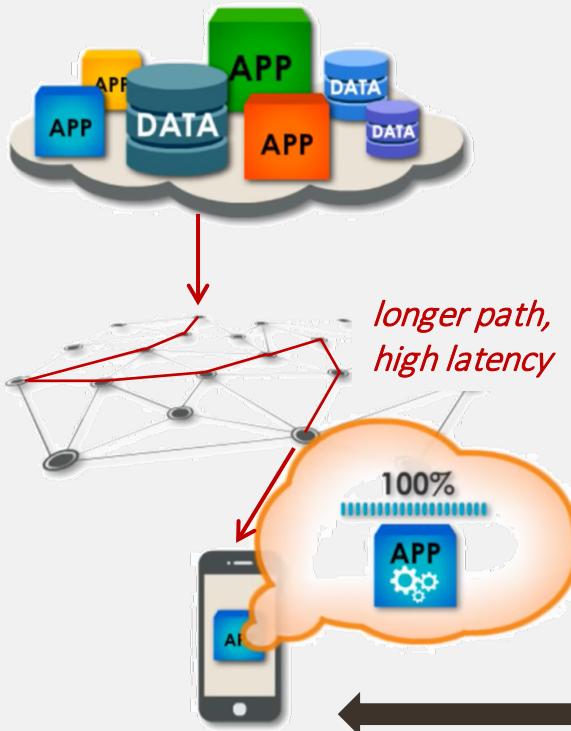
- Non-Orthogonal Multiple Access
- Allow **different signals share the same channel simultaneously**



Mobile Edge Computing (MEC)

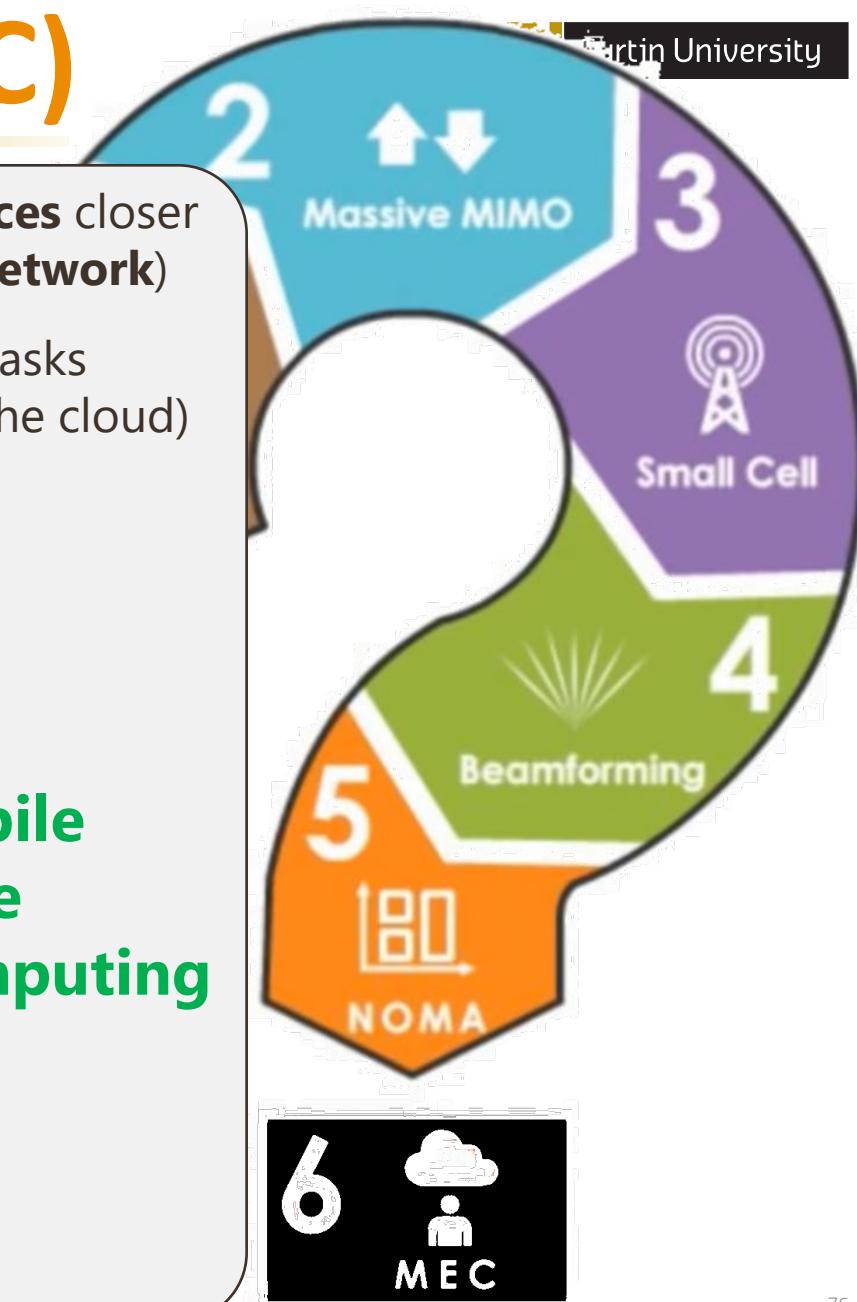
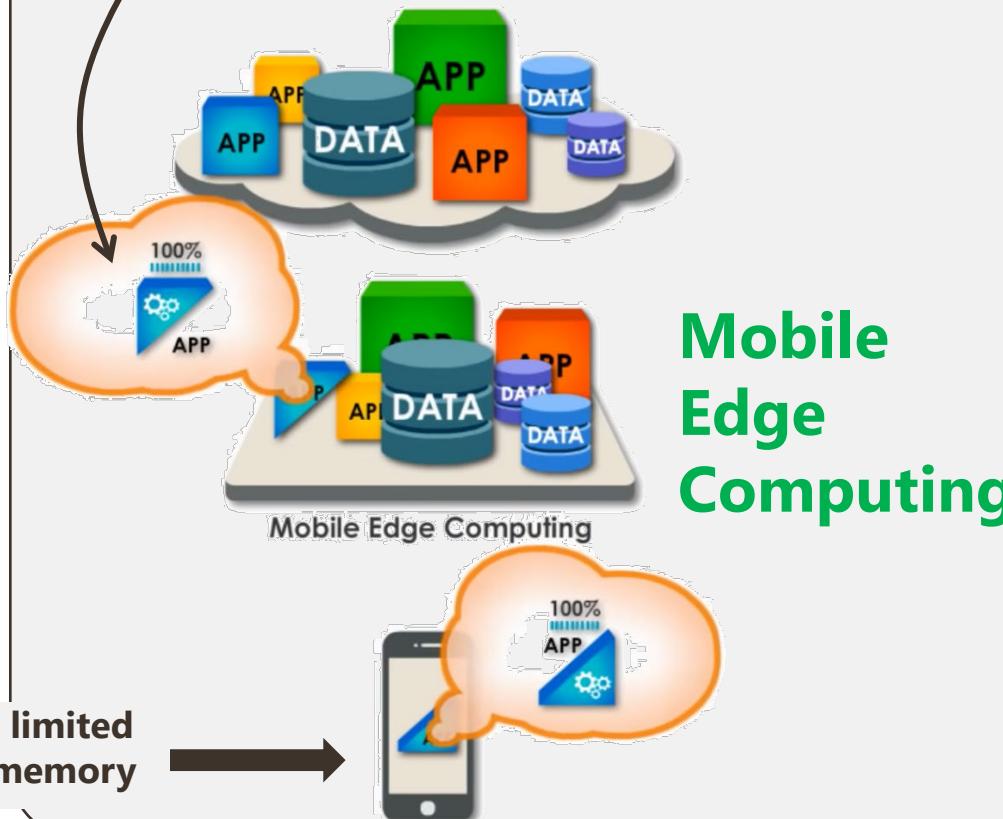
Cloud Computing

- Cloud is far away from users
- Users required to download many data and apps



Move cloud computing and services closer to the device (edge of the local network)

Application splitting (some tasks performed at device and some at the cloud)





▪ **Physical Layer**

- Fundamentals
- Services

▪ **Signals**

- Analog vs Digital
- Analog to Digital vice versa
- Signal Attenuation and Amplification

▪ **Digital Encoding**

- Bit Encoding
- Manchester Encoding
- Differential Manchester Encoding
- MLT-3

▪ **Medium Capacity**

- Bandwidth, Speed, Lag, Throughput
- Multiplexing
 - FDM
 - TDM

▪ **Network Topologies**

- Physical topology
- Logical topology
- Hybrid topology

▪ **Transmission Media**

- Transmission Modes
- Simplex, Half-Duplex, Full-Duplex
- Guided Media
- Unguided Media

▪ **Last Mile Technologies**

- Dial-up
- ISDN
- DSL
- NBN
- Cellular Networks
 - 1G, 2G, 3G, 4G, 5G
 - 5G Technologies

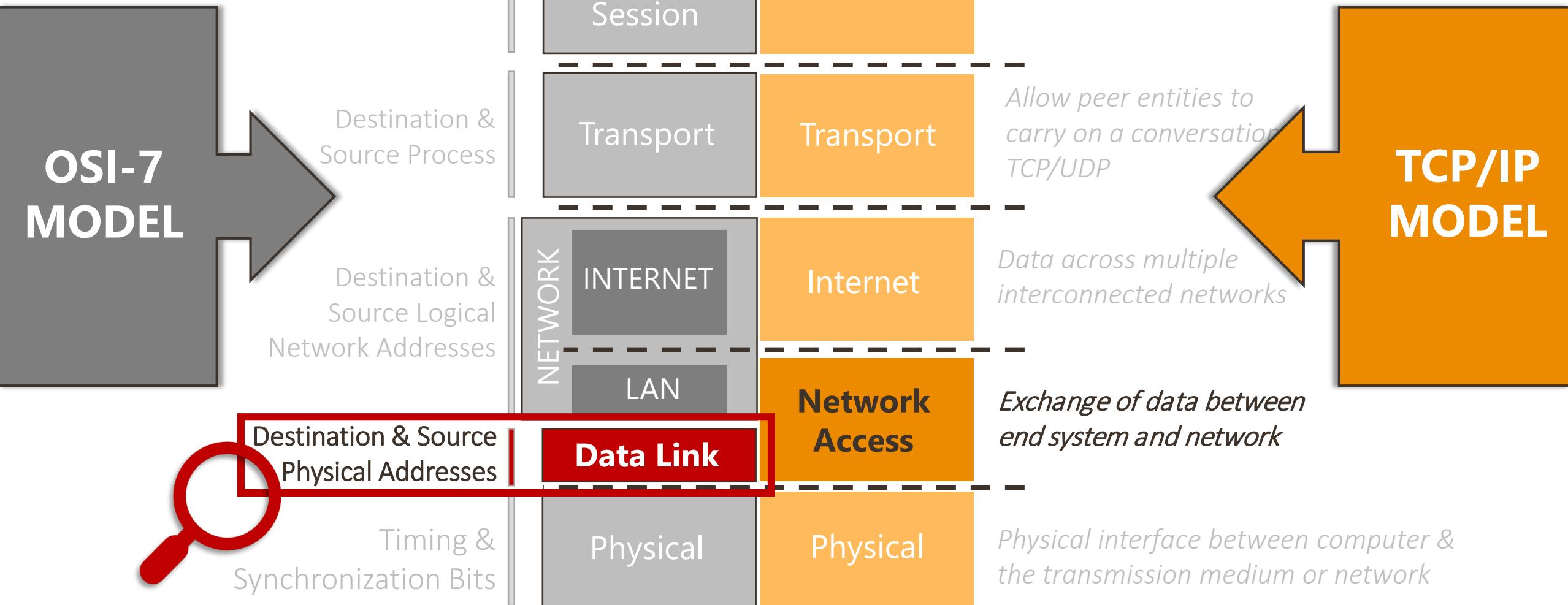
THANK YOU

Make tomorrow better.

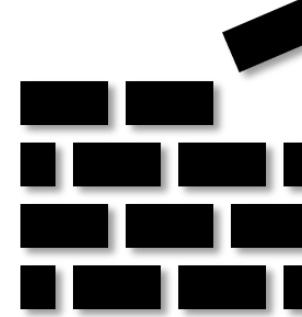
Data Link Layer I

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 03

Semester 1, 2021



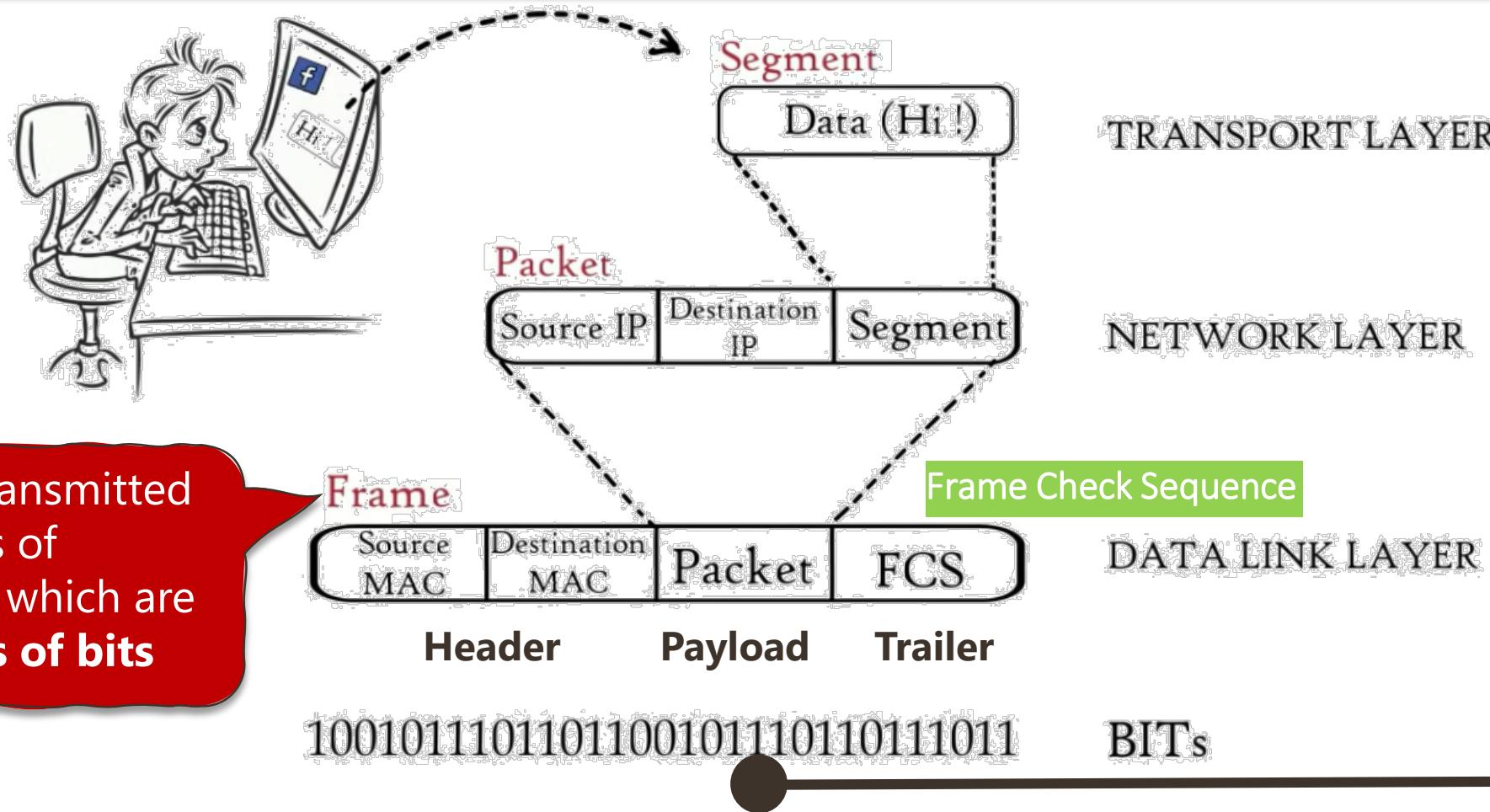
Data Link Layer



The data link layer provides the **building blocks for communication** across a variety of physical media

Data Link Layer

The data link layer is concerned with local delivery of **frames** between nodes



▪ Implementation

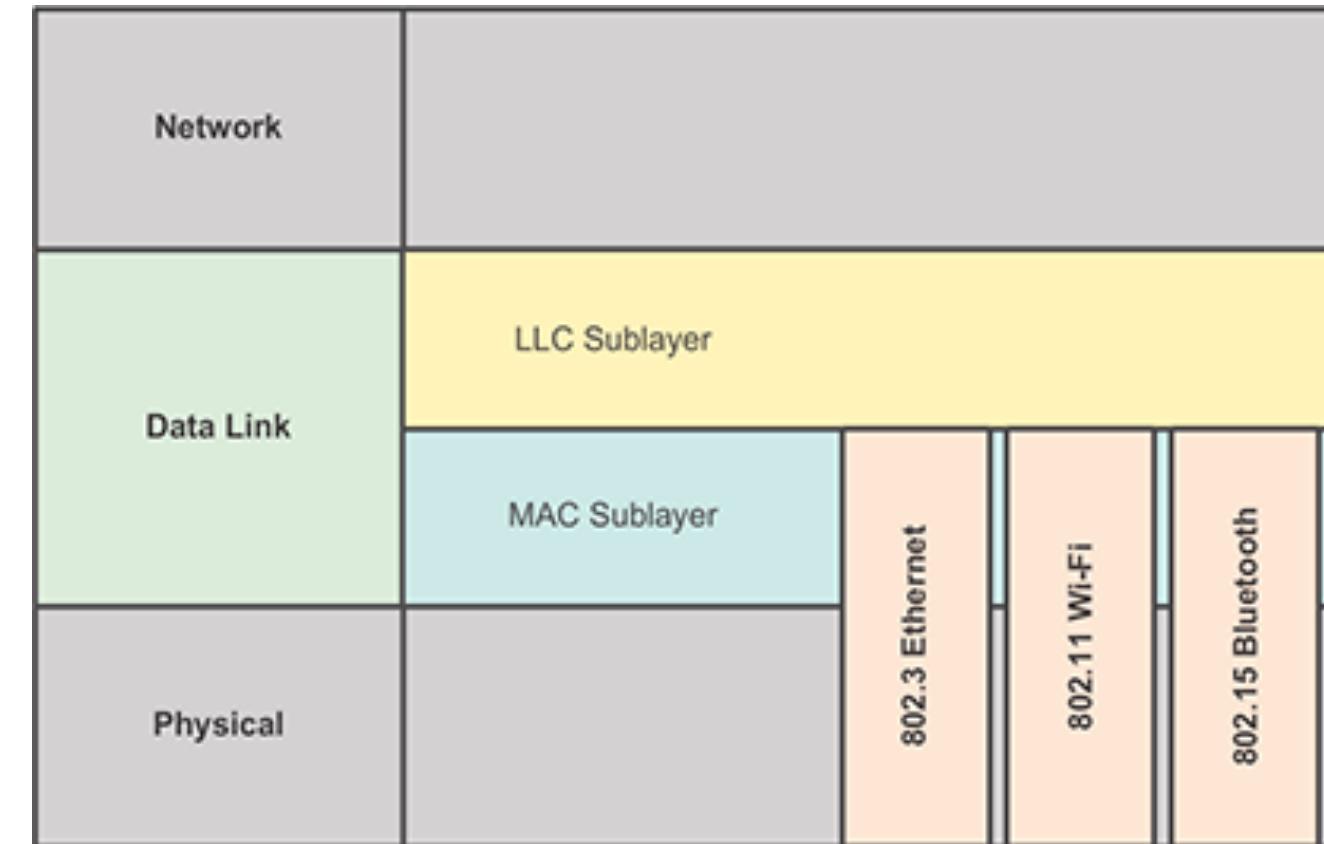
- Primarily implemented in software
- May be embedded in physical devices such as switches and network adapters (firmware)

▪ Data Link Layer Protocols

- Point-Point Protocol (PPP)
- High-Level Data Link Control (HDLC)
- IEEE 802.3 Ethernet LAN
- IEEE 802.11 (Wi-Fi) LAN

▪ Two Sub Layers

1. **LLC (Logical Link Control)**
2. **MAC (Media Access Control)**



Data Link Layer: Services

LLC:

- Provide services to network layer protocols
- **Flow Control**
- **Error Control**

MAC:

- **Framing:** bits to frame (vice versa)
- Physical addressing (**MAC addressing**)
- Multiple access methods for channel-access control (**CSMA/CD, CSMA/CA**)
- LAN switching (packet switching), including MAC filtering, **Spanning Tree Protocol (STP)**
- Data packet queuing or scheduling
- Store-and-forward switching or cut-through switching
- Quality of Service (QoS) control
- **Virtual LANs (VLAN)**



LLC Sub-Layer

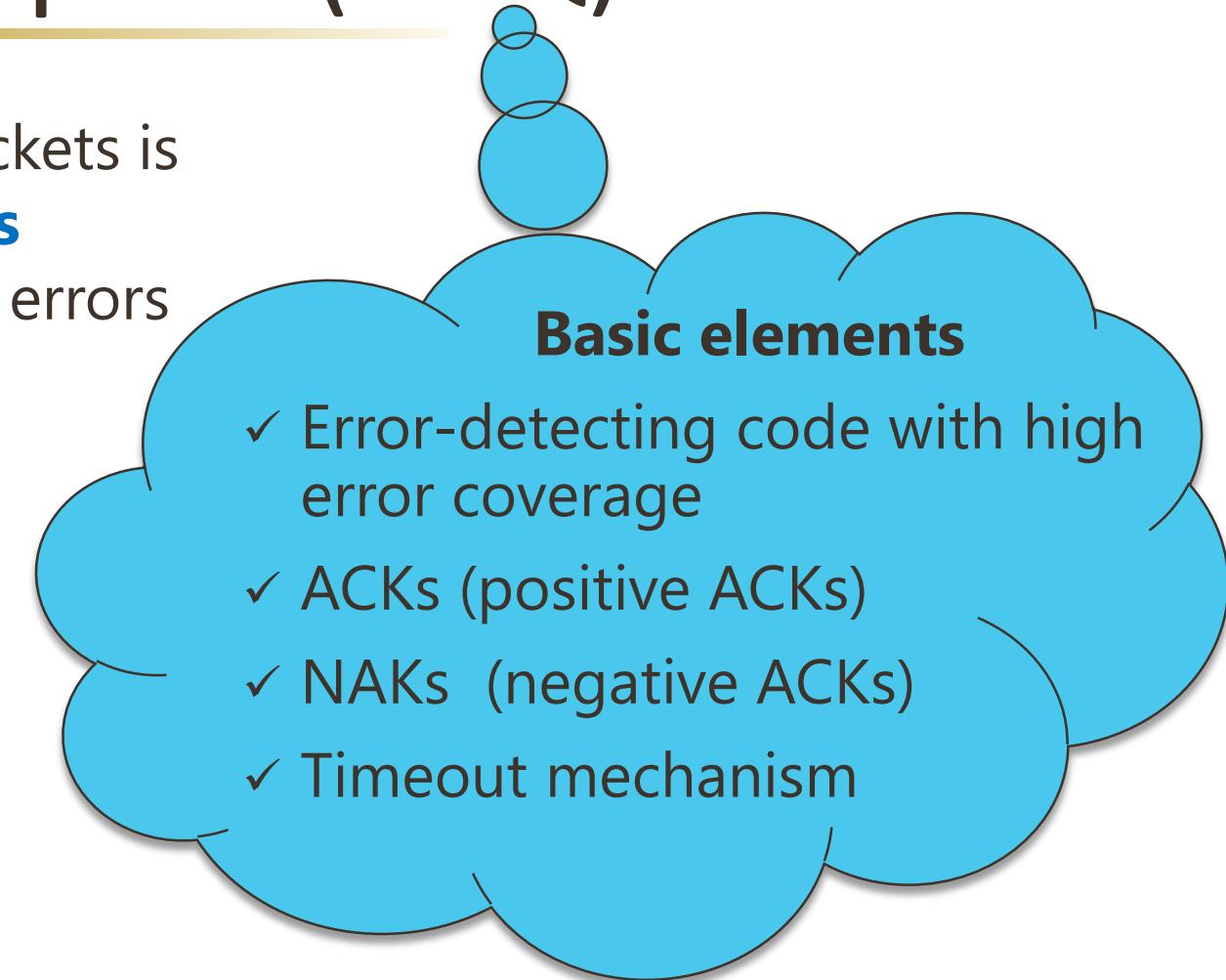
Flow Control

- Stop-And-Wait Protocol
- Sliding Window Protocols
 - Go-Back-N ARQ
 - Selective-Reject ARQ

Automatic Repeat Request (ARQ)

- Ensure a sequence of information packets is delivered **in order** and **without errors** or **duplications** despite transmission errors & losses

1. **Stop-and-Wait Protocol/ARQ**
2. **Sliding Windows Protocol**
 - **Go-Back N ARQ**
 - **Selective Reject ARQ**



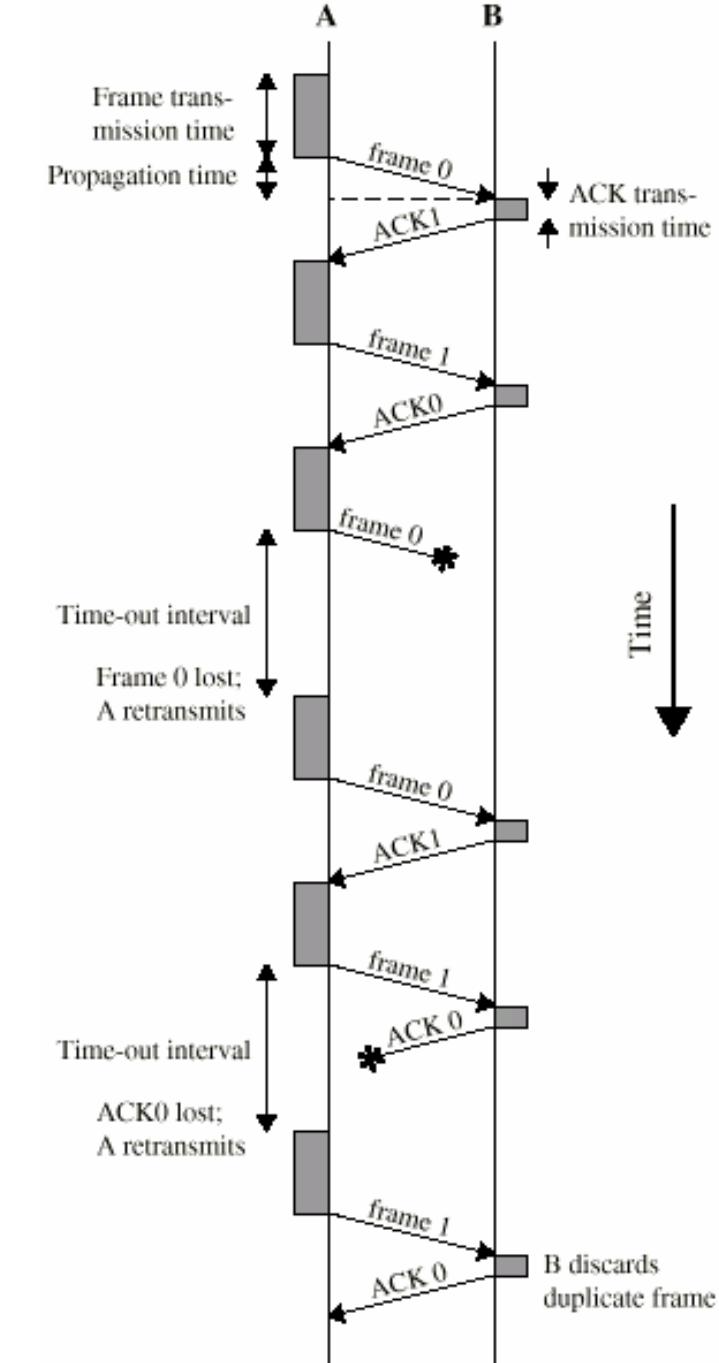
Stop-and-Wait (SW) ARQ

- ✓ Simplest form of flow control

1. Source transmits a frame
2. Destination receives the frame and replies with ACK
3. Source waits for ACK before sending next frame
4. Destination can stop flow by not sending ACK

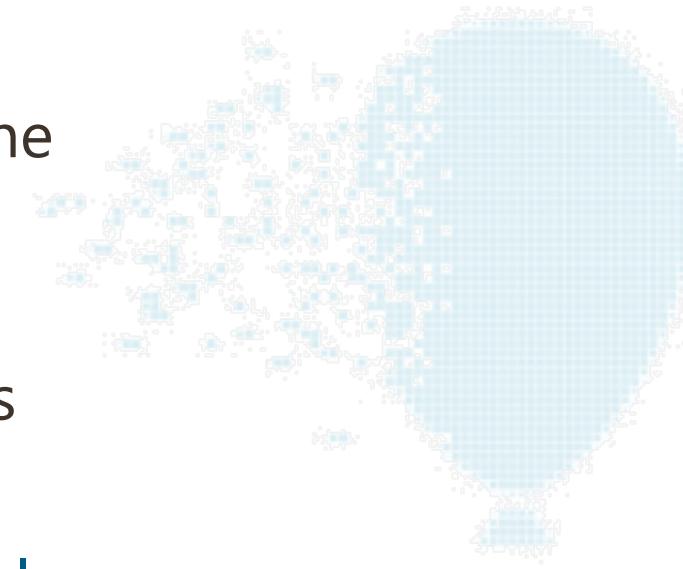
- ✓ Works well when a message is sent in a **few large frames**

However, source may break up a large block of data into smaller blocks (**Fragmentation**)

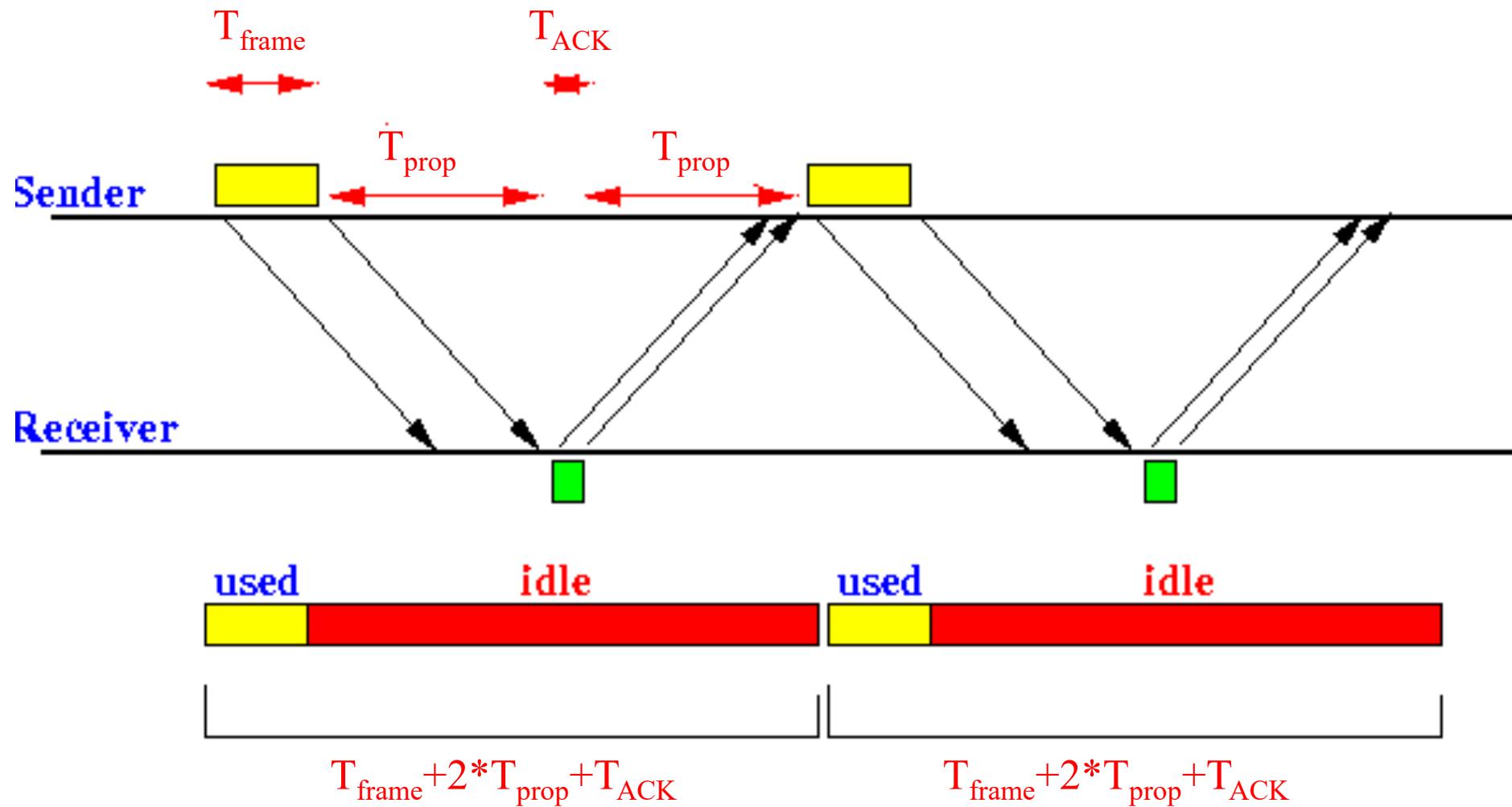


Fragmentation

- Large block of data may be split into small frames. Reasons:
 - ✓ Limited buffer size of the receiver
 - ✓ Errors detected sooner with smaller frames (the longer the transmission, the more likely there will be an error)
 - ✓ On error, retransmission of smaller frames is needed
 - ✓ Prevents one station occupying medium for long periods
- **Stop-and-wait** becomes **inadequate** with the use of multiple frames for a single message
 - *Only one frame at a time can be in transit* !



(SW) Link Utilization



(SW) Link Utilization

T_{frame} = time to transmit a frame (*time to send out all bits of the frame onto the line*)

T_{prop} = propagation time between A and B (*either direction*)

T_{ack} = time to transmit an acknowledgement

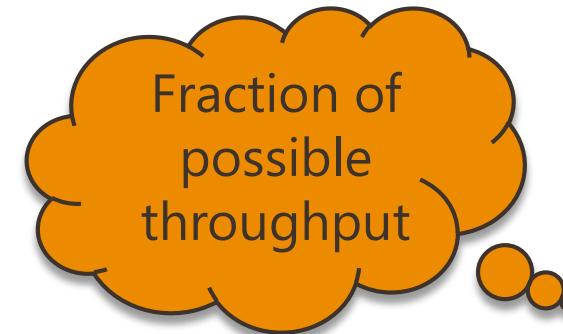
- Time to transmit one frame $\textcolor{orange}{T}$

$$\textcolor{orange}{T} = T_{frame} + 2 \times T_{prop} + T_{ack} + T_{processing}$$

$$\textcolor{orange}{T} = T_{frame} + 2 \times T_{prop} + T_{ack} = 0 \quad (<< T_{frame}), \quad T_{processing} = 0 \quad (\text{negligible})$$

- Total time to send $\textcolor{orange}{n}$ frames = nT

(SW) Link Utilization



$$\text{Throughput} = \frac{1}{T} = \frac{1}{T_{frame} + 2 \times T_{prop}}$$

Normalised

$$\text{Normalised throughput, } S = \left\{ \frac{\frac{1}{T_{frame} + 2 \times T_{prop}}}{\frac{1}{T_{frame}}} \right\}$$

Actual frame sending rate

Base frame sending rate

$$\text{Therefore, } S = \frac{T_{frame}}{T_{frame} + 2T_{prop}}$$

(SW) Link Utilization

Let $a = \frac{T_{prop}}{T_{frame}}$ then,

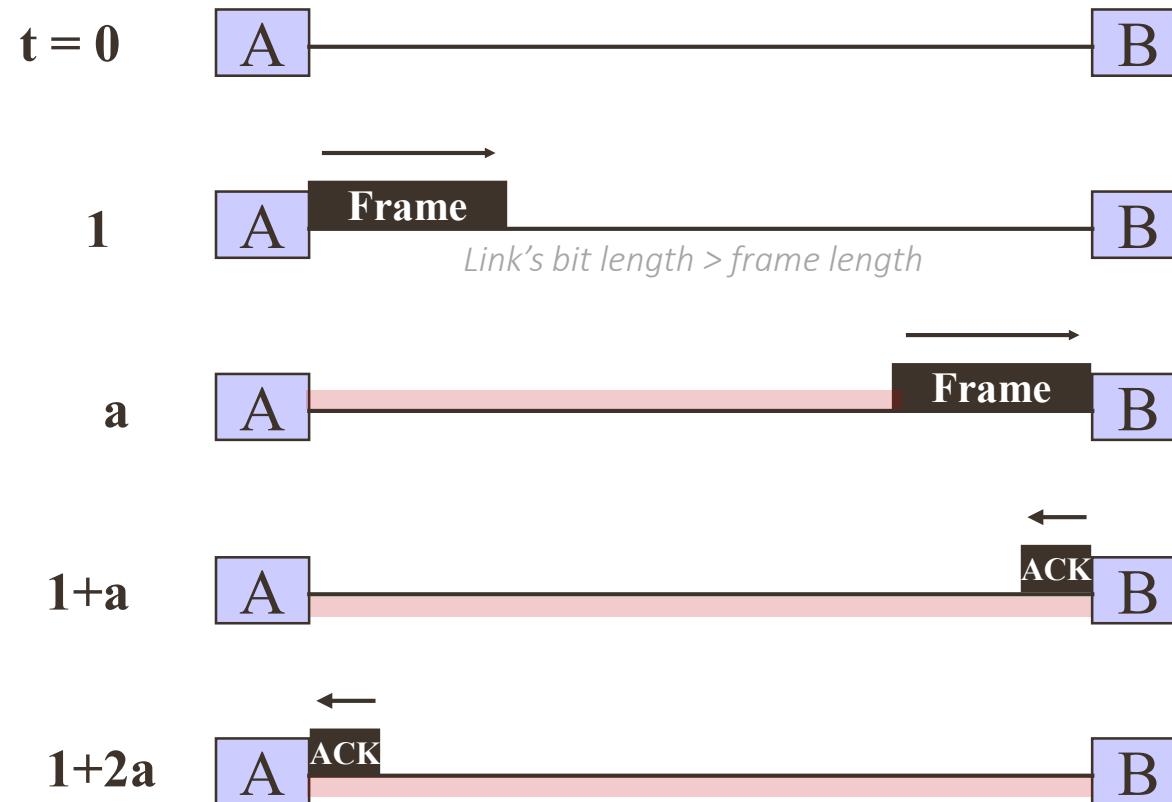
$$S = \frac{1}{1 + 2a}$$

Parameter **a**:

- Useful in **comparing the performance** of different link control schemes
- Provides insight into the factors affecting performance

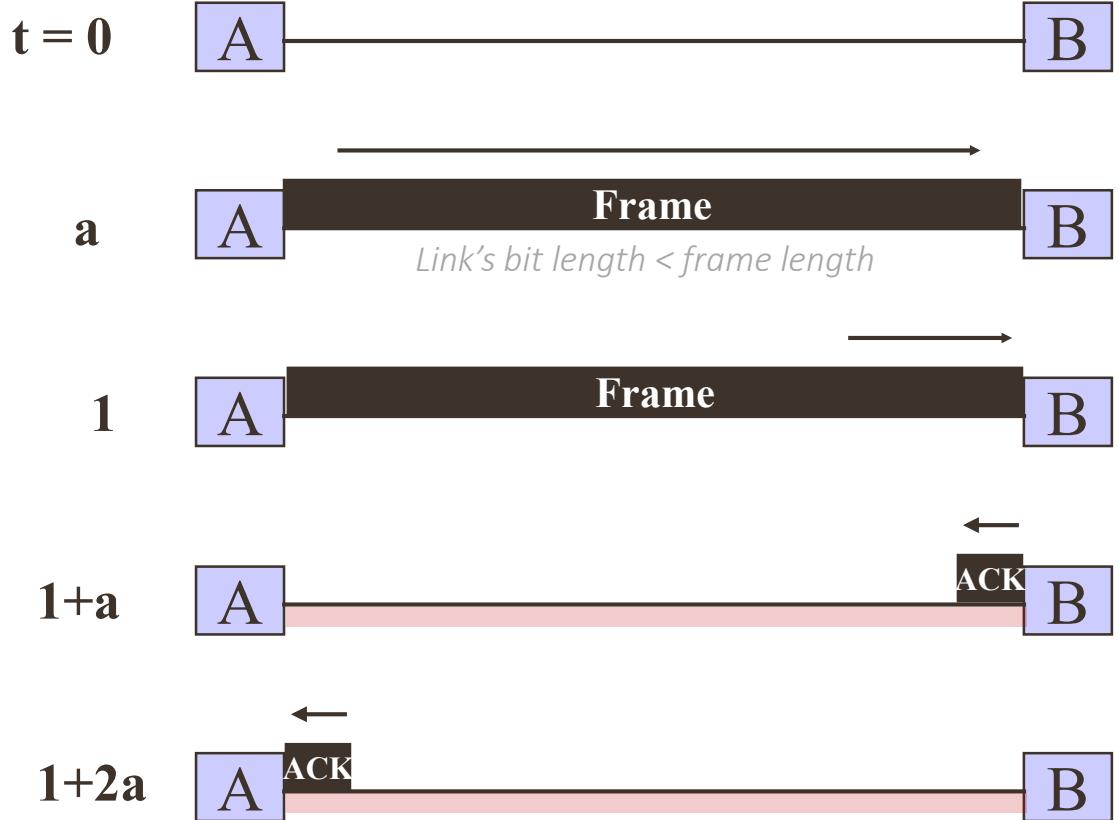
Stop-And-Wait

$a > 1, (T_{prop} > T_{frame})$



$a > 1$, Line is **under-utilized**

$a < 1, (T_{prop} < T_{frame})$

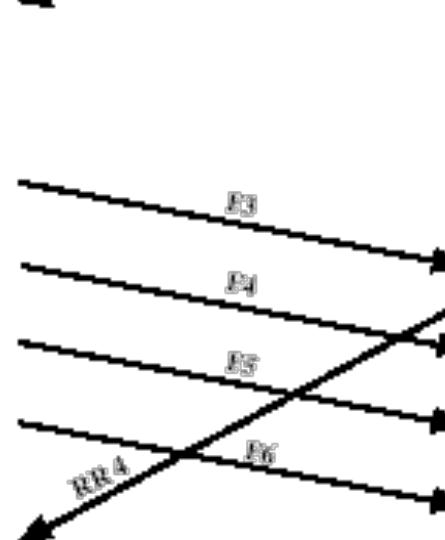
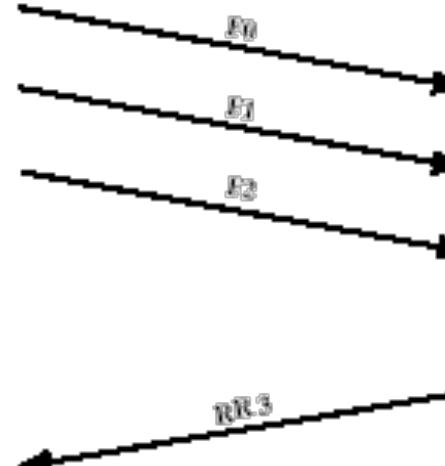
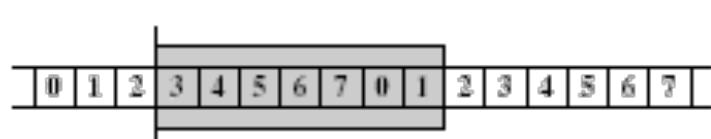
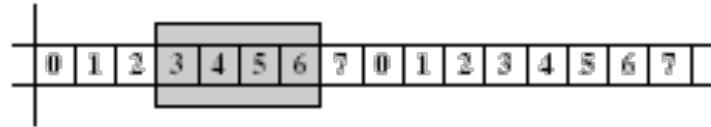
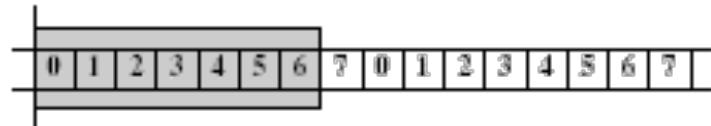
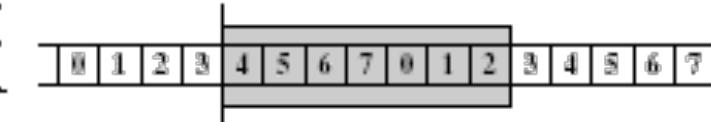
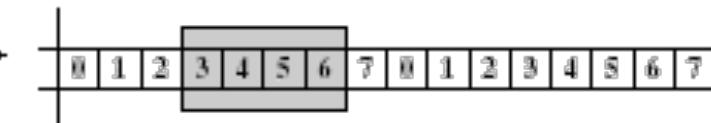
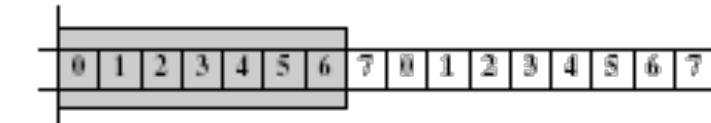


$a < 1$, Line is **better-utilized**

Sliding Window Protocols

- Stop-and-wait is **inefficient** if $T_{\text{prop}} > T_{\text{frame}}$
- Sliding Window:
 - ✓ Allowing multiple frames to propagate from A to B (*efficient*)
 - ✓ A is allowed to **send n frames** without waiting for ACKs
 - ✓ A has a **buffer** of size n
 - ✓ B also has a **buffer** of size n, and **accepts n frames**
 - ✓ Each frame is **labeled with a number** modulo some maximum

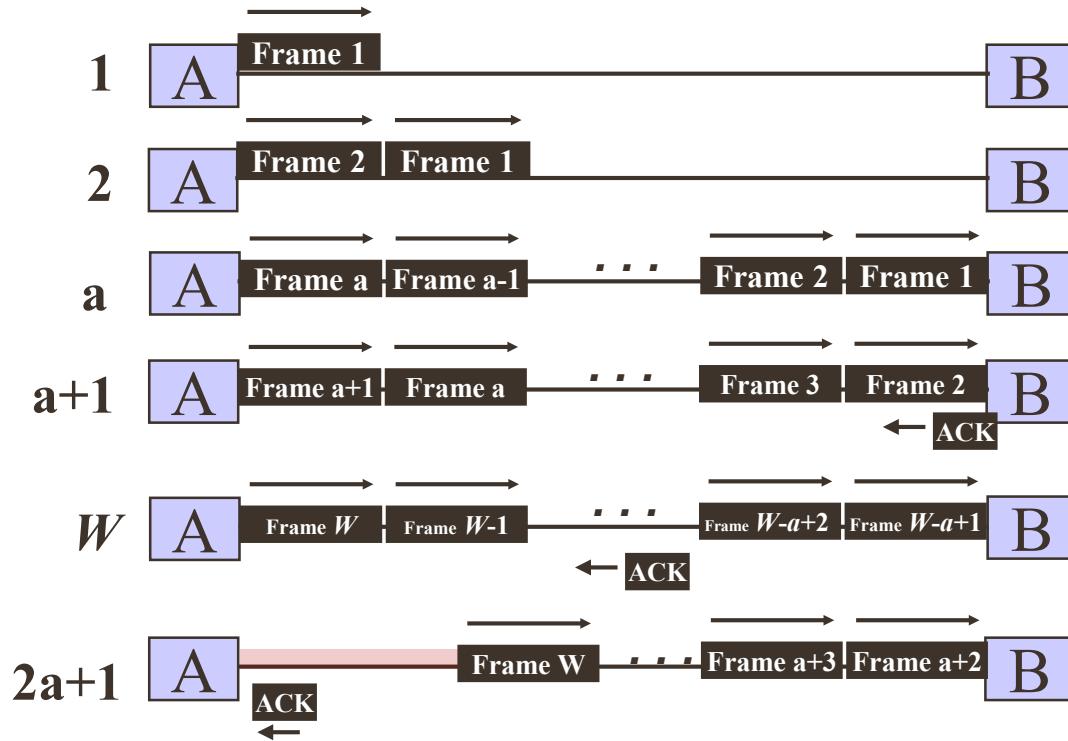
1. **Go Back N ARQ**
2. **Selective Reject ARQ**

Source System A**Destination System B**

Sliding Window

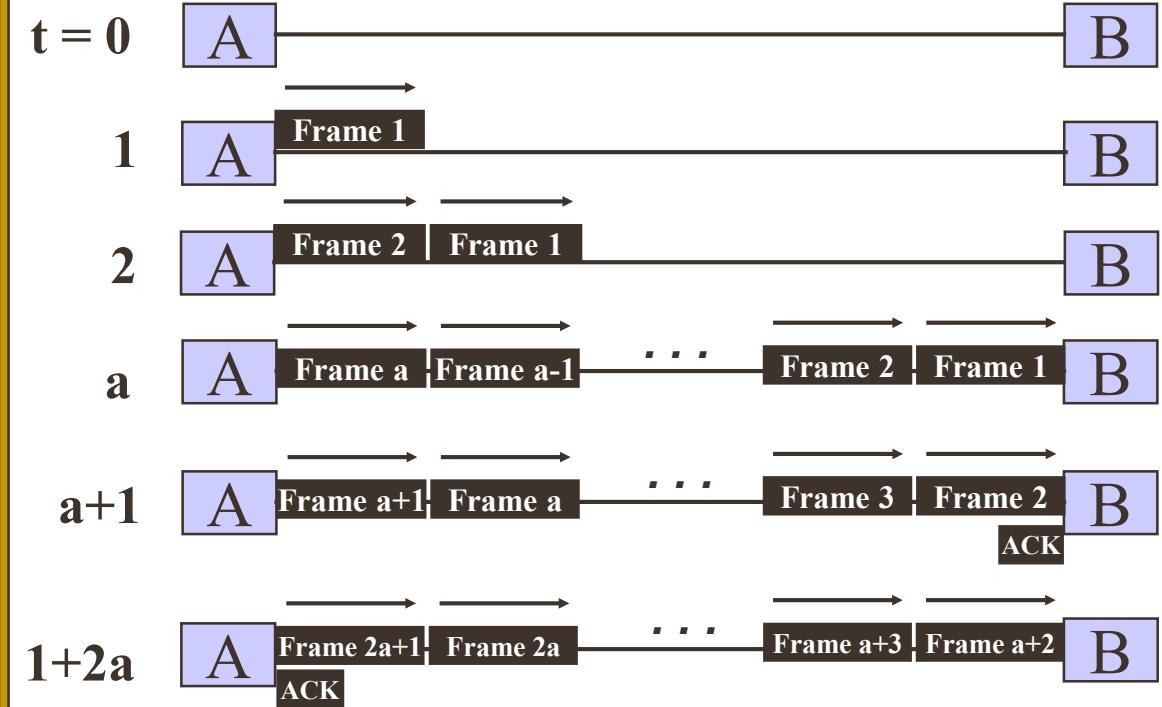
Sliding Window

$$W < 2a + 1$$



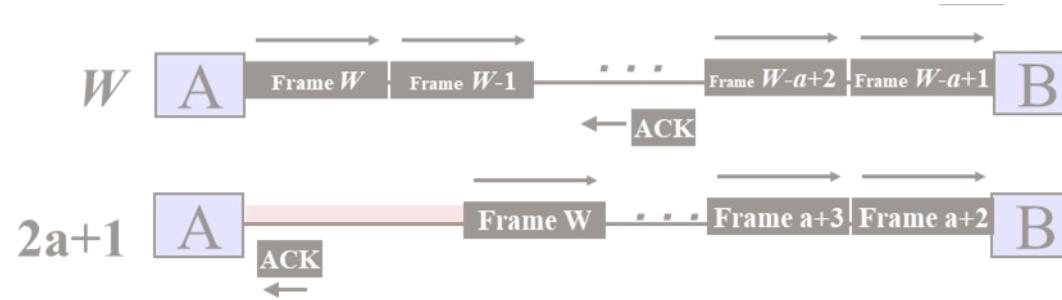
$W < 2a+1$, Line is **under-utilized**

$$W \geq 2a + 1$$



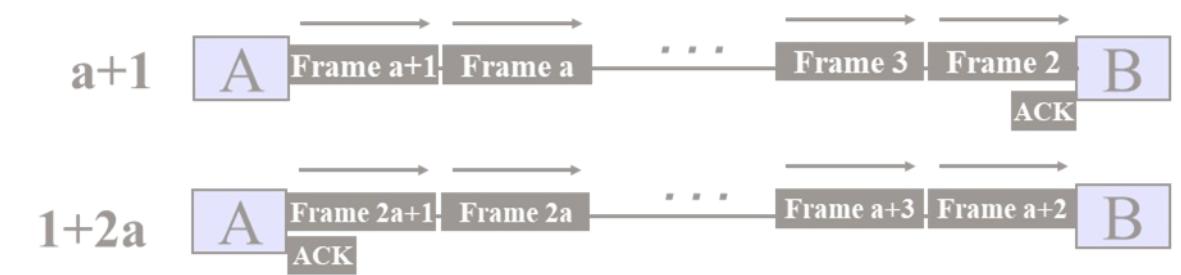
$W \geq 2a+1$, Line is **fully-utilized**

$$W < 2a + 1$$



A exhausts its window at $t = W$ and cannot send any more frames **until** $t = 2a+1$

$$W \geq 2a + 1$$



ACK reaches A **before** A has **exhausted** its window. A can transmit continuously

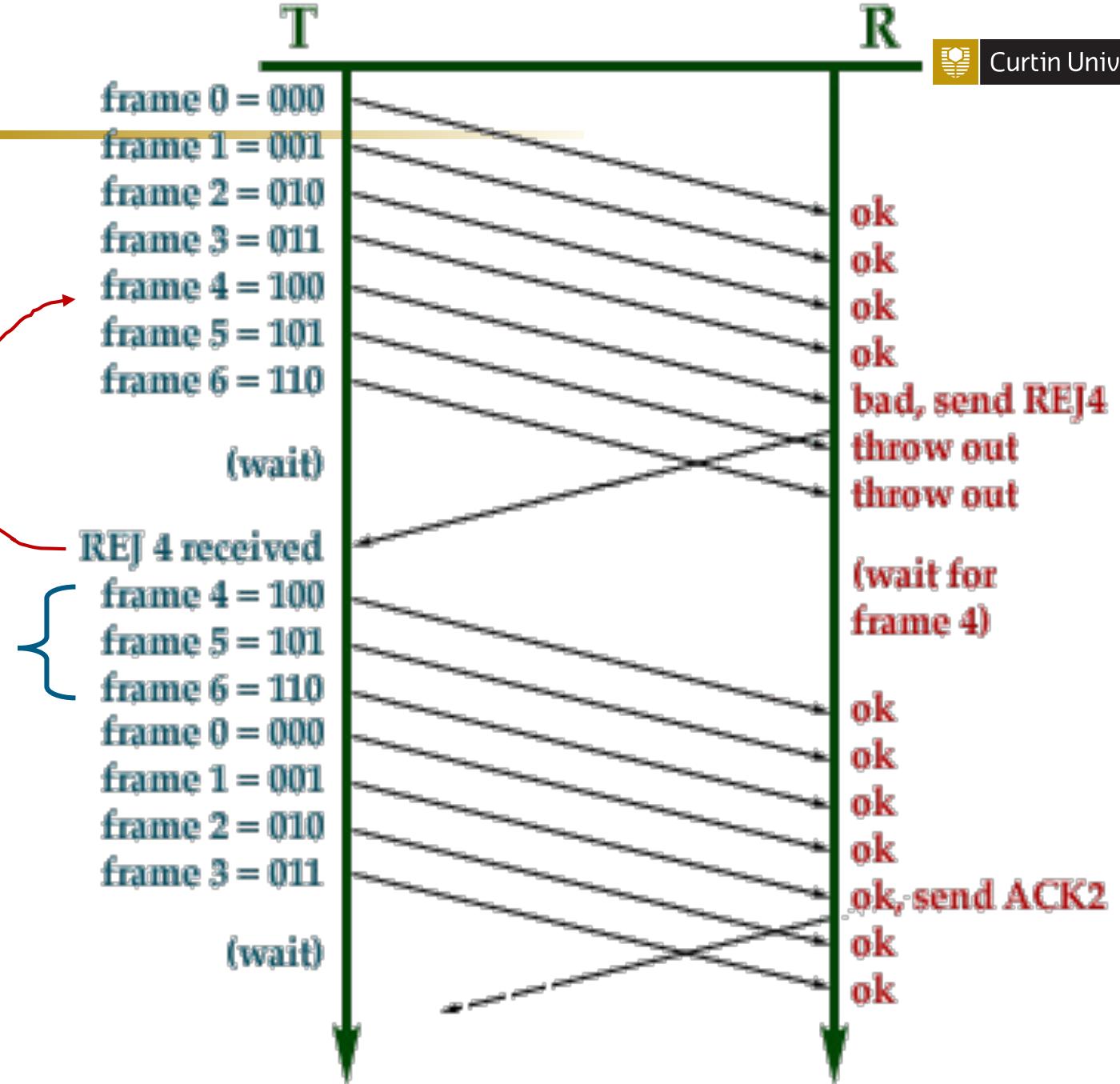
Sliding Window Link-Utilization

$$S = \begin{cases} 1, & W \geq 2a + 1 \\ \frac{W}{(2a + 1)}, & W < 2a + 1 \end{cases}$$

Go-Back-N ARQ

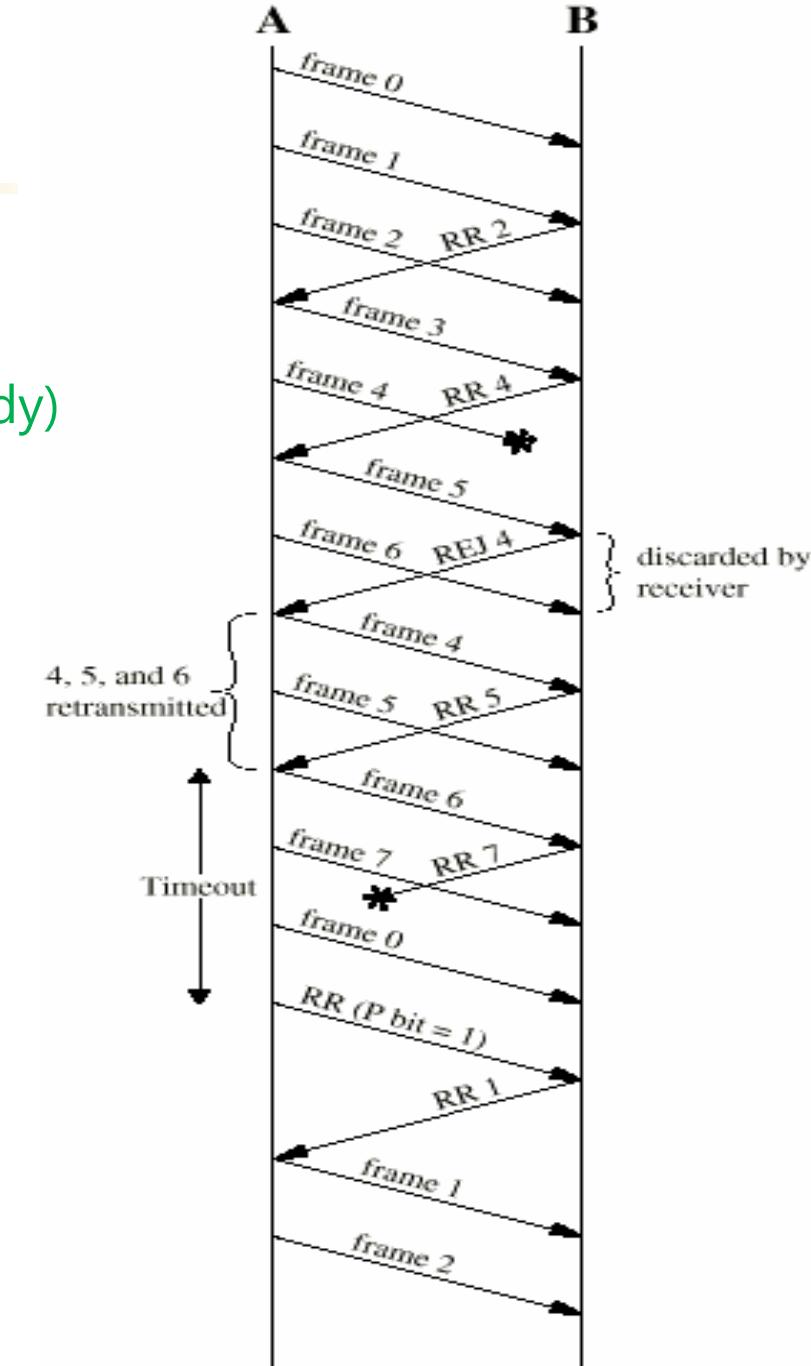
Go-Back-N Frames

Retransmits frame k (=4) and all **succeeding frames** that were transmitted (say N) in the interim period.

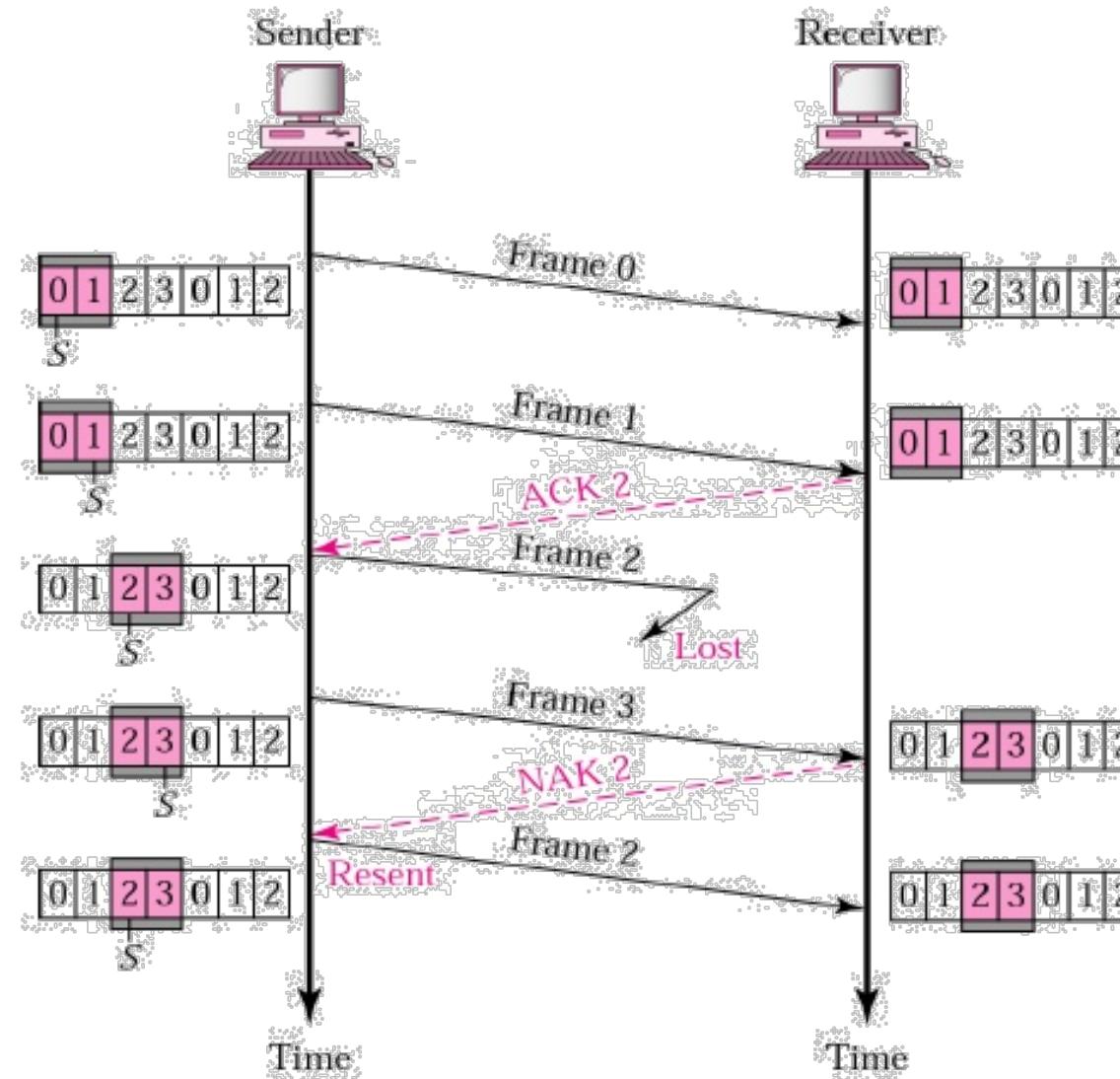


Go-Back-N ARQ – cont.

- Station A sends frames to station B.
- Station B ACKs incoming frames with an **RR** (Receive Ready)
- Suppose Frame 4 is damaged.
- Frames 5 and 6 are received **out-of-order** and are discarded by B
- When frame 5 arrives, B sends a **REJ 4**
- When the **REJ** to frame 4 is received, frames 4, 5 and 6 must be retransmitted
- Transmitter must keep a copy of all un-acked frames



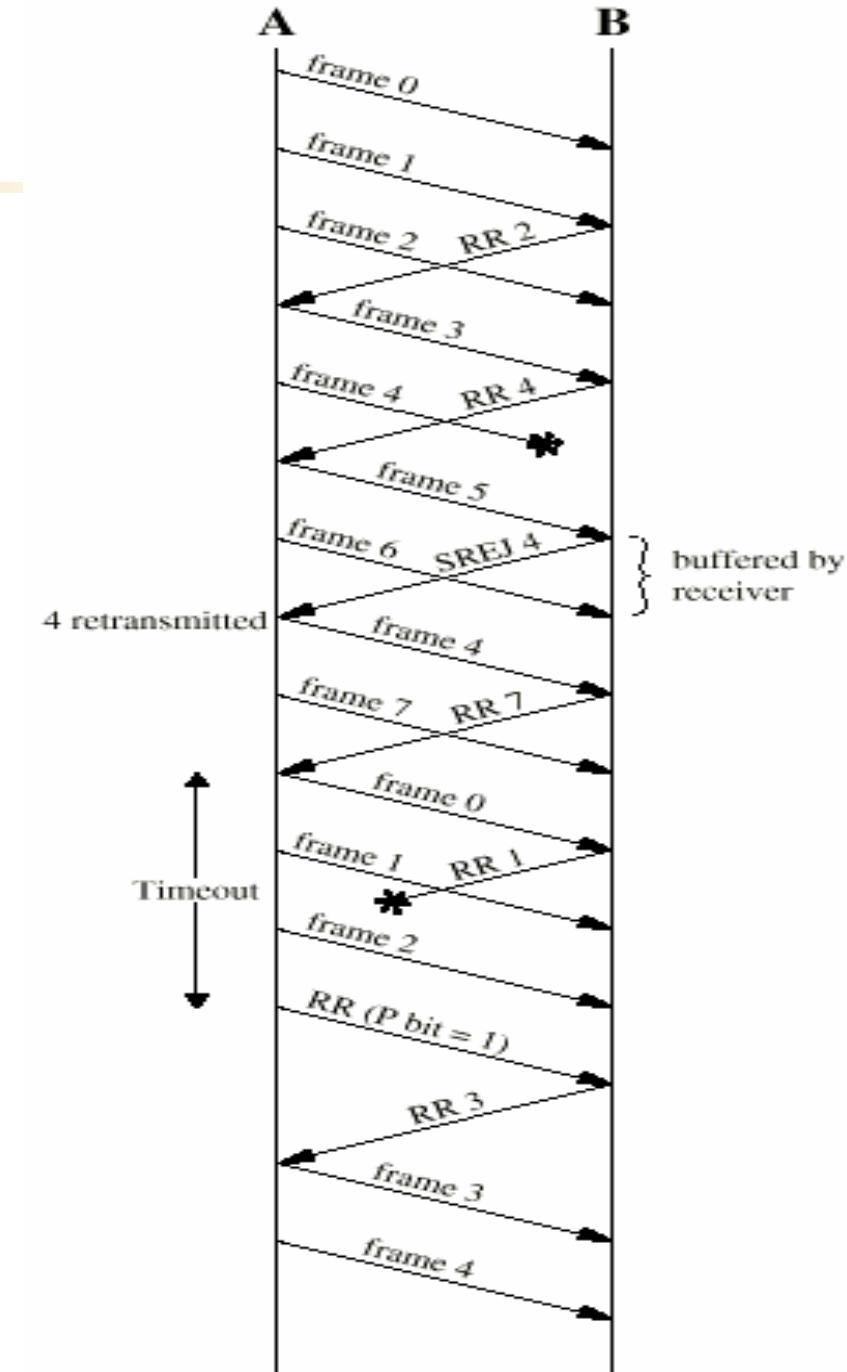
Selective-Reject ARQ



Selective-Reject ARQ

- The only frames retransmitted are those
 - that receive **NAK** (or **SREJ**)
 - that time out
- B sends **NAK *i*** if frame ***i*** is in error or lost
- B is required to **buffer frames** in order **until the correct frame arrives**

Complex logic required to send out-of-sequence frames and insert frames in appropriate places



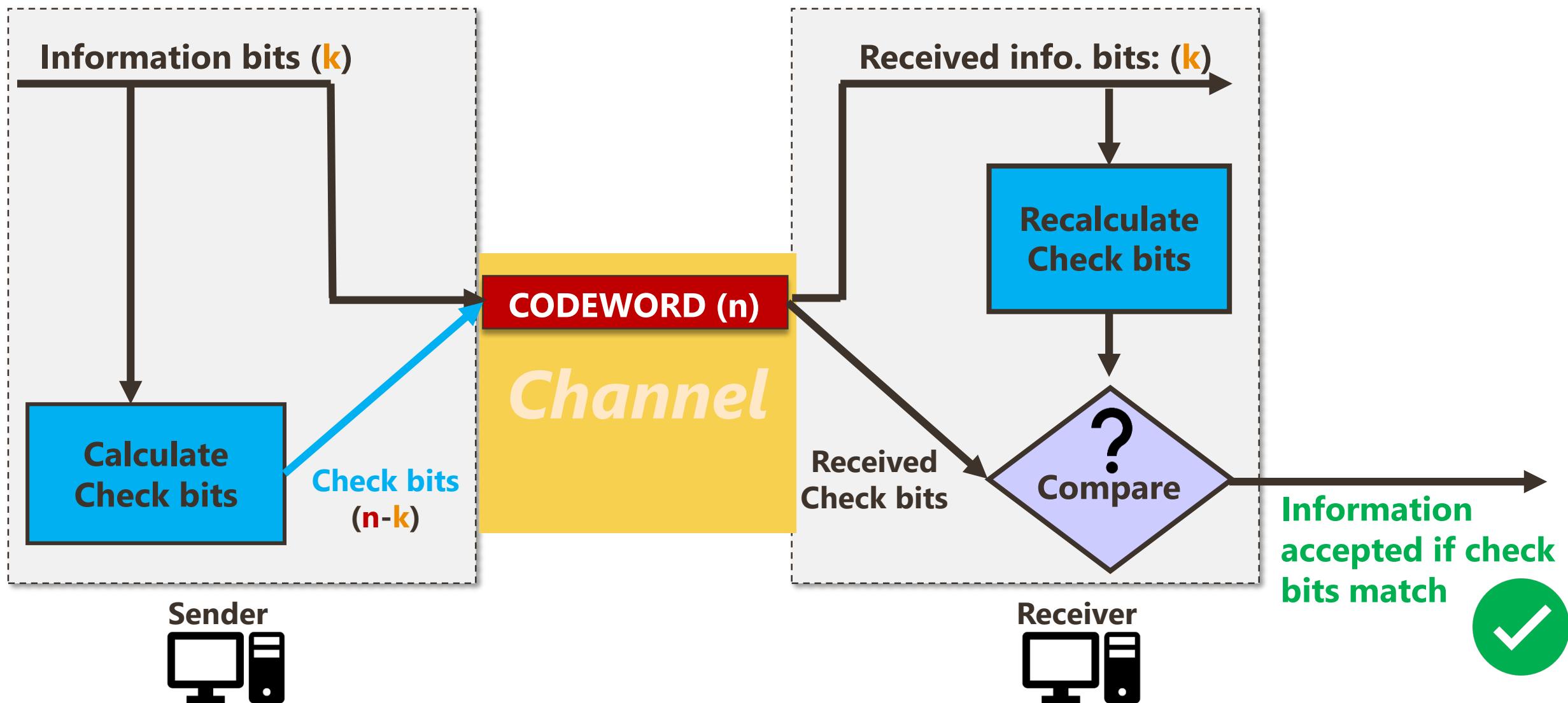


LLC Sub-Layer

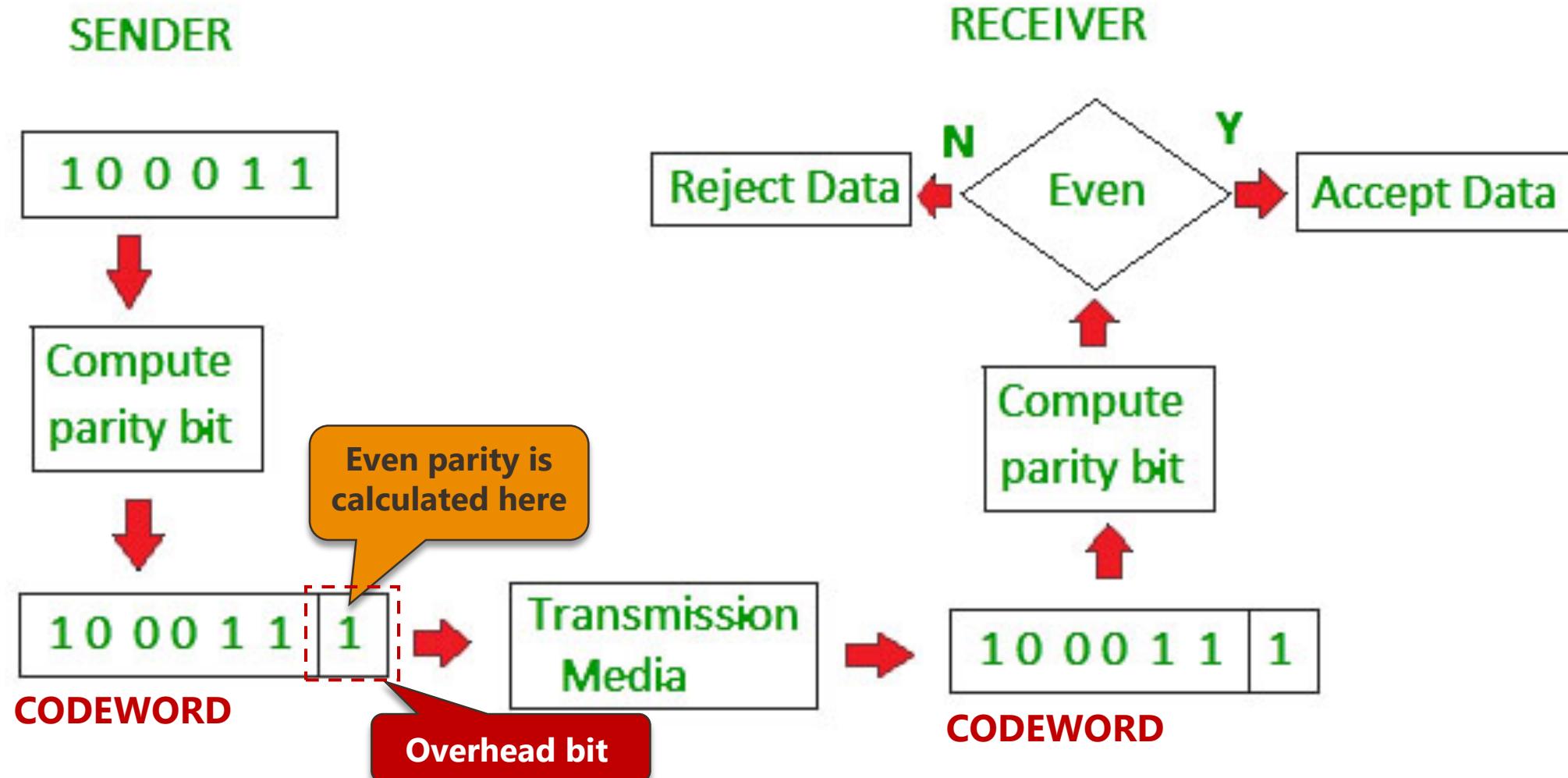
Error Detection and Correction

- Parity Check
- 2D Parity
- Checksum
- CRC

Error Detection Pipeline



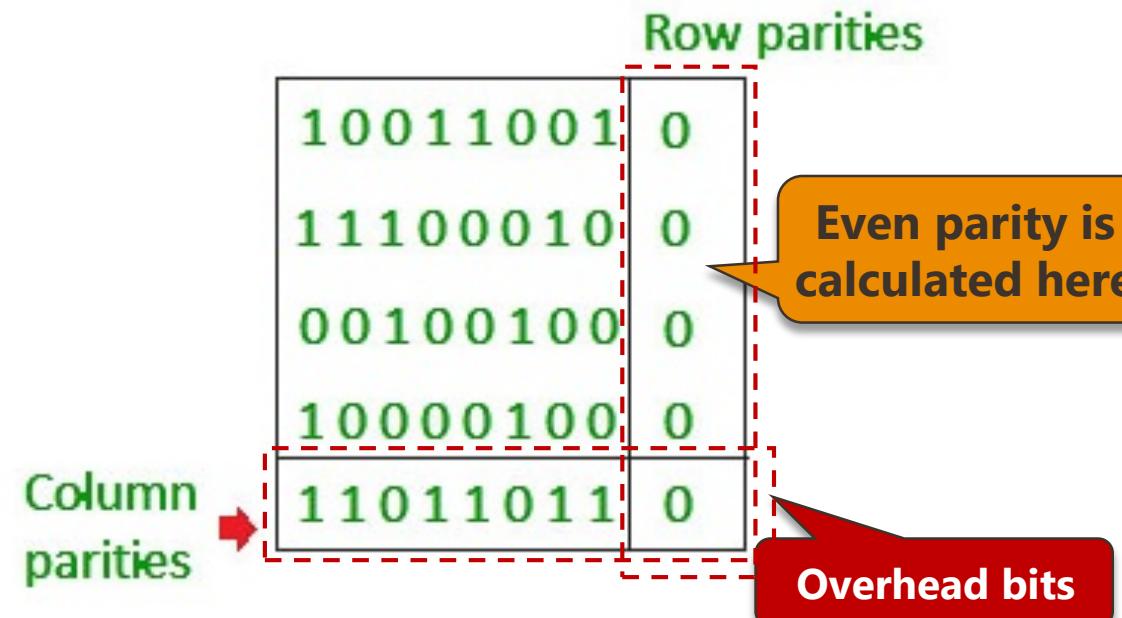
Parity Check (Even Bit/Odd Bit)



2D Parity Check

Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------



CODEWORD =

100110010	111000100	001001000	100001000	110110110
-----------	-----------	-----------	-----------	-----------

Data to be sent

Checksum

Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

1

2

3

4

 $k=4, m=8$

Sender

1	10011001
2	11100010
	<hr/>
	101111011
	<hr/>
	01111100
3	<hr/>
	00100100
	<hr/>
	10100000
4	<hr/>
	10000100
	<hr/>
	100100100
	<hr/>
	00100101

Sum: 00100101

CheckSum: 11011010

Overhead bits

Receiver

1	10011001
2	11100010
	<hr/>
	101111011
	<hr/>
	01111100
3	<hr/>
	00100100
	<hr/>
	10100000
4	<hr/>
	10000100
	<hr/>
	100100100
	<hr/>
	00100101

Sum: 11111111

Complement: 00000000

Conclusion: Accept Data





Checksum

Method 02

• Checksum Calculation Steps (Sender)

1. $x = b_0 + b_1 + b_2 + \dots + b_{L-1}$ modulo $(2^{block_size} - 1)$
2. **checksum** = $-x$ modulo $(2^{block_size} - 1)$

• Checksum Validation Step (Receiver)

3. **0** = $(b_0 + b_1 + b_2 + \dots + b_{L-1} + \text{checksum})$ modulo $(2^{block_size} - 1)$

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

i.e. @ Sender

$$b1 = 10011001 = 153$$

$$b2 = 11100010 = 226$$

$$b3 = 00100100 = 36$$

$$B4 = 10000100 = 132$$

$$x = 153 + 226 + 36 + 132 \text{ modulo } (2^8 - 1) = 37$$

$$\text{checksum} = -x \text{ modulo } (2^8 - 1) = 218 = \textcolor{red}{11011010}$$

i.e. @ Receiver

$$= (153 + 226 + 36 + 132 + 218) \bmod (2^8 - 1)$$

$$= \textcolor{red}{0}$$

Cyclic Redundancy Check (CRC)

- a.k.a **Polynomial Codes**



- **In this technique,**

1. Treat the entire string of data bits as a **single number $D(x)$**
2. Divide this number by a pre-defined polynomial, called the **generator polynomial $G(x)$** , and **append the remainder $R(x)$** to the data string
3. The **receiver performs the same division** and obtain the remainder
4. A **non-zero remainder indicates an error**

CRC – cont.

- **Polynomials** instead of vectors for codewords

$$D(x) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + x^0 \end{bmatrix} = x^6 + x^4$$

- **Polynomial arithmetic** instead of checksums

- Follows laws of ordinary algebra, except **addition** is done in modulo 2:

$$x^a + x^a = 0 \text{ (similar to } 1 \text{ XOR } 1 = 0\text{)}$$

- Mostly used for error detection but a basis for error-correction methods

1 $D(x) = 1010000 = x^6 + x^4$

- info bits ($k = 7$)

2 $G(x) = 1001 = x^3 + 1$

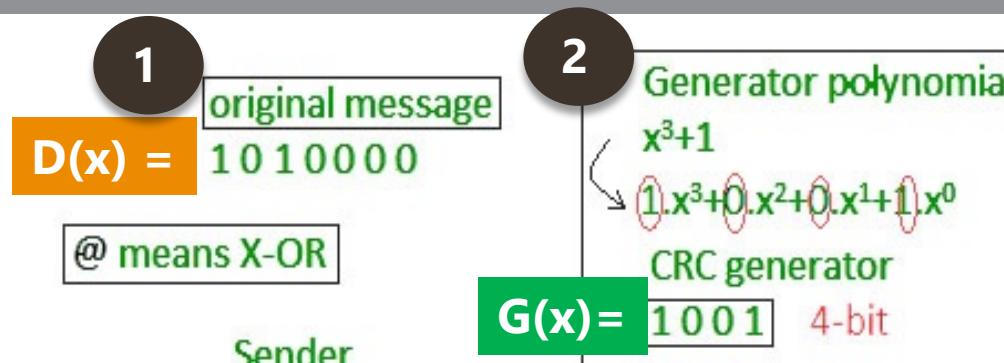
- Codeword length

$$n = \text{degree of } G(x) + k = 3 + 7 = 10$$

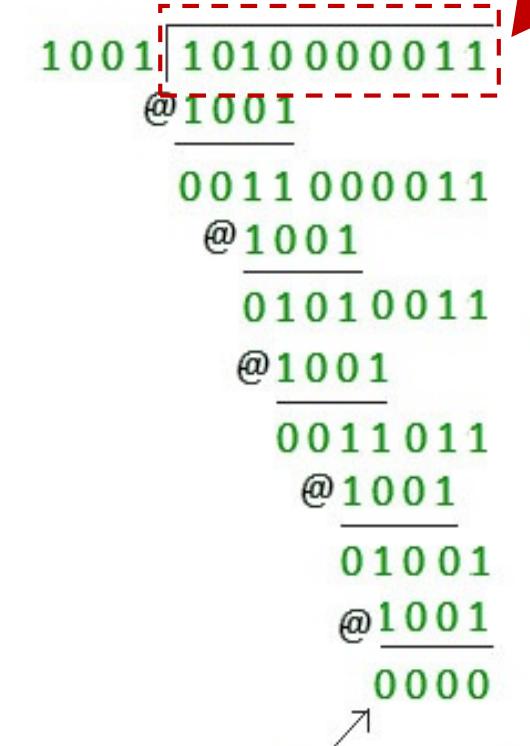
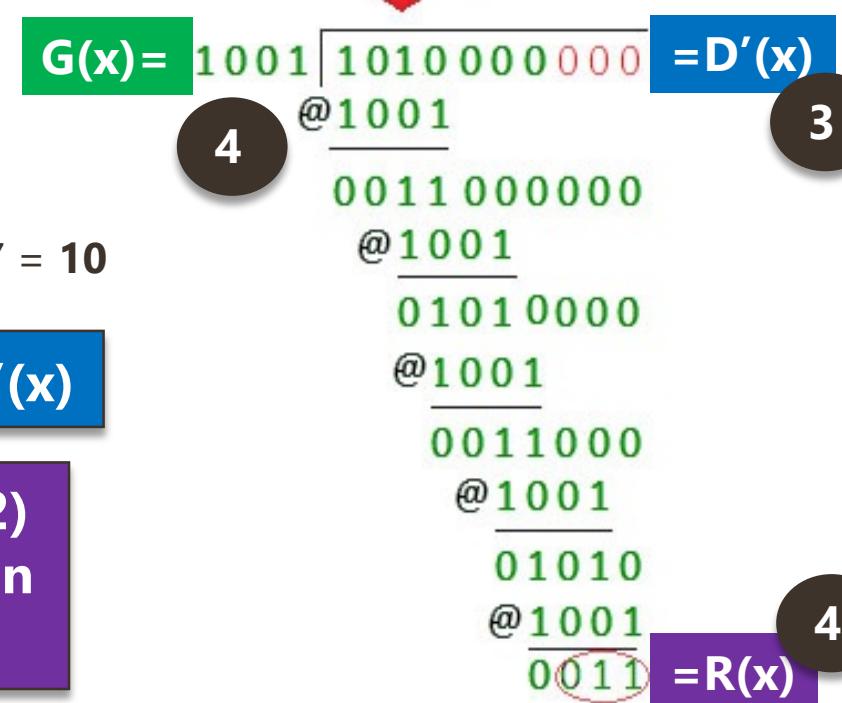
3 Pre-encoding = $x^3 D(x) = D'(x)$

4 Perform polynomial (mod 2) division ($D'(x) / G(x)$), obtain remainder $R(x)$

5 Codeword = $C(x) = D'(x) + R(x)$



If CRC generator is of n bit then append $(n-1)$ zeros in the end of original message



Zero means data is accepted

Choice of Generator Polynormal $G(x)$

It has been proven that a strong $G(x)$ can detect:

- All **single-bit errors**
- Almost all **double-bit errors**, if $G(x)$ has a factor with at least three terms
- Any **odd number of errors**, if $G(x)$ has the factor $x + 1$
- All **bursts of** up to m **errors**, if $G(x)$ is of degree m
- **Longer burst errors** with probability $1 - 2^{-m}$, if bursts are randomly distributed

Standard Generator Polynomials

- **CRC-8:**

$$x^8 + x^2 + x + 1$$

- ✓ ATM

- **CRC-16:**

$$x^{16} + x^{15} + x^2 + 1$$

- ✓ HDLC, XMODEM, V.41

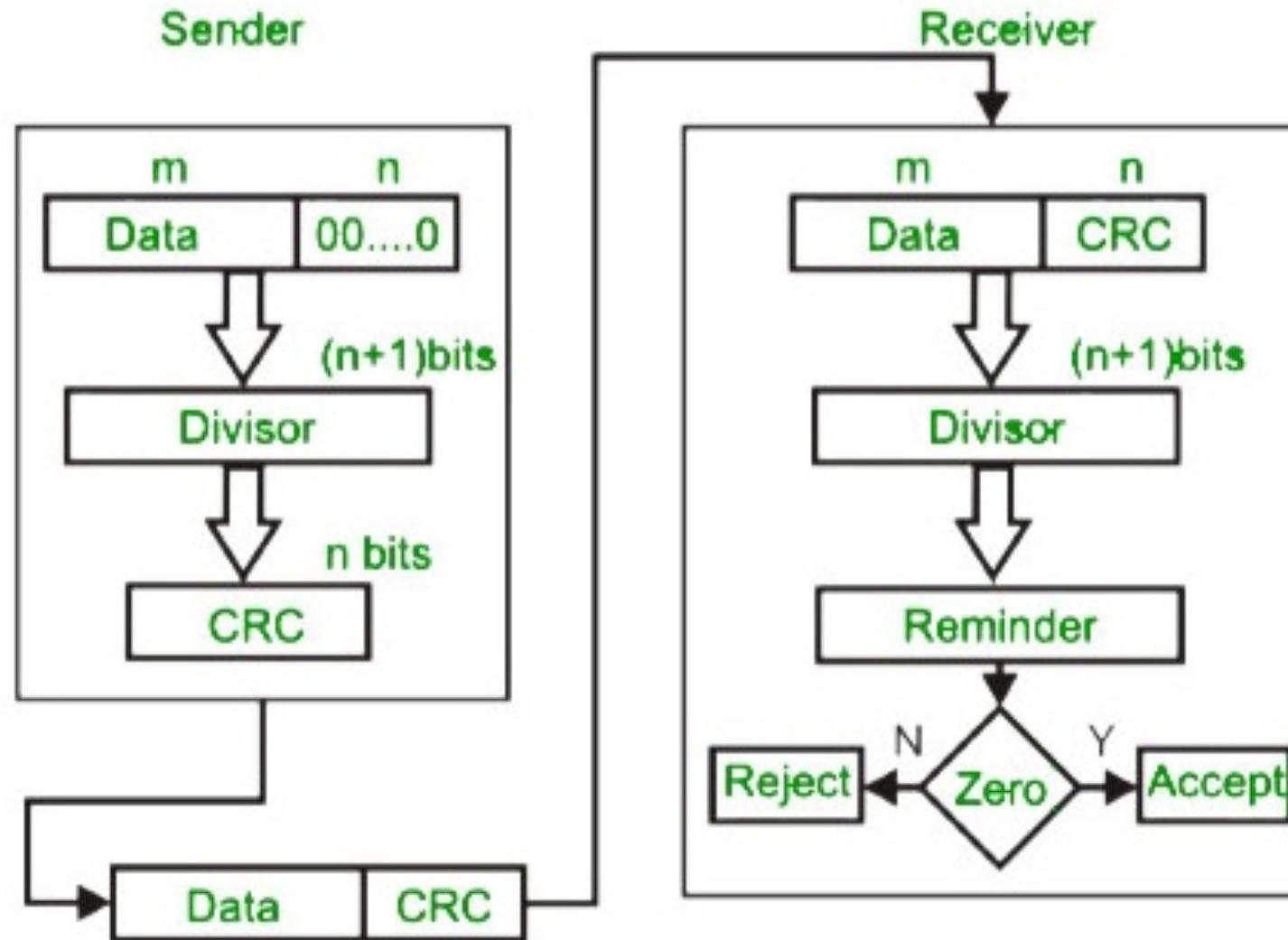
- **CRC-32:**

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

- ✓ IEEE 802, TCP/IP Model, **V.42**

V. 42 permits computer modems to work with both digital and analog phone lines

CRC Summary





MAC Sub-Layer

Framing

- Application protocols: **HDLC, PPP**
- **HDLC** – Framing with Bit Stuffing
- **PPP** – Framing with Byte Stuffing

Applications of HDLC & PPP

▪ HDLC (High-level Data Link Control)

- Bit-oriented
- LAPD (Link Access Protocol D-Channel) in ISDN
- LAPD (Link Access Procedure for Modems) in cellular telephone signaling

▪ PPP (Point to Point Protocol)

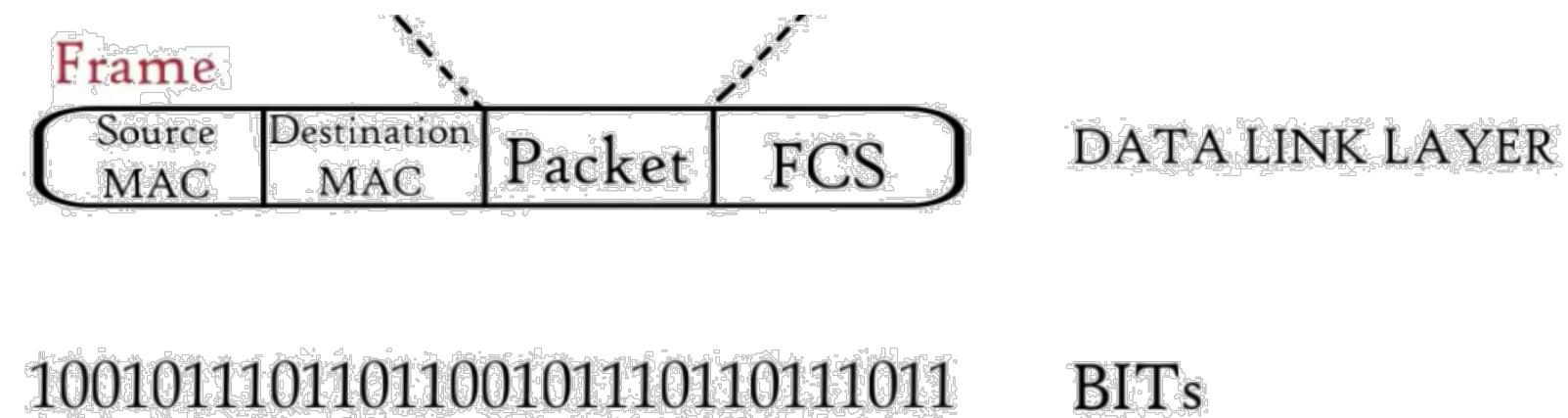
- Telephone Modem Links (30 Kbps)
- Packet over SONET (600-10000Mbps) - Synchronous Optical Network
 - IP->PPP->SONETAPD (Link Access Protocol D-Channel) in ISDN
- PPP over shared links
 - PPP over Ethernet (RFC 2516)
 - Used over DSL

used to transmit a large amount of data over relatively large distances using **optical fibre**

Framing / De-framing

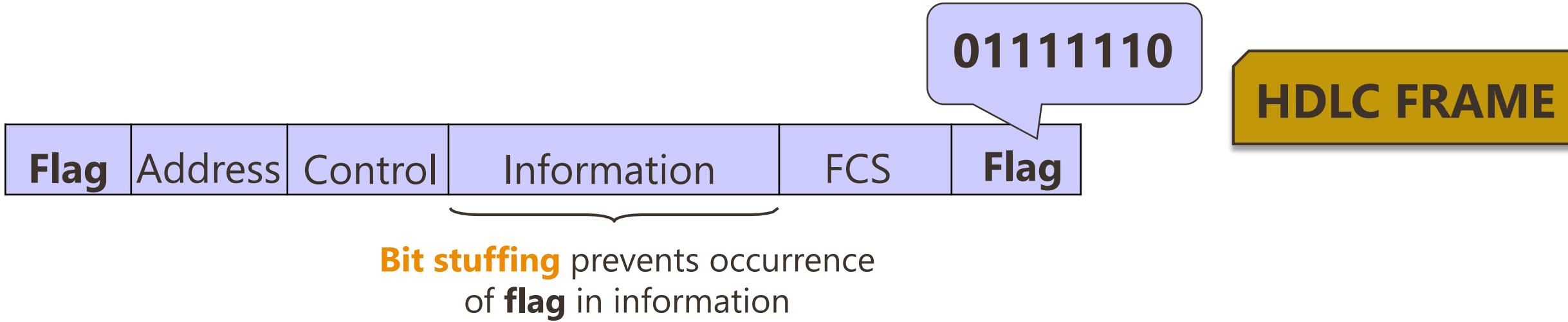
- Map stream of physical layer **bits** into **frames** (vice-versa)
- **Frame boundaries can be determined using:**

- ✓ Character Counts
- ✓ Control Characters
- ✓ **Flags**
- ✓ CRC Checks



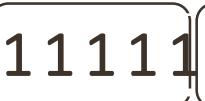
Framing & Bit Stuffing

- **HDLC** uses bit-stuffing



METHOD

- **Transmitter** inserts extra 0 after each consecutive five 1s inside the frame
- **Receiver** checks for five consecutive 1s
 1. if next bit = 0, it is removed
 2. if next two bits are 10, then flag is detected
 3. If next two bits are 11, then frame has errors

Data to be sent: 0110  

After stuffing and framing

 011011110  

**HDLC Frame
Bit-Stuffing**

Data received:  0001110  

After destuffing and framing

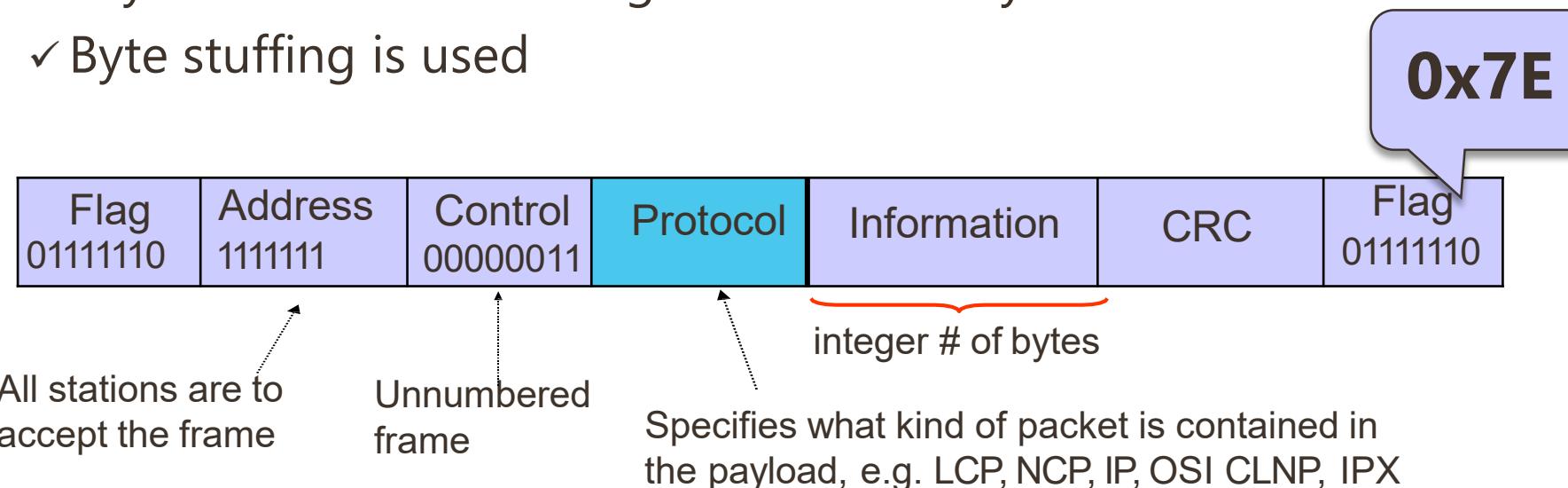
*000111011111  -111111  *

**HDLC Frame
Bit-destuffing**

Framing & Byte Stuffing

- **PPP** (Point-to-Point Protocol) is **character-oriented**
- **PPP** Frame $\sim =$ **HDLC** Frame, **except**

- ✓ Protocol type field
- ✓ Payload contains an integer number of bytes
- ✓ Byte stuffing is used



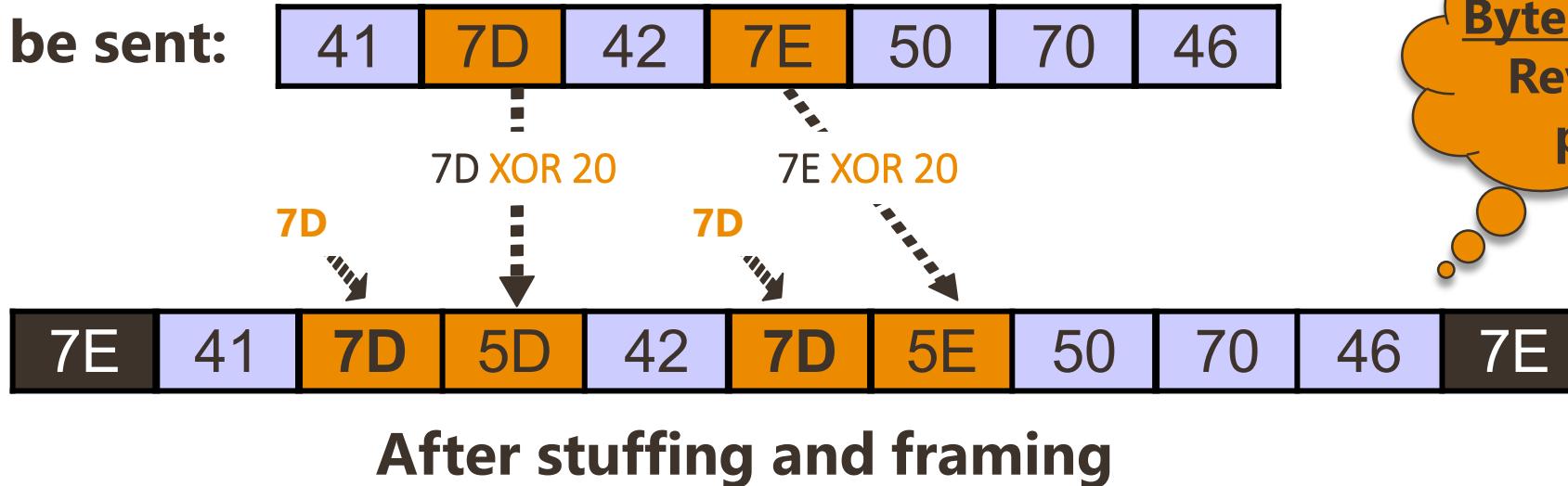
PPP FRAME

Use **control escape (7D)** in front of any 7E in the payload to indicate it's not a flag.

Byte Stuffing

Any occurrence of **flag(7E)** or **control escape (7D)** inside the frame is replaced with 0x7D(01111101) followed by them XORed with 0x20 (00100000)

Data to be sent:



▪ Problems with PPP Byte Stuffing !

- Size of frame varies unpredictably due to byte insertion
- Malicious users can inflate bandwidth by inserting 7D & 7E

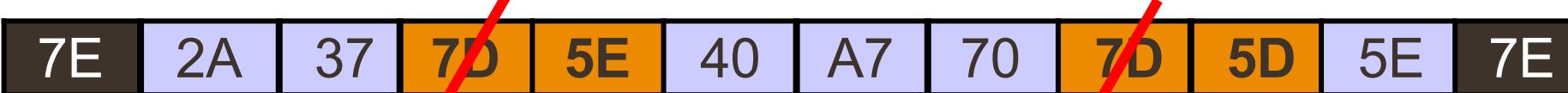
Data to be sent:**Byte Stuffing**

7E XOR 20

7D XOR 20

7D

7D

**After stuffing and framing****Data received:****After destuffing and framing**

5E XOR 20

5D XOR 20



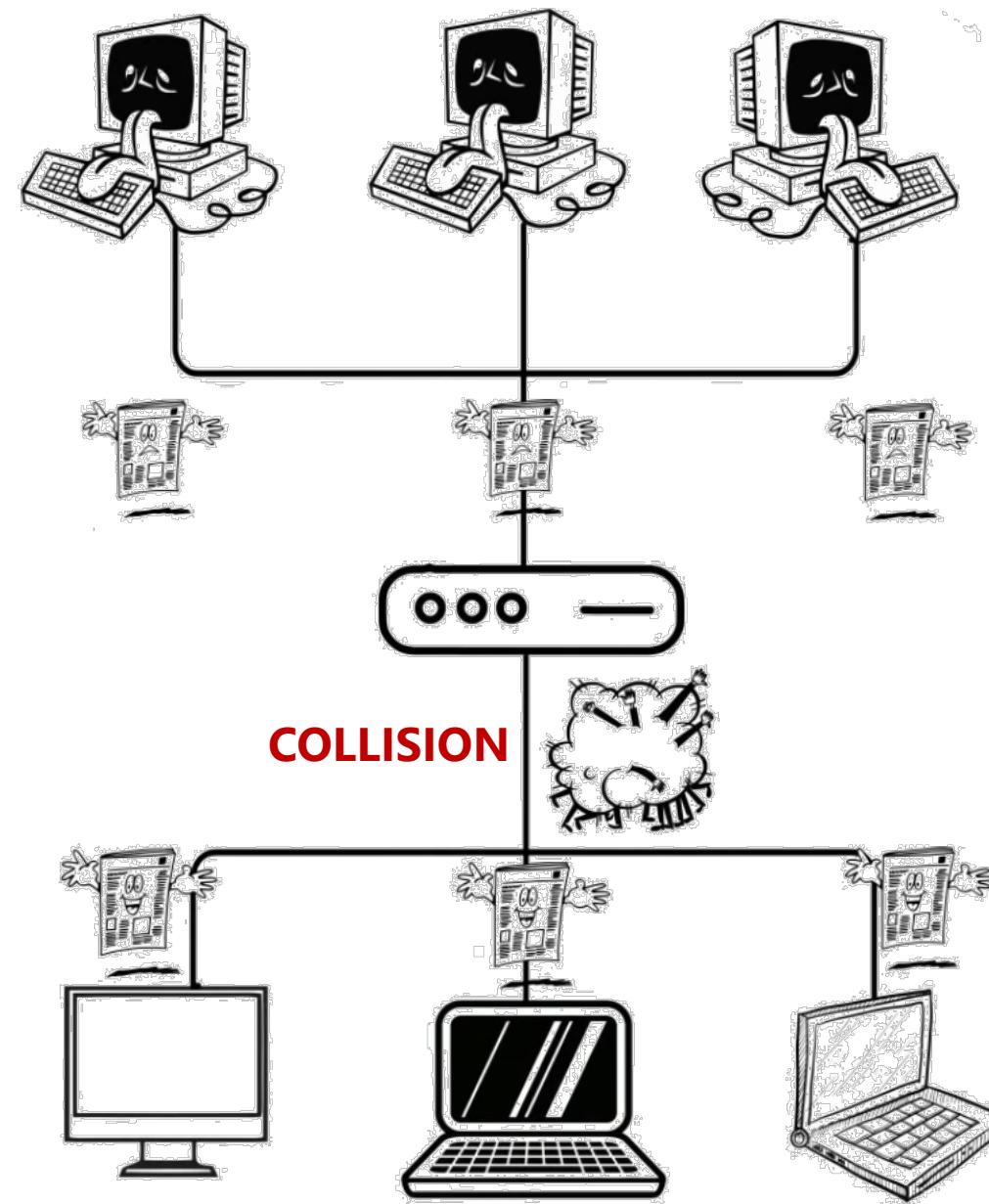


MAC Sub-Layer

Media Access Control/Sharing

- Scheduling Methods
 - Polling
 - Token-passing
- Random Access Methods
 - ALOHA
 - CSMA, CSMA Options
 - CSMA/CD
- Media Sharing Example: Wireless LAN

What happens when everybody start sending data ?



Medium Access Control

Static Channelization

- Partition Medium
- Dedicated Allocation to users
- Satellite Transmission
- Cellular Telephone

Dynamic Medium Access Control

- ### Scheduling
- Polling: Take turns
 - Request for slot in transmission schedule
 - Token Ring
 - Wireless LANs

- ### Random Access
- Loose coordination
 - Send, wait, retry if necessary
 - Aloha
 - Ethernet

Selecting a Medium Access Control Method

▪ Applications

- What type of traffic?
- Voice streams? Steady traffic, low delay/jitter
- Data? Short messages? Web page downloads?
- Enterprise or Consumer market? Reliability, cost

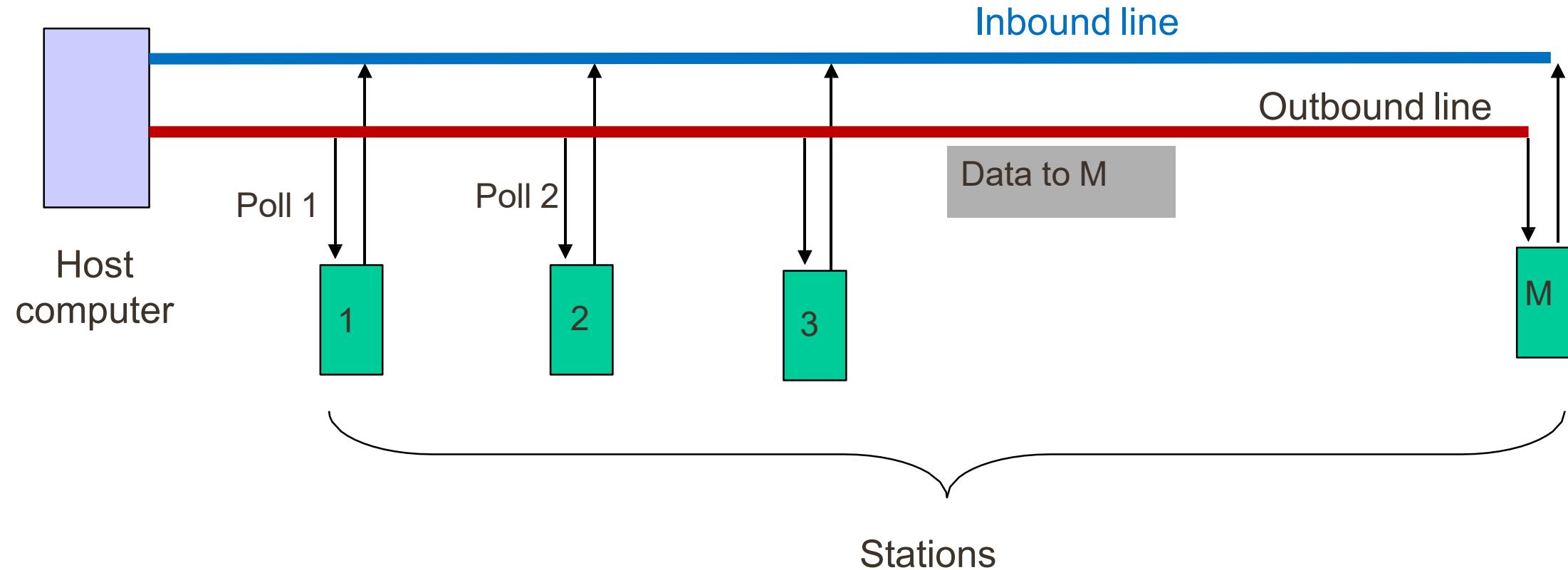
▪ Scale

- How much traffic can be carried?
- How many users can be supported?

▪ Current Examples:

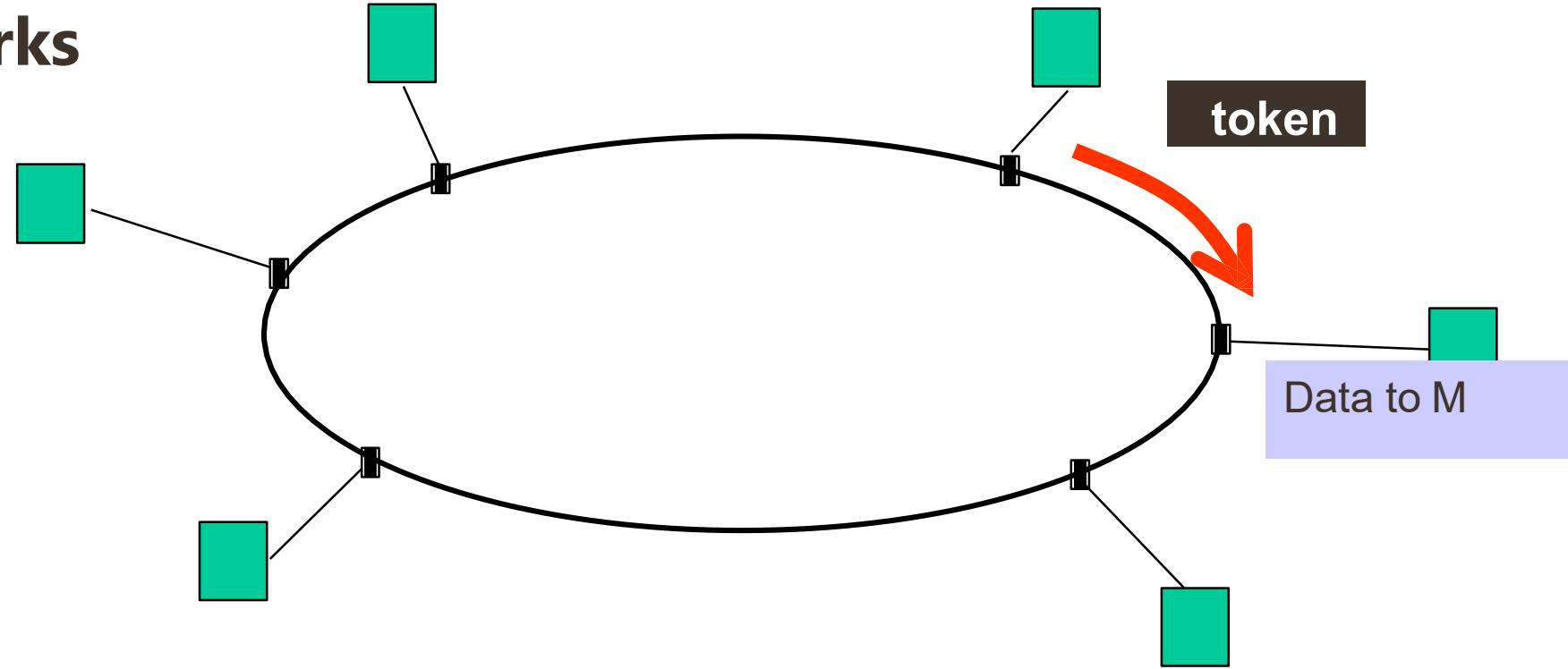
- Design MAC to provide wireless DSL-equivalent access to rural communities
- Design MAC to provide Wireless-LAN-equivalent access to mobile users
(user in car travelling at 130 km/h)

Scheduling: Polling



Scheduling: Token-Passing

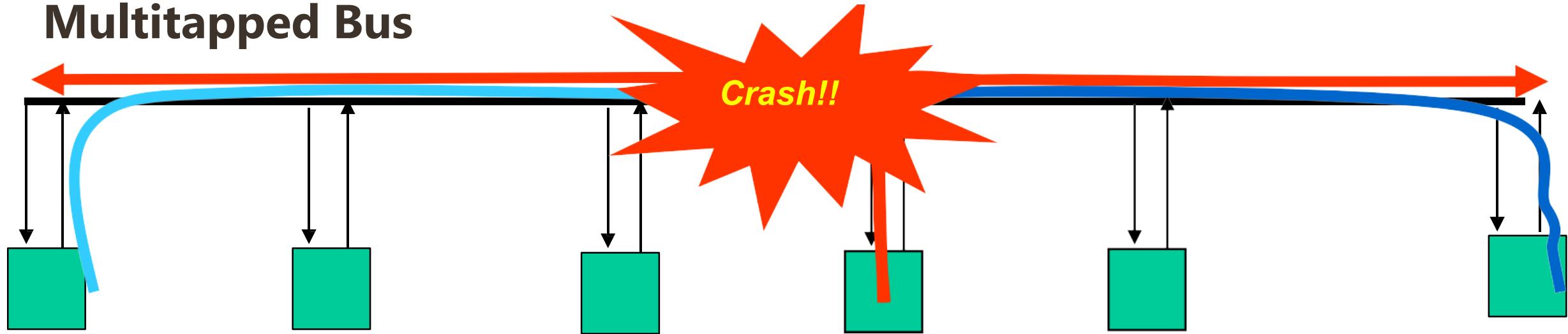
Ring Networks



Station that holds token transmits into ring

Random Access

Multitapped Bus



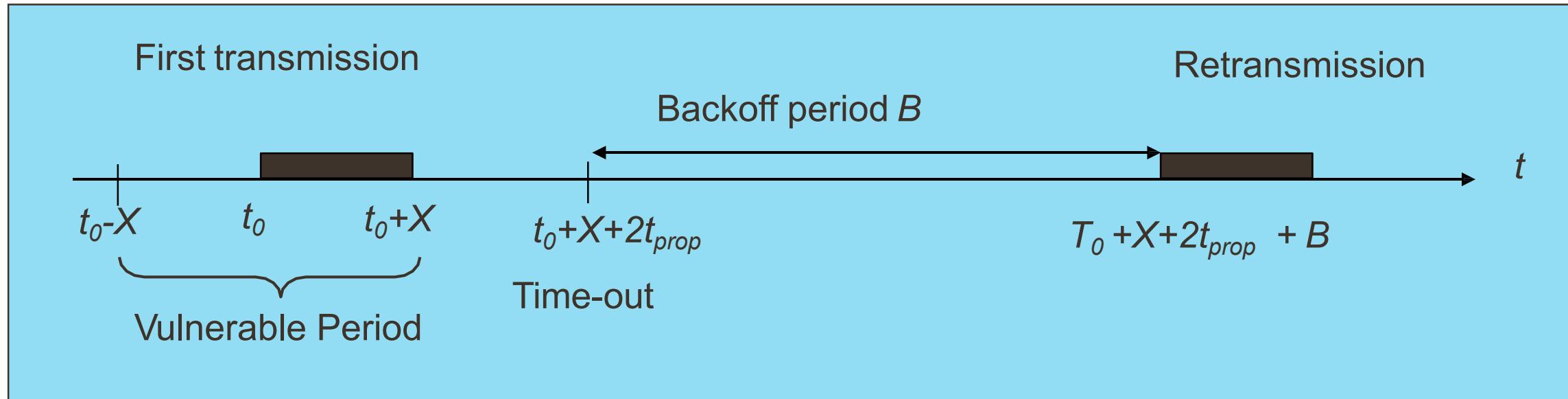
Transmit when ready

Collisions can occur;

Need a retransmission strategy

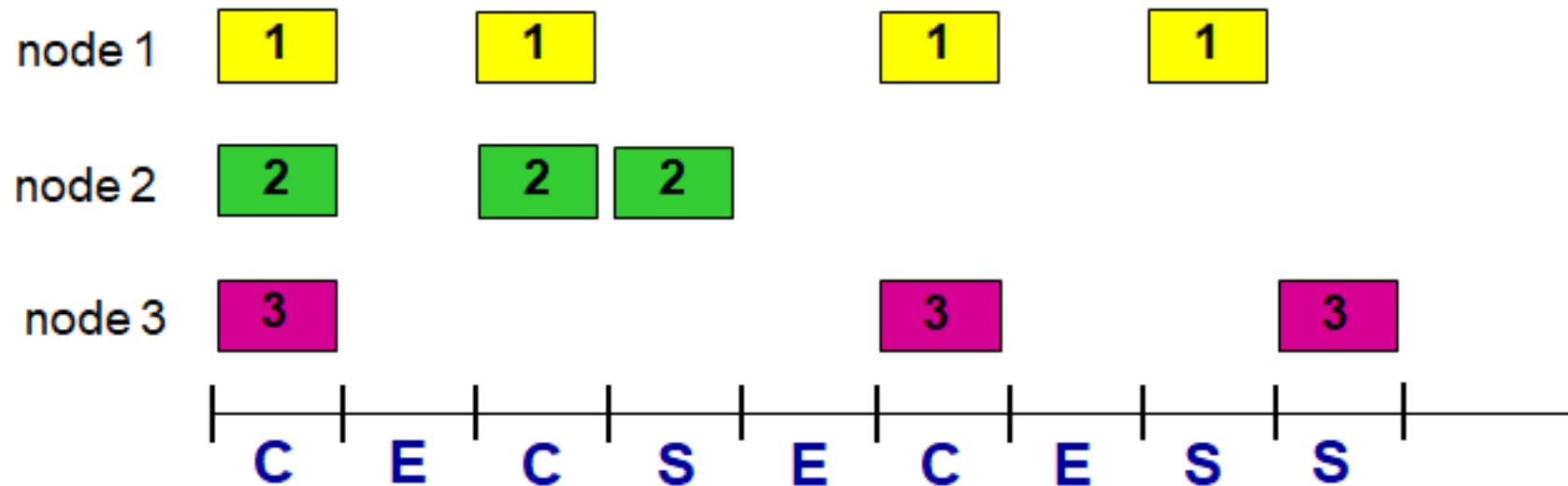
Random Access: ALOHA

- ✓ A station transmits whenever it has data to transmit
- ✓ If more than one stations are transmitting (**frame collision**) !
- ✓ If ACK not received before timeout, a station picks random backoff time (**to avoid repeated collision**)



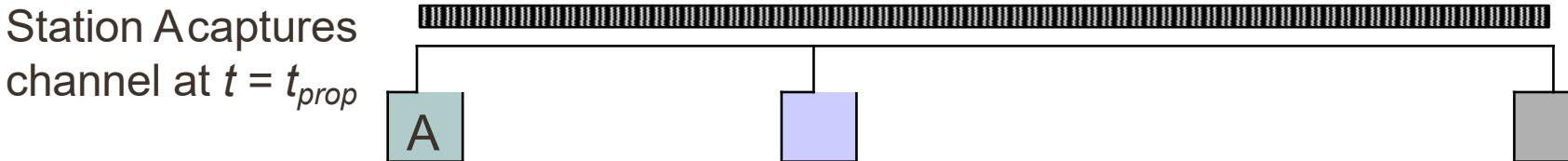
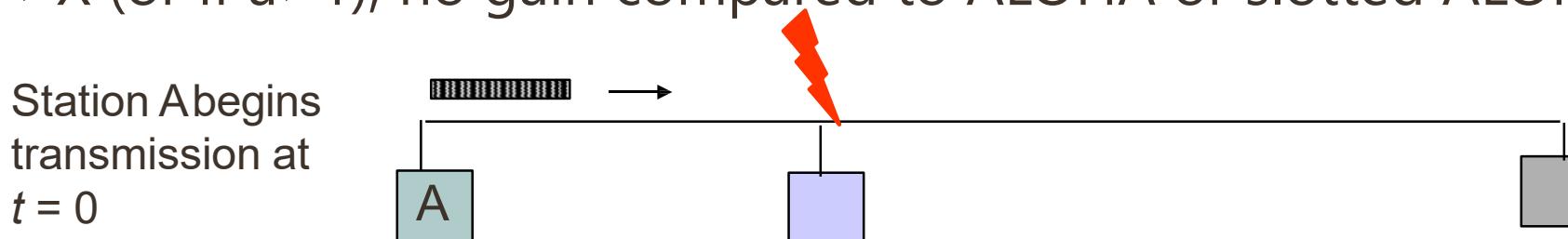
Random Access: Slotted ALOHA

- All frames are same size
- Time is divided into equal size slots (time to transmit 1 frame)
- Stations (nodes) are synchronized
- Nodes start to transmit at the beginning of the next slot after data is ready
- if 2 or more nodes transmit in a slot, all nodes detect collision
- *if collision:* node retransmits frame in each subsequent slot with probability p until success



Random Access: CSMA

- ✓ Carrier Sense Multiple Access
- ✓ A station senses the channel before it starts transmission
- ✓ If idle, start transmission
- ✓ If busy, either wait or schedule backoff (**CSMA Options**)
- ✓ When collisions occur, they involve entire frame transmission times
- ✓ If $t_{prop} > X$ (or if $a > 1$), no gain compared to ALOHA or slotted ALOHA



If a channel sensed busy,

1. **1-persistent CSMA** (most greedy)

- ✓ Start transmission as soon as the channel becomes idle
- ✓ Low delay & low efficiency

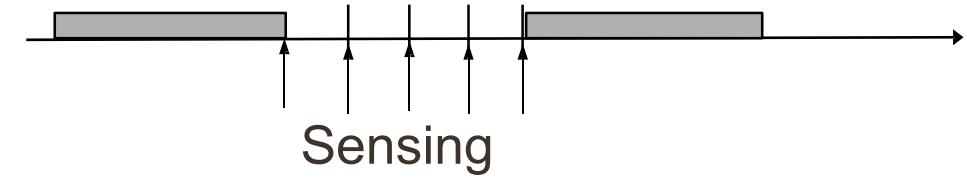
2. **Non-persistent CSMA** (least greedy)

- ✓ Wait a backoff period, then sense carrier again
- ✓ High delay & high efficiency

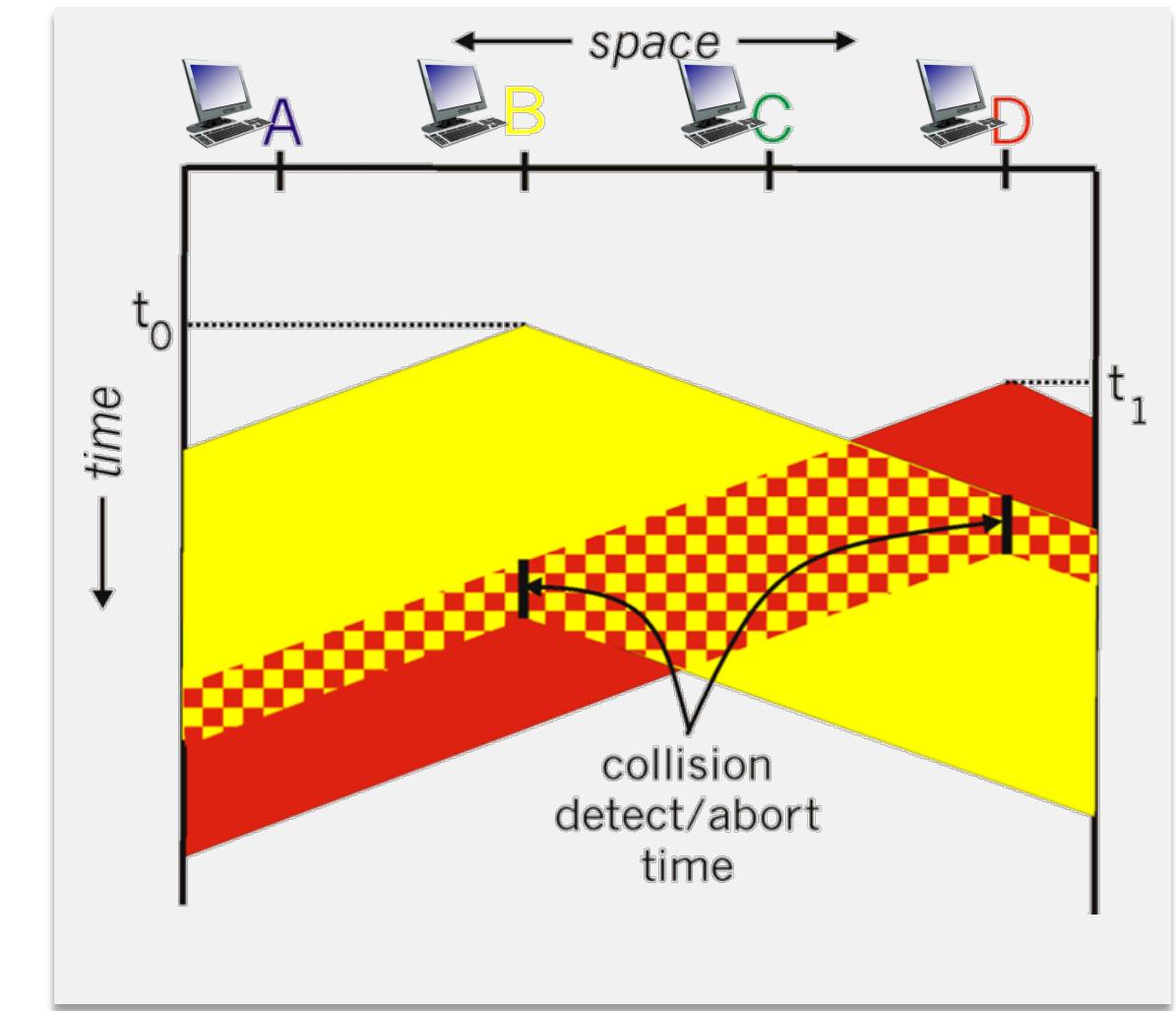
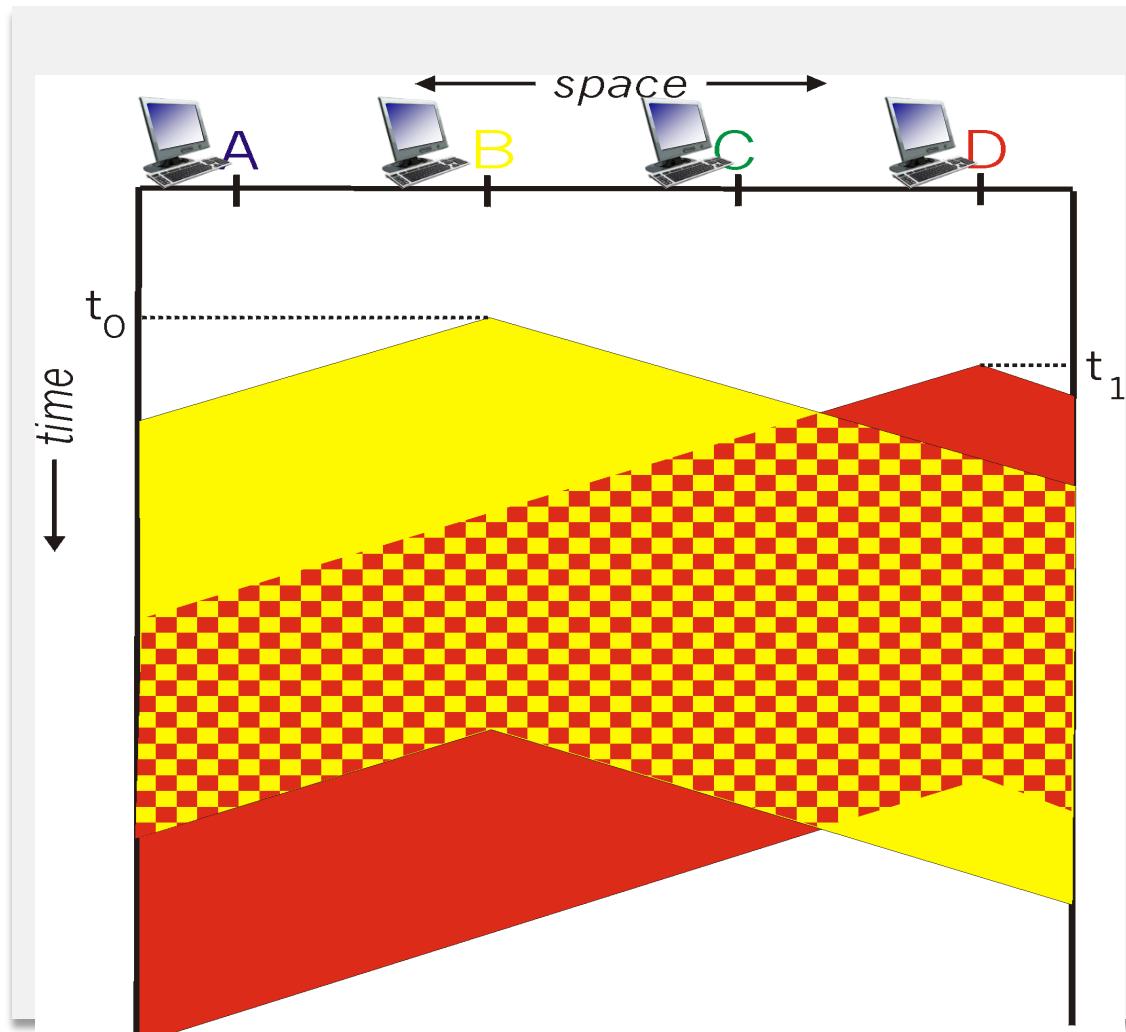
3. **p-persistent CSMA** (adjustable greedy)

- ✓ Wait till channel becomes idle, transmit with probability p ;
- ✓ Or wait one mini-slot time & re-sense with probability $1-p$
- ✓ Delay & efficiency can be balanced

**CSMA
OPTIONS**



Random Access: CSMA-CD



Random Access: CSMA-CD

- Collision Detection

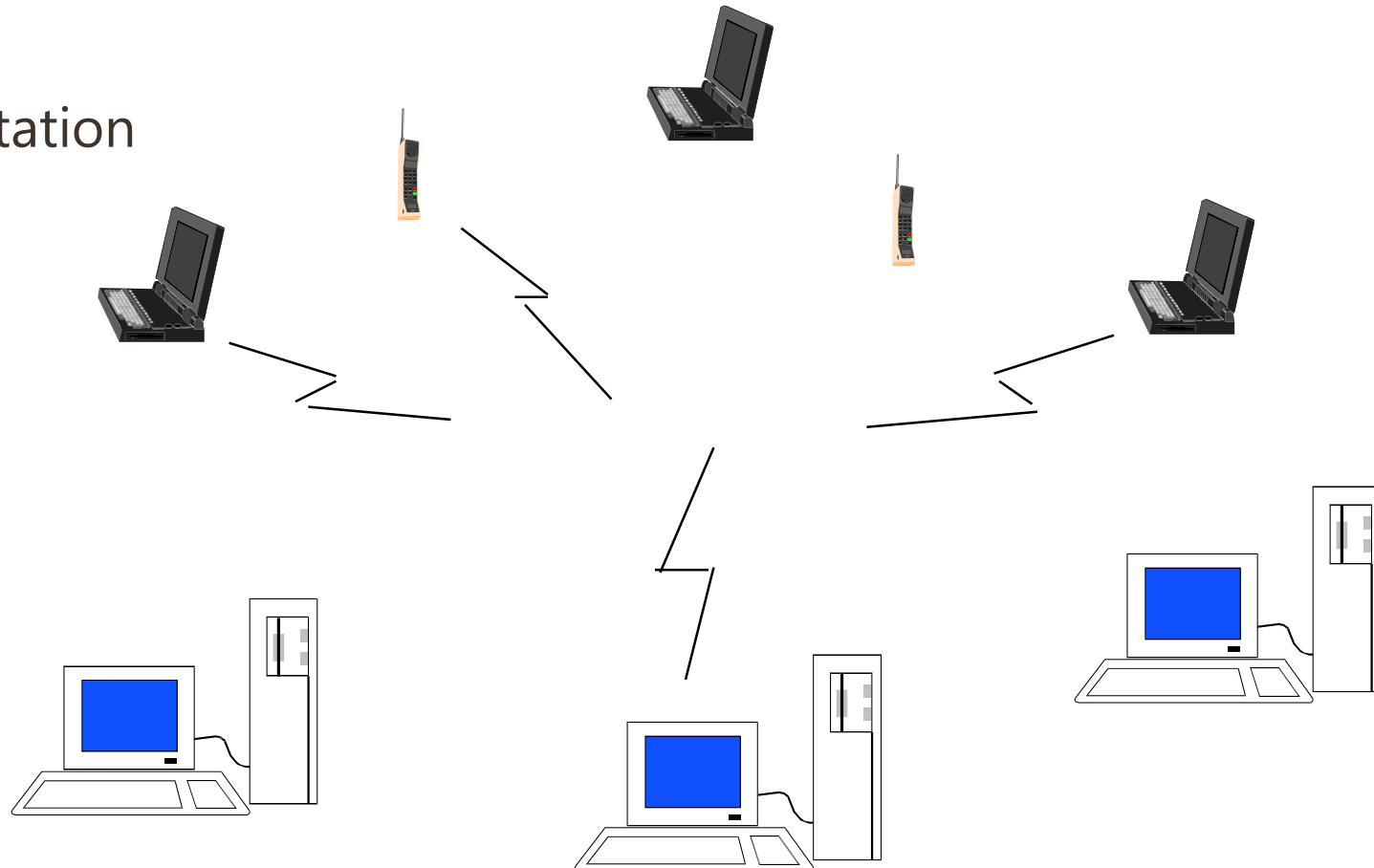
- **Easy in wired LANs:** measure signal strengths, compare transmitted, received signals
- **Difficult in wireless LANs:** received signal strength overwhelmed by local transmission strength

Choice of MAC methods

▪ E.g. In Wireless LAN

- **Ad Hoc:** station-to-station
- **Infrastructure:** stations to base station
- **Access Method:**

Random Access & Polling





■ Data-link Layer

- Fundamentals
- Protocols, Sub-layers
- Sub Layers (LLC, MAC)
- Sub Layers: Services

■ LLC: Flow Control

- Stop-and-Wait Protocol
- Sliding Window Protocols
 - Go-Back-N ARQ
 - Selective-Reject ARQ

■ LLC: Error Detection

- Parity-check (1d, 2d)
- Checksum
- CRC

■ MAC: Framing

- Bit-stuffing / HDLC
- Byte-stuffing / PPP
- HDLC, PPP Applications

■ MAC: Media Access Control/Sharing

- Scheduling Methods
 - Polling
 - Token-passing
- Random Access Methods
 - ALOHA
 - CSMA
 - CSMA/CD

THANK YOU

Make tomorrow better.

Data Link Layer II

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 04

Semester 2, 2020



Ethernet

- LAN, Ethernet Fundamentals
- History and Evolution
- Ethernet, Fast Ethernet, Gigabit Ethernet and Cabling
- IEEE802.1 Ethernet DLL
 - LLC Services
 - MAC data frame
 - MAC protocol
 - Deployment
- Adaptive Learning

What is LAN?

- **Private ownership**

- ✓ freedom from regulatory constraints of WANs

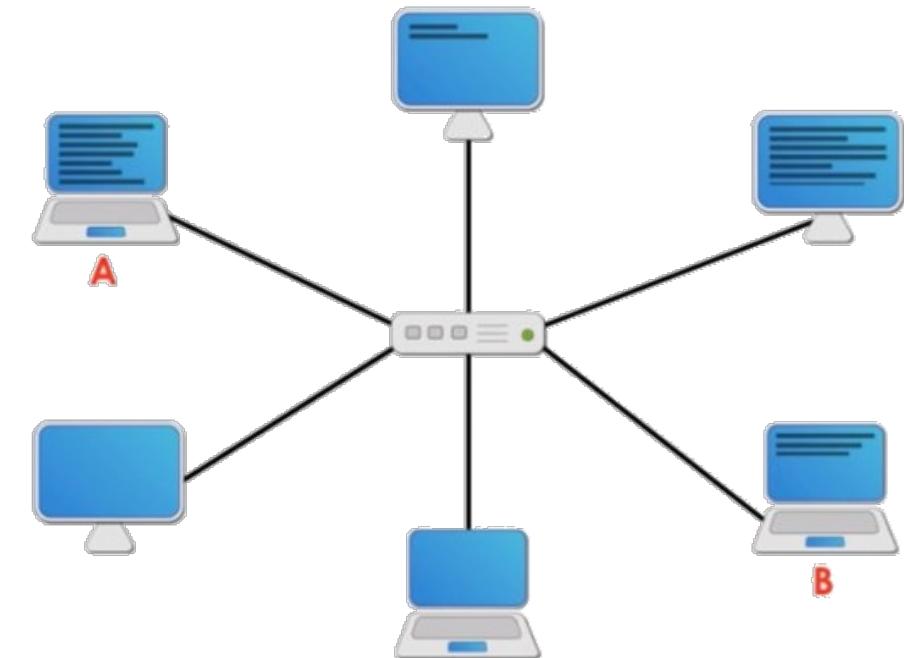
- **Short distance** (~1km)

- ✓ low cost
 - ✓ very high-speed, relatively error-free communication
 - ✓ complex error control unnecessary

- **LAN** characterizes:

- ✓ Topology (Star, Bus, Mesh, etc.)
 - ✓ Protocols (CSMA/CD, CSMA/CA, etc.)
 - ✓ Media (twisted pair, coaxial, fiber optic)

Ethernet network - Local Area Network (LAN)



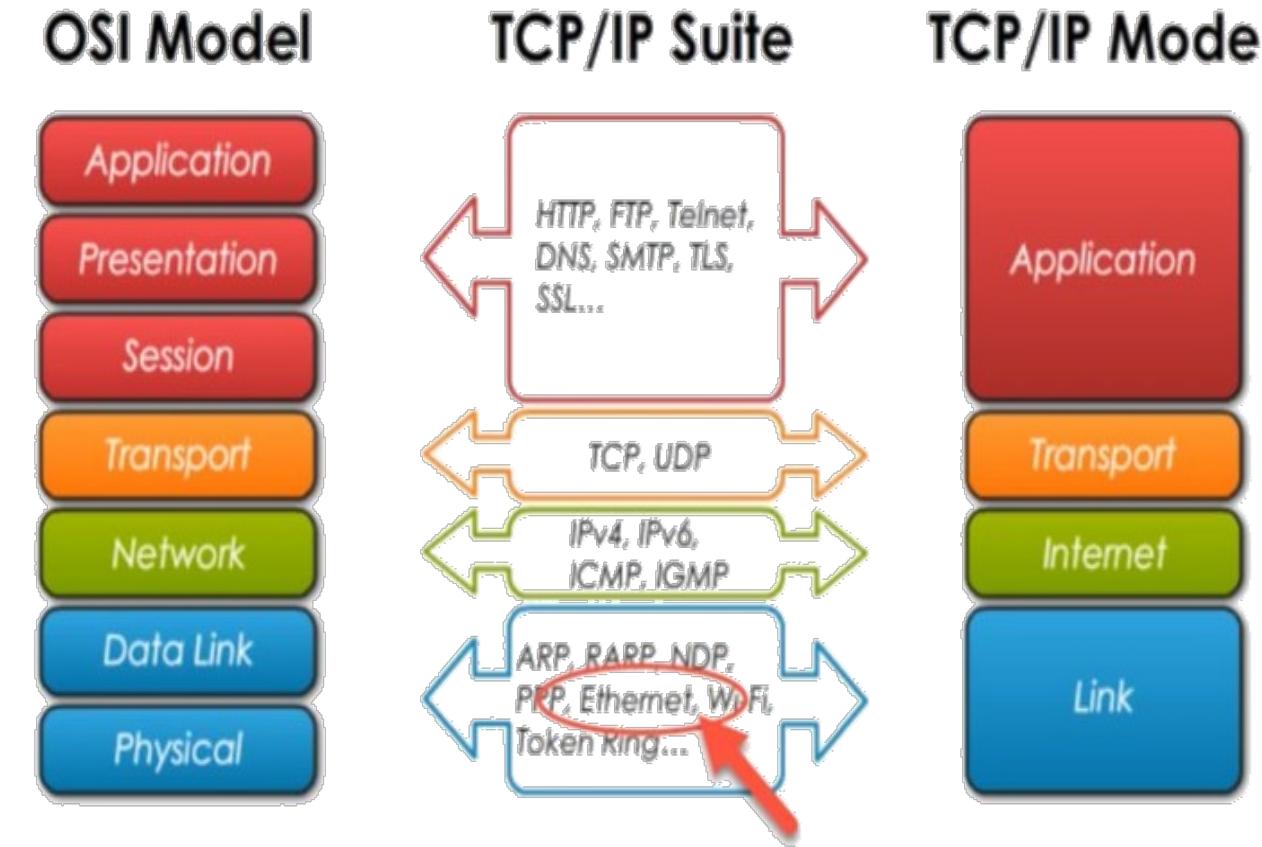
Ethernet

- **is a LAN Technology** (*most popular*)

- Other LAN technologies

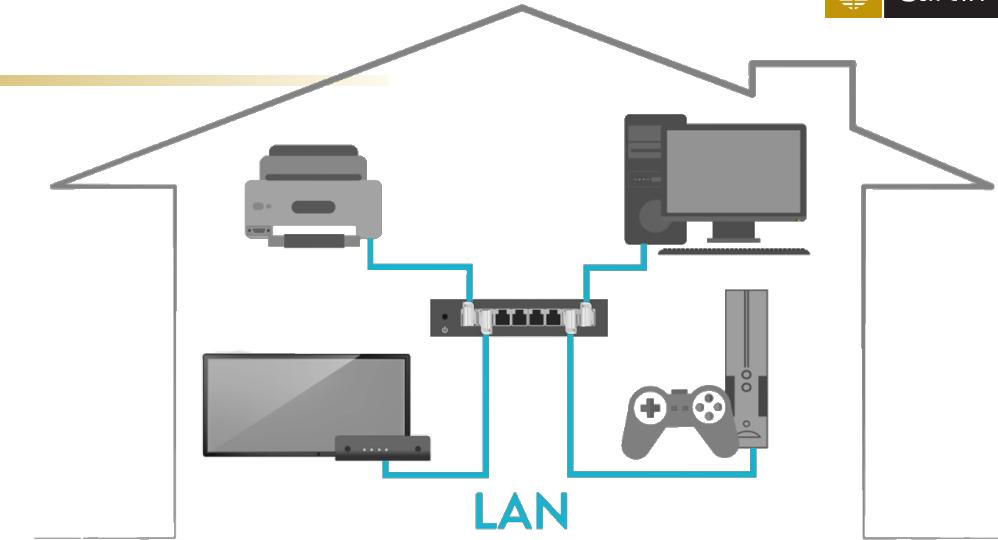
- ✓ Token Ring
- ✓ FDDI (Fiber Distributed Data Interface)
- ✓ ARCNET

- Operates at both **physical** and **data link layer**



Ethernet

- Is a **baseband** system

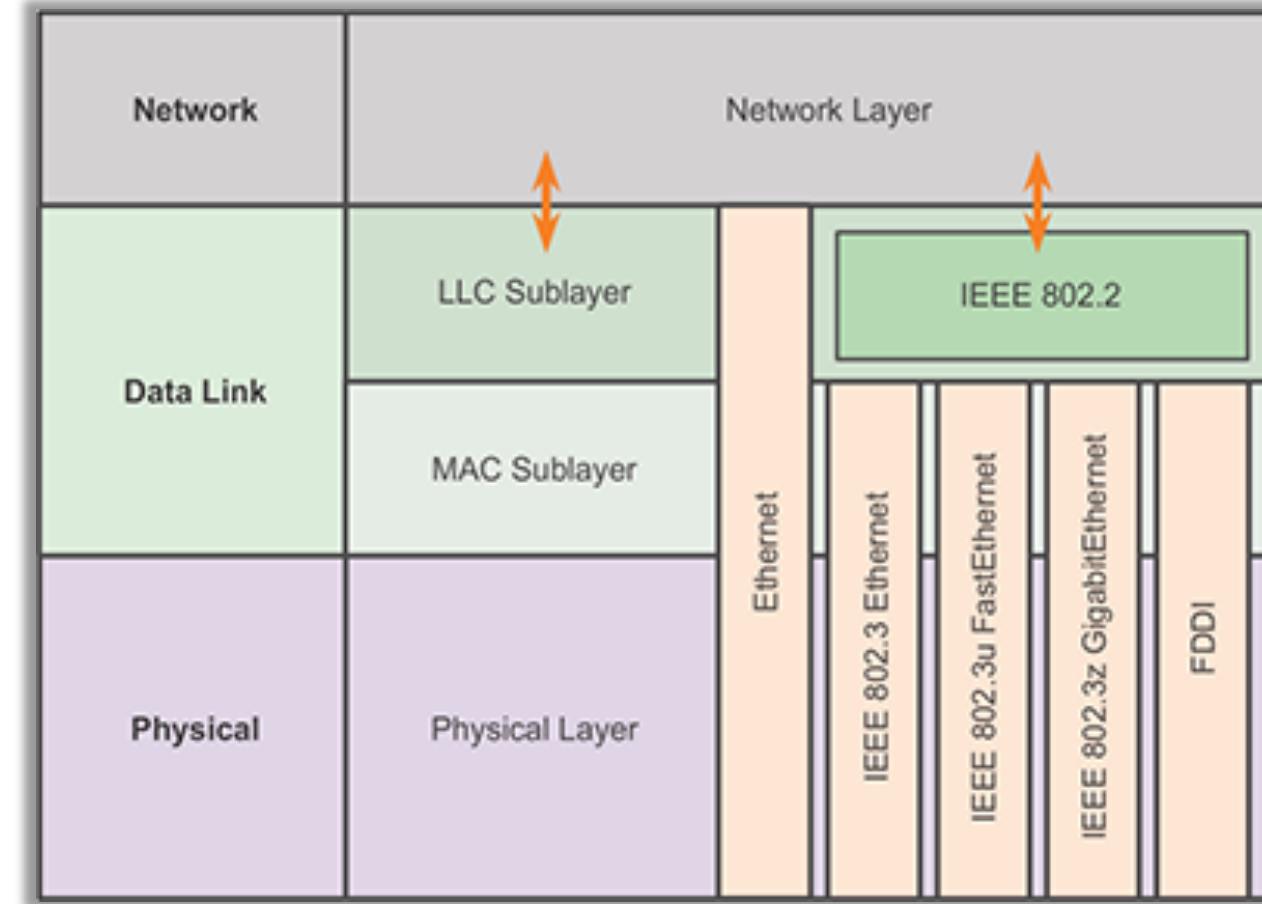


- Ethernet implementations of the network may differ
- But,
 - **Basic Topology**
 - **Frame Type**
 - **Network Access Method**

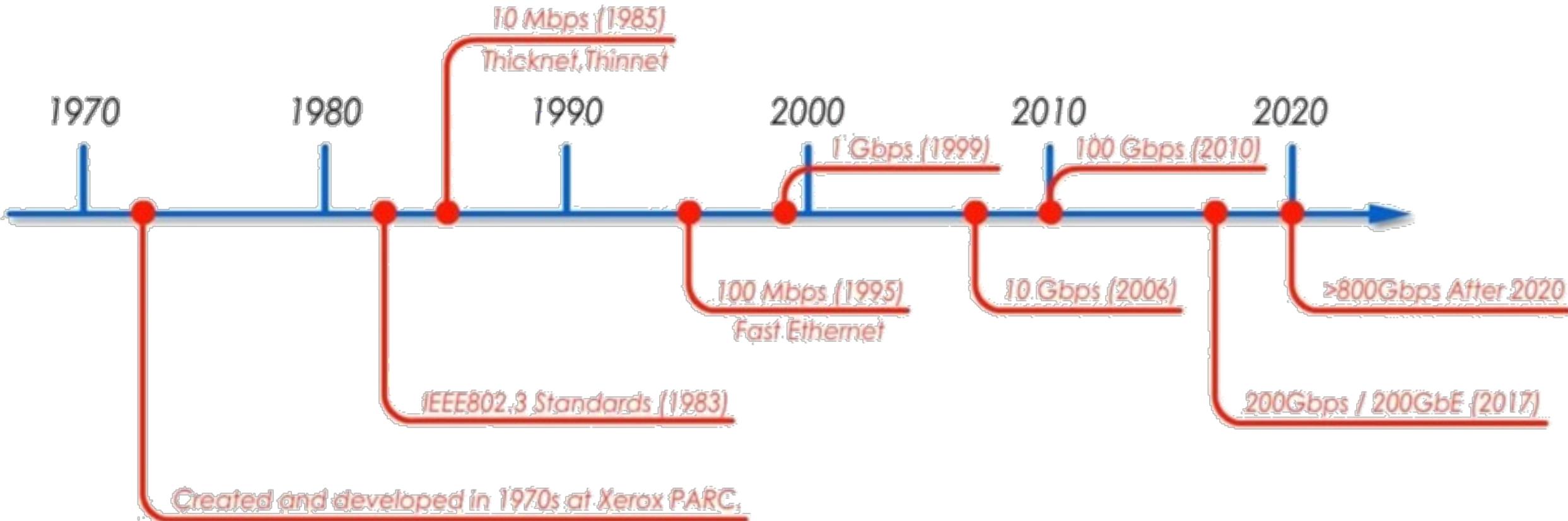
} Same

Project 802 of the IEEE

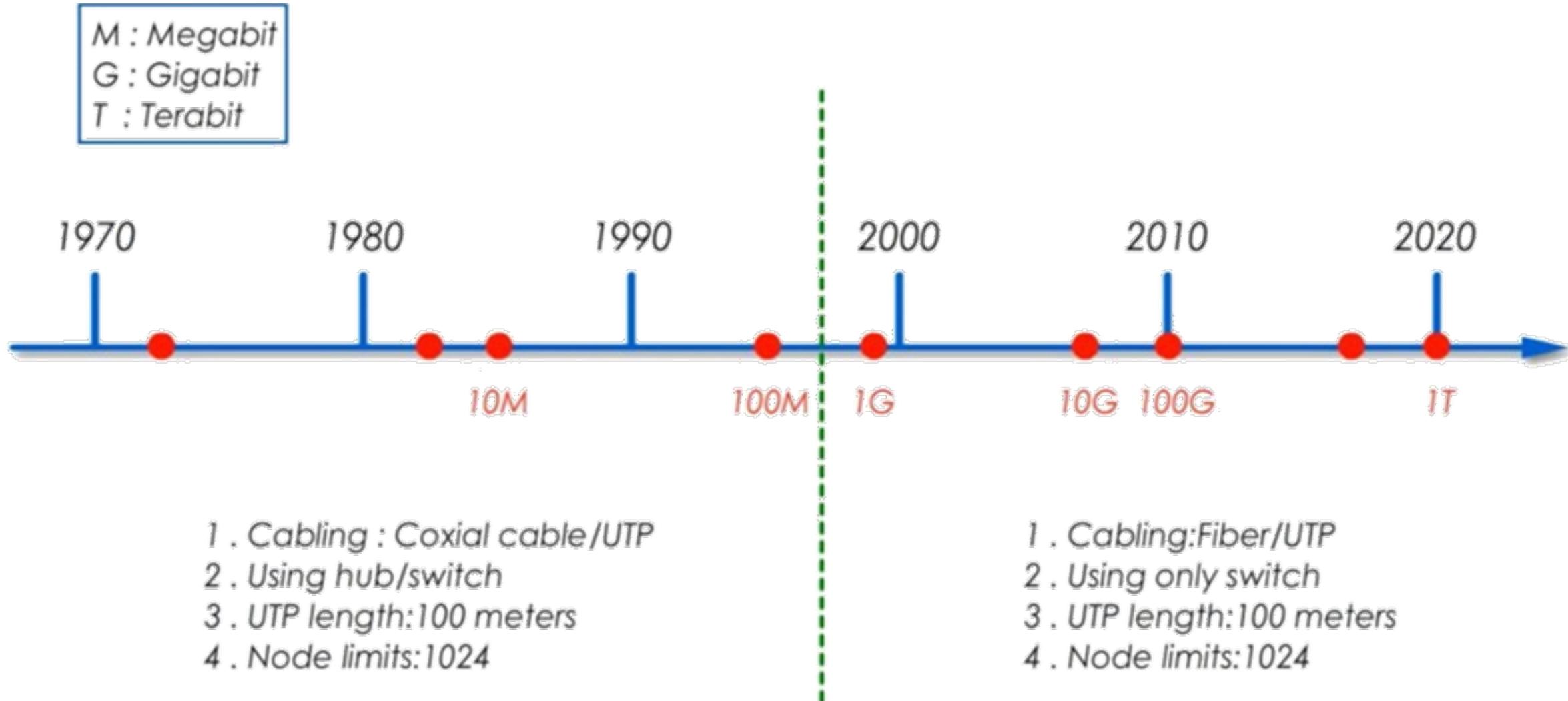
- The Computer Society of the **IEEE** began a special project in 1985 that was known as **Project 802**
- Its purpose was to set **standards** that would enable **intercommunication between equipment** from a variety of manufacturers
- **Project 802** is a way of specifying functions of the **physical layer and the data link layer** of major LAN protocols



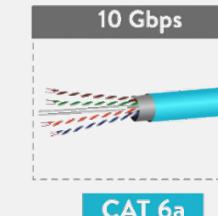
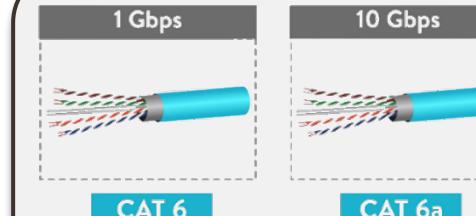
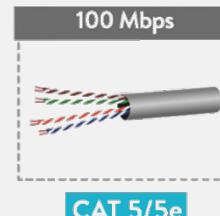
Ethernet Timeline



Ethernet Evolution – Physical Layer



Ethernet Cabling



Basis for Comparison	Ethernet	Fast Ethernet	Gigabit Ethernet
Speed	10 Mpbs	100 Mbs	1000 Mbps
Alternative Name (Baseband ver.)	10Base-T	100Base-T	1000Base-T 1000Base-LX (Fiber, long haul) 1000Base-SX (Fiber, short haul)
IEEE Standard	IEEE 802.3	IEEE 802.3u	IEEE 802.3ab IEEE 802.3z (Fiber standards)
Medium	Copper	Copper	Copper, Fiber
Maximum Network Segment Size	100 meters	100 meters	100 meters – copper 550 meters – SX 5 kilometers - LX



Data Link Layer: Services

LLC:

- Provide services to network layer protocols
- **Flow Control**
- **Error Control**

MAC:

- **Framing:** bits to frame (vice versa)
- Physical addressing (**MAC addressing**)
- Multiple access methods for channel-access control (**CSMA/CD, CSMA/CA**)
- LAN switching (packet switching), including MAC filtering, **Spanning Tree Protocol (STP)**
- Data packet queuing or scheduling
- Store-and-forward switching or cut-through switching
- Quality of Service (QoS) control
- **Virtual LANs (VLAN)**

From
Last Week!

IEEE 802.1 Data Link Layer

Two main Sub Layers

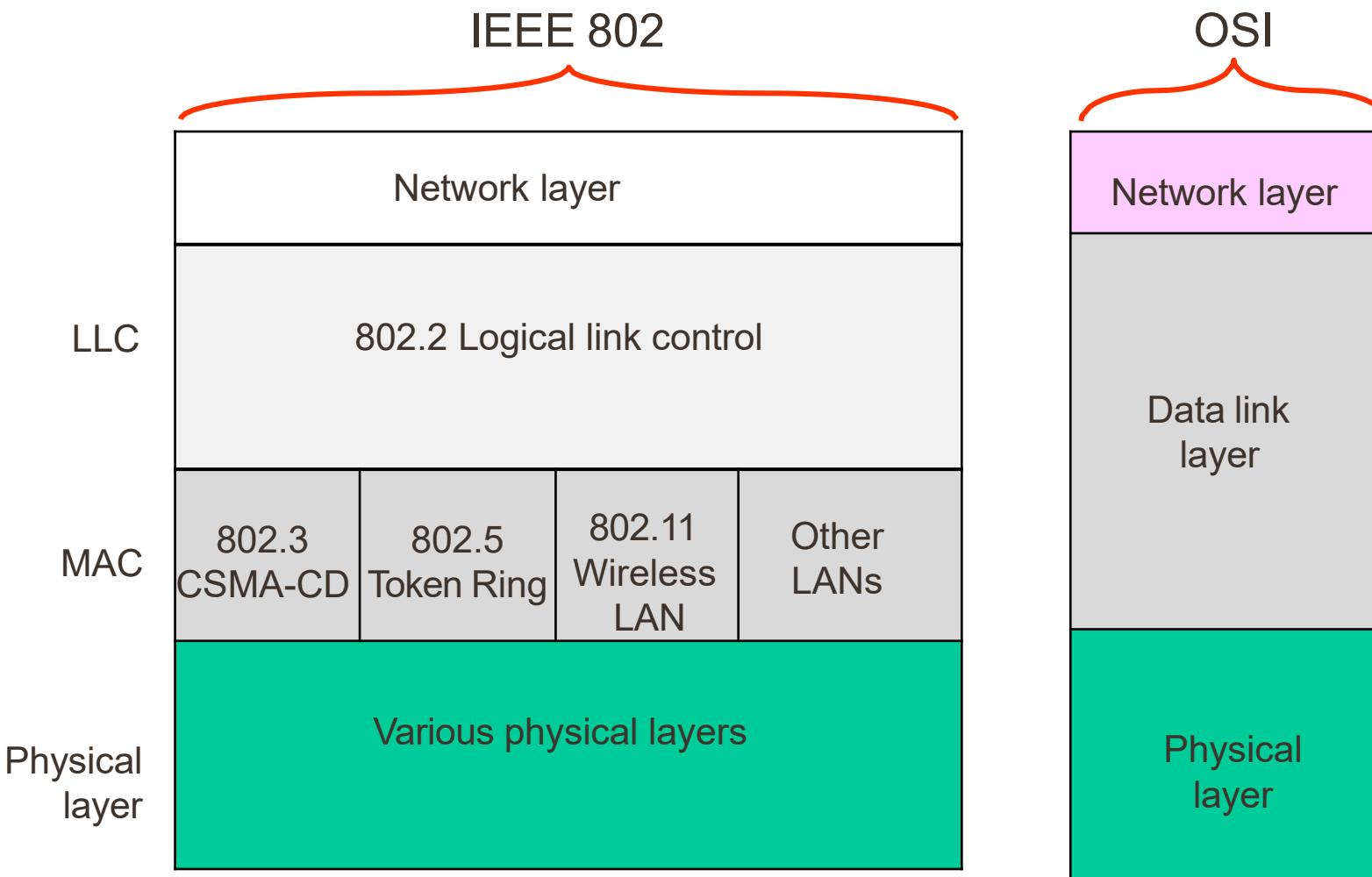
1. Logical Link Control (LLC)

- ✓ Between Network layer & MAC sublayer

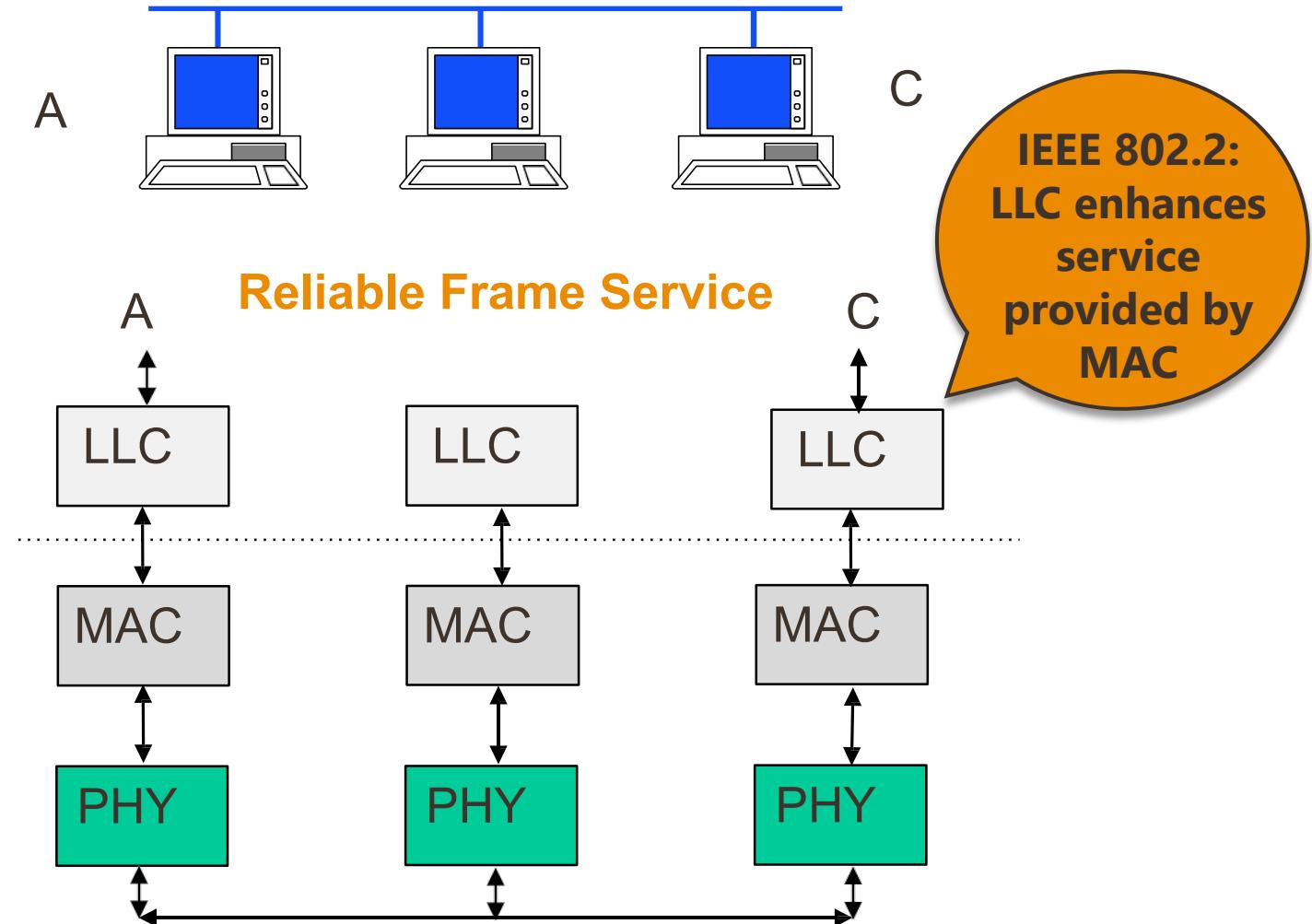
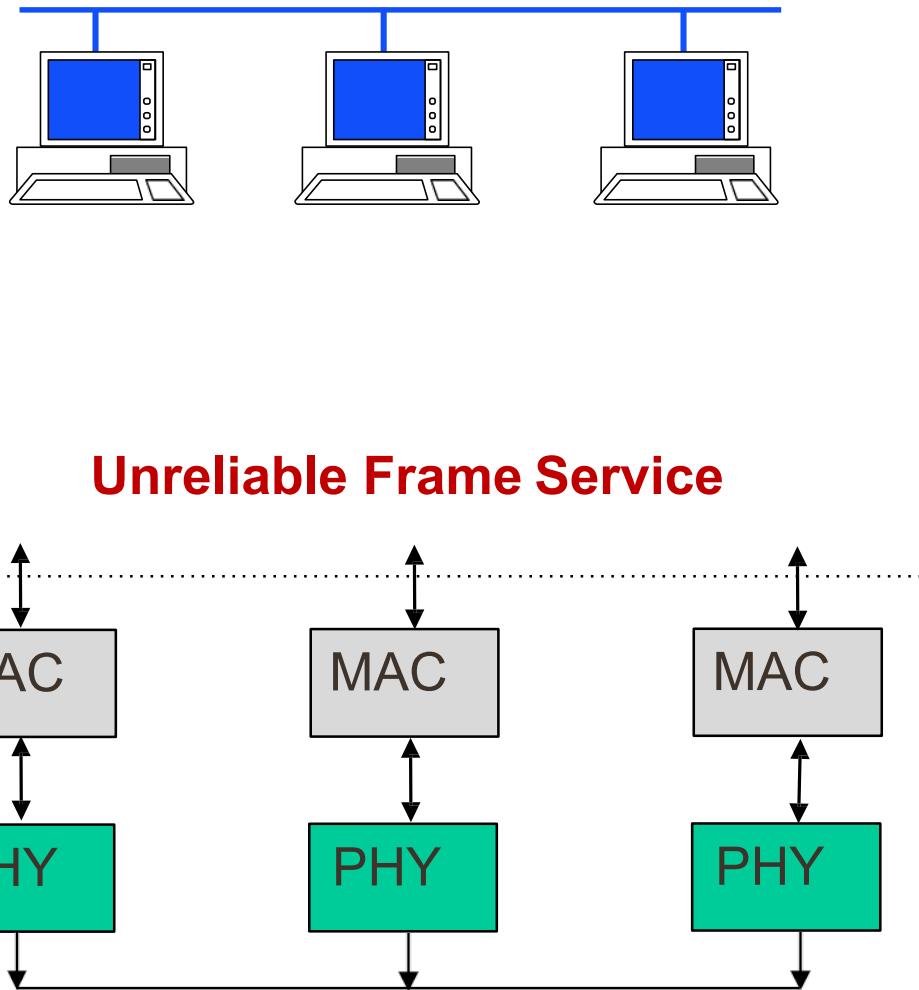
2. Medium Access Control (MAC)

- ✓ Coordinate access to medium
- ✓ Connectionless frame transfer service
- ✓ Machines identified by MAC/physical address
- ✓ Broadcast frames with MAC addresses

IEEE 802.1 Data Link Layer



Logical Link Control



Logical Link Control Services

- **Type 1: Unacknowledged connectionless** service

- ✓ Unnumbered frame mode of HDLC

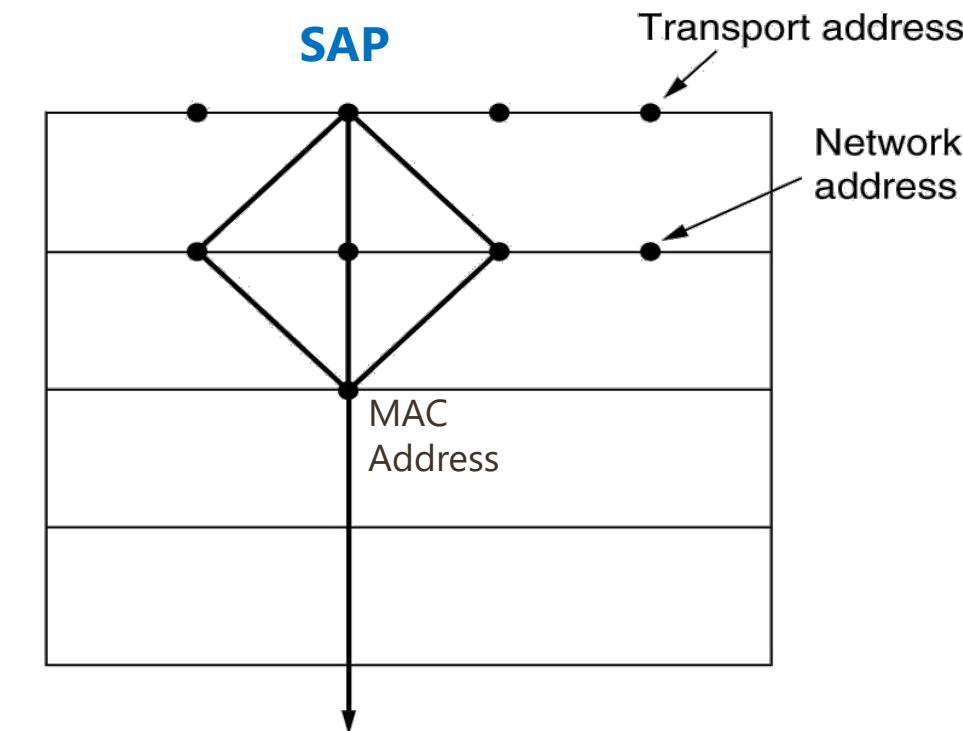
- **Type 2: Reliable connection-oriented** service

- ✓ Asynchronous balanced mode of HDLC

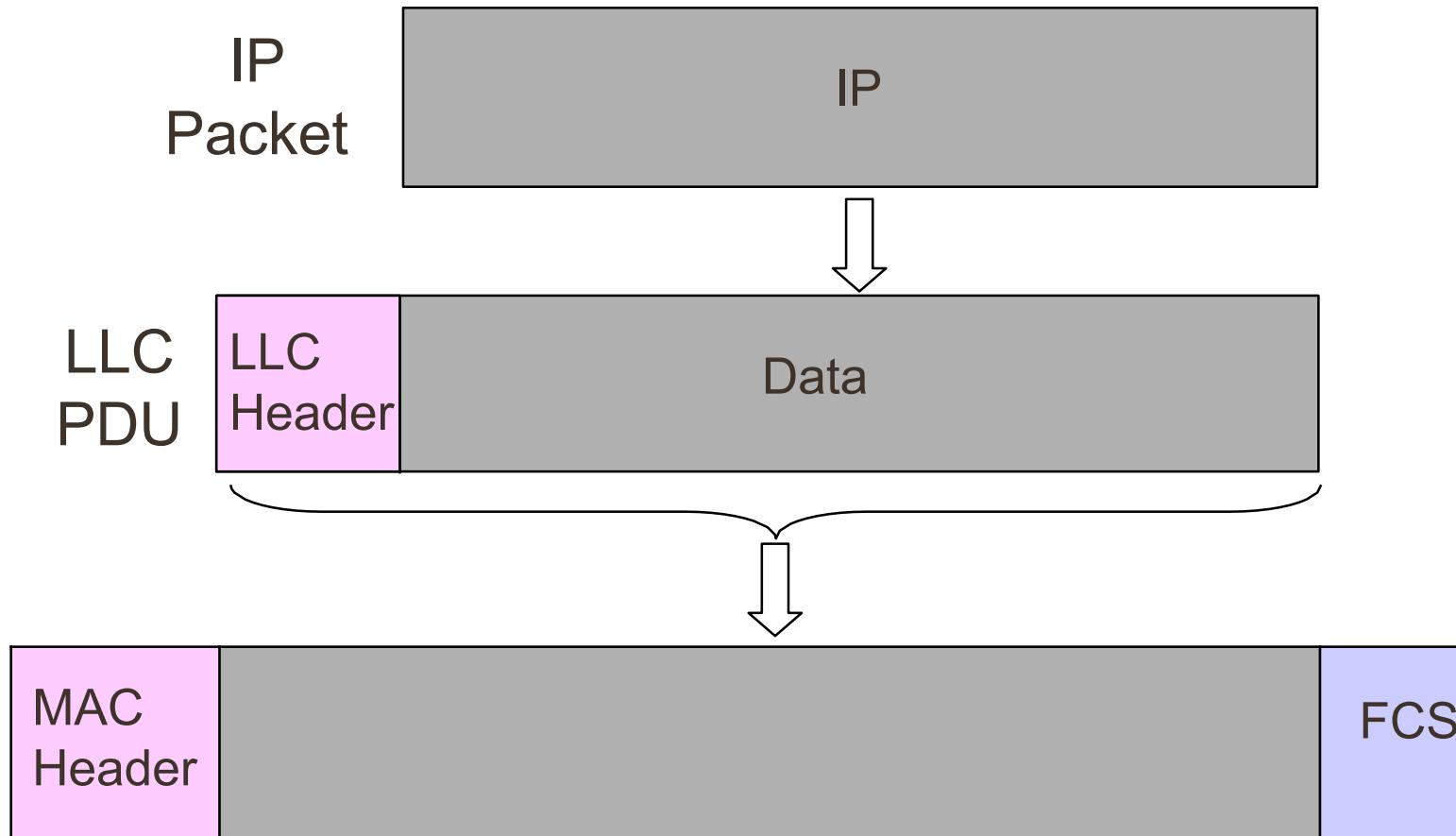
- **Type 3: Acknowledged connectionless** service

- **Additional addressing**

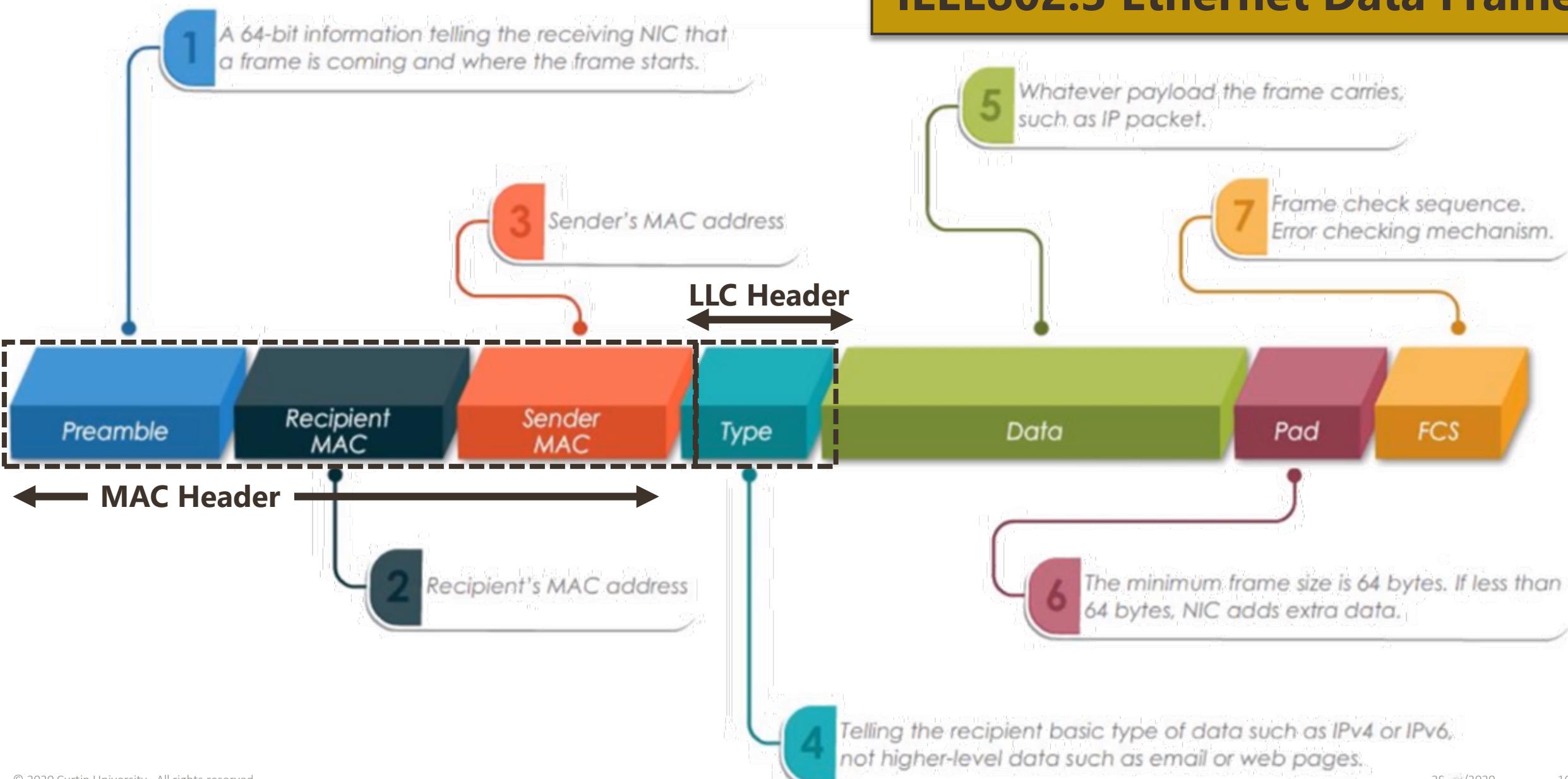
- ✓ A workstation has a single MAC physical address
 - ✓ Can handle several logical connections, distinguished by their **SAP** (service access points).



Encapsulation of MAC Frame



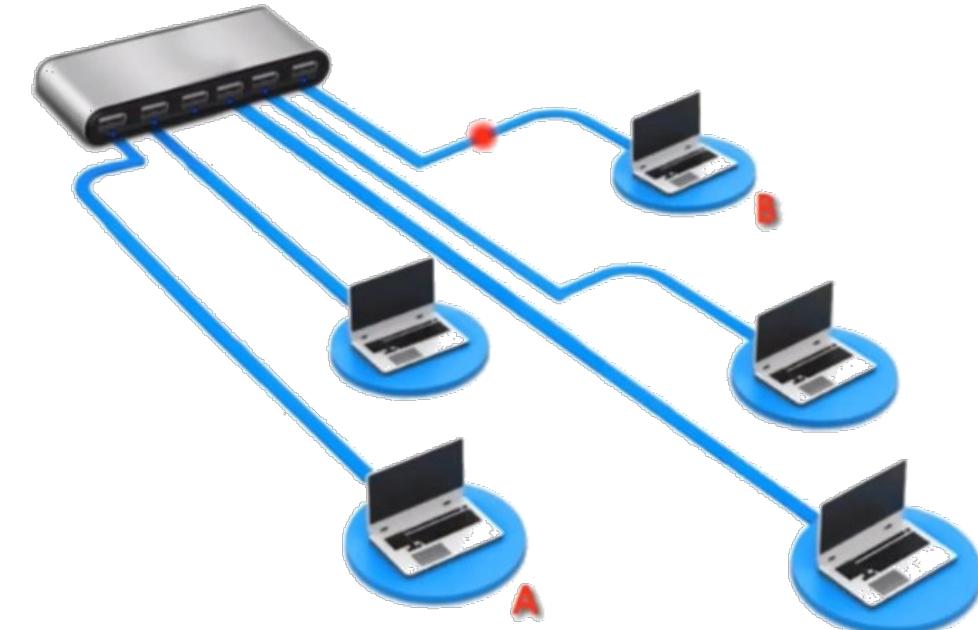
IEEE802.3 Ethernet Data Frame

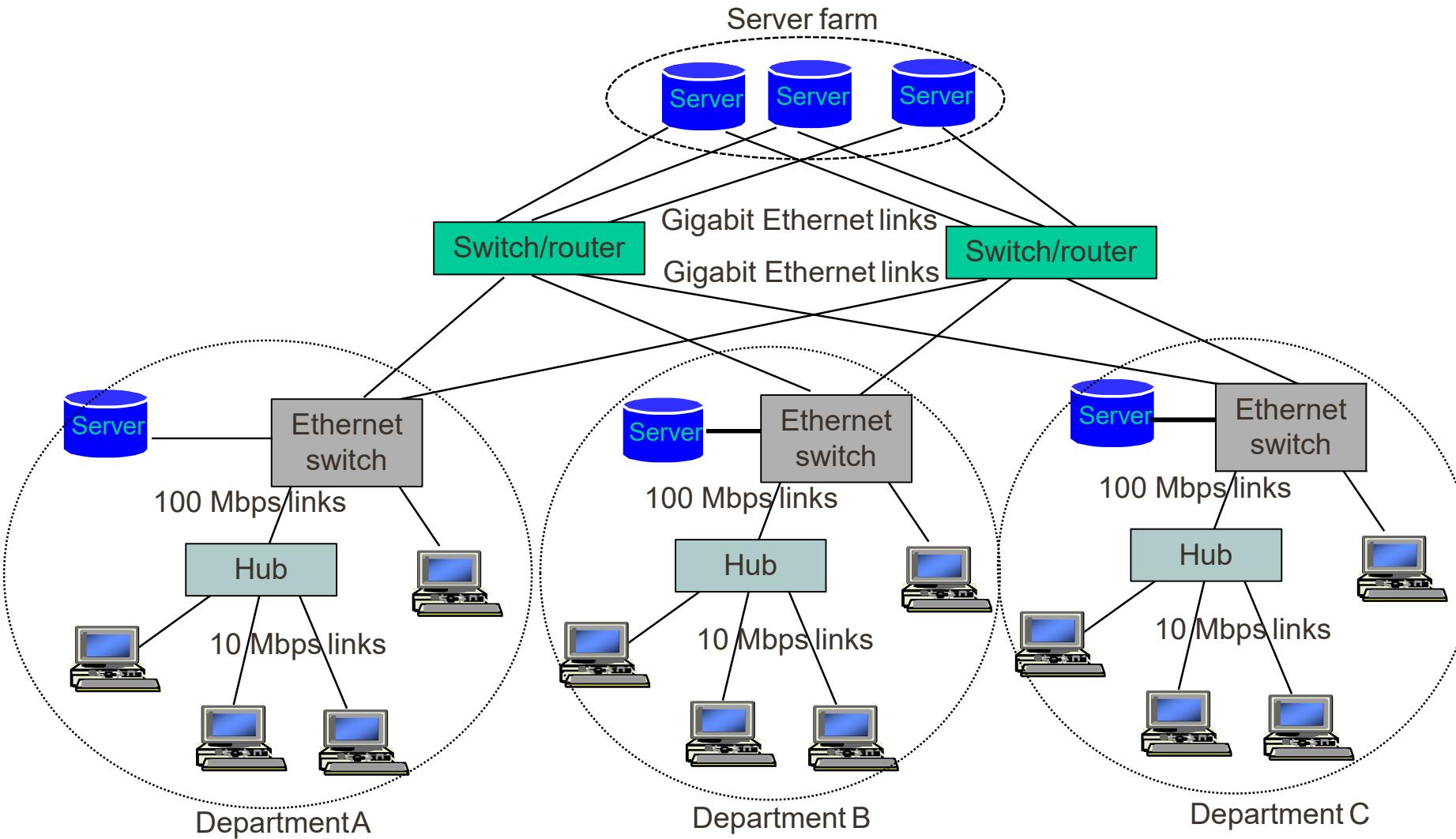


IEEE 802.3 Ethernet

▪ MAC Protocol: CSMA/CD

- ✓ Uses **binary exponential backoff delay**
 - After a collision has occurred each node waits either 0 or 1 time slots before retransmitting
 - If a further collision occurs each node waits 0,1,2 or 3 time slots
 - In general, after **n collisions**, a random number **between 0 and $2^n - 1$** time slots is chosen, and the node waits that number of time slots before attempting to retransmit, for **$n \leq 10$**





Typical Ethernet Deployment

Adaptive Learning

- In a static network, tables eventually store all addresses & learning stops
- In practice, stations are added & moved all the time
- Introduce timer (minutes) to age each entry & force it to be relearned periodically
- If frame arrives on port that differs from frame address & port in table, update immediately



Collision vs Broadcast Domain

- What is collision, broadcast domain?
- Hub
- Bridge
- Switch
- Router
- Wi-Fi collision/broadcast domain
- Full/half duplexity and collision domain

Collision vs. Broadcast Domain

▪ Collision Domain

- ✓ A group of nodes that can hear each other
- ✓ A part of a network where packet collisions can occur

▪ Broadcast Domain

- ✓ Contains devices that can reach each other at the data link layer by using **broadcast**

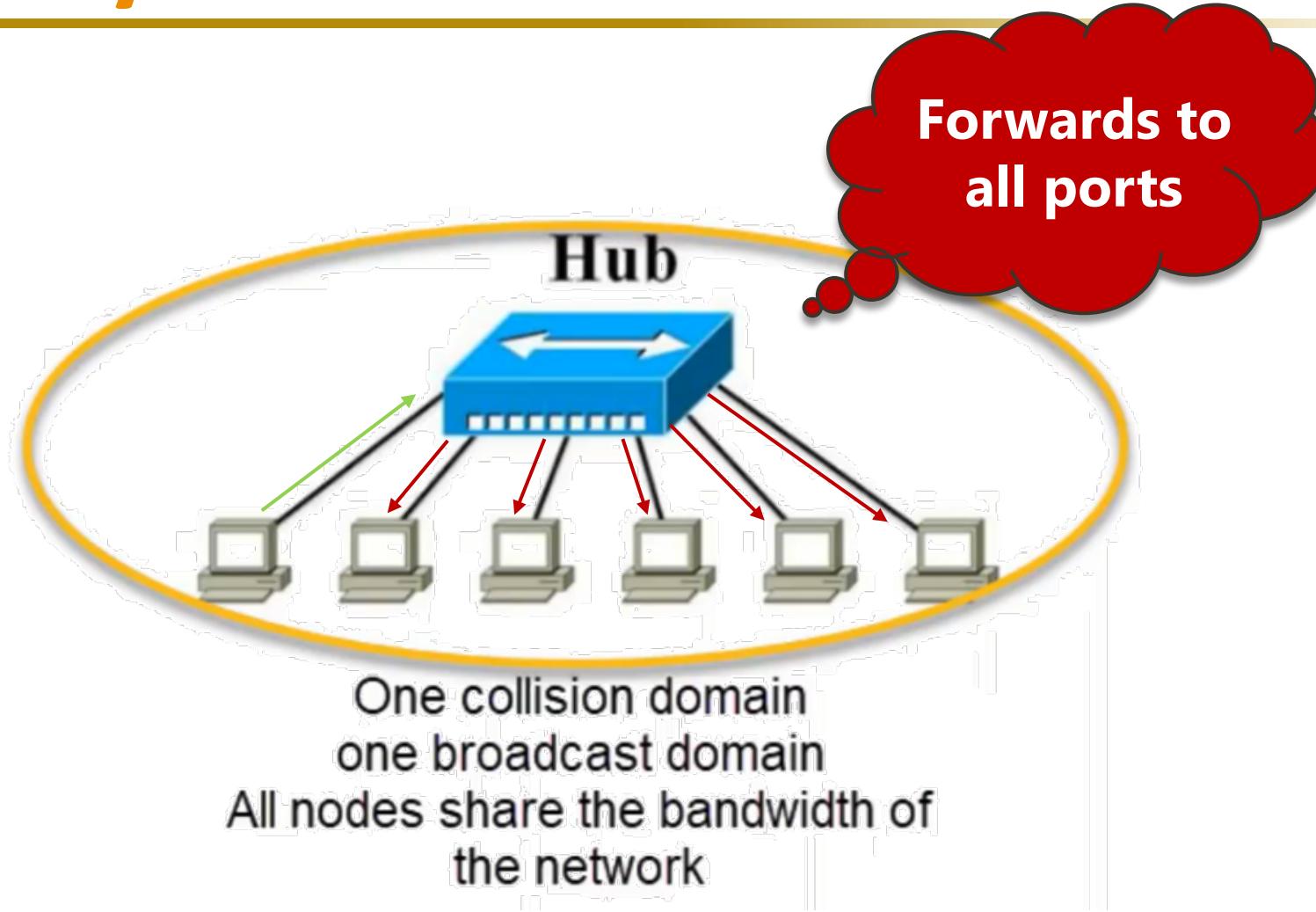


✓ **A broadcast stops at the router**

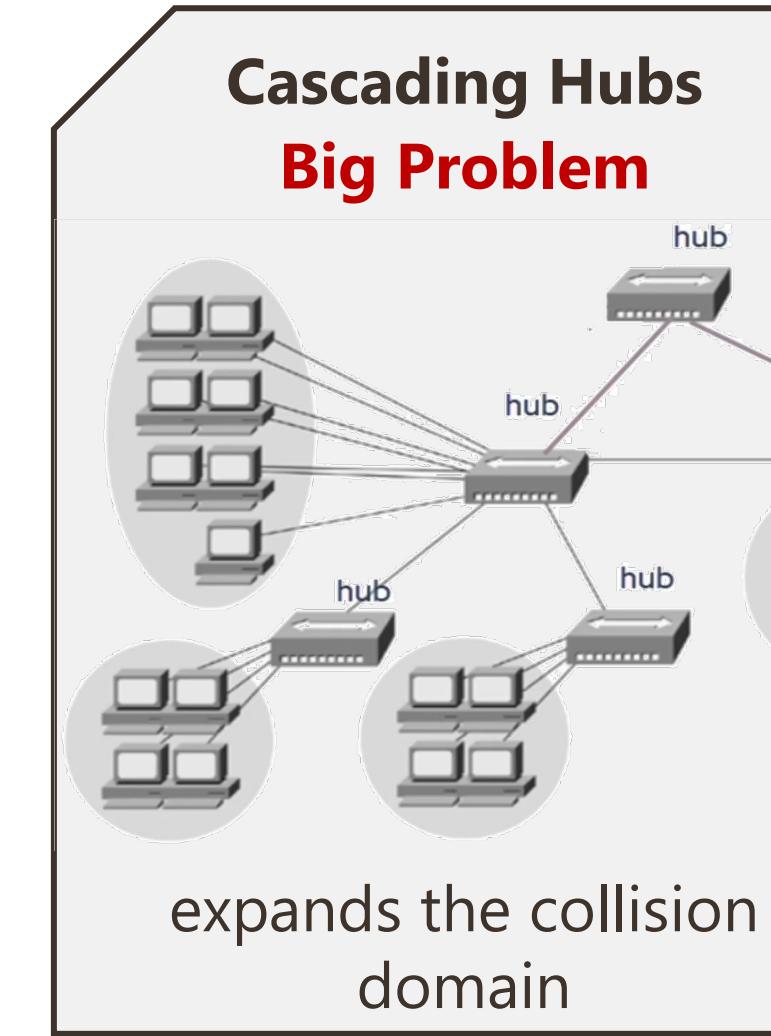
- It removes the header of the frame including the destination MAC address with another destination MAC address to send it to another broadcast domain



Layer 01: Hub



Only operate in **Half Duplex** mode

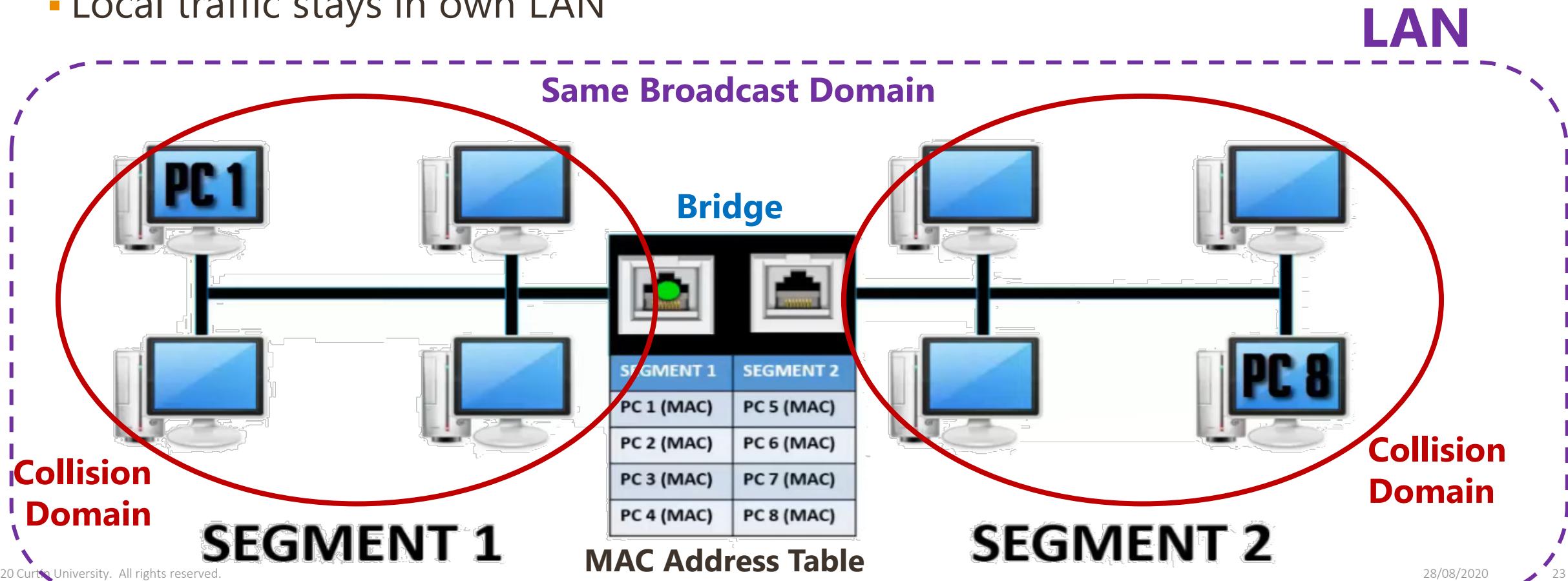


Layer 02: Bridge

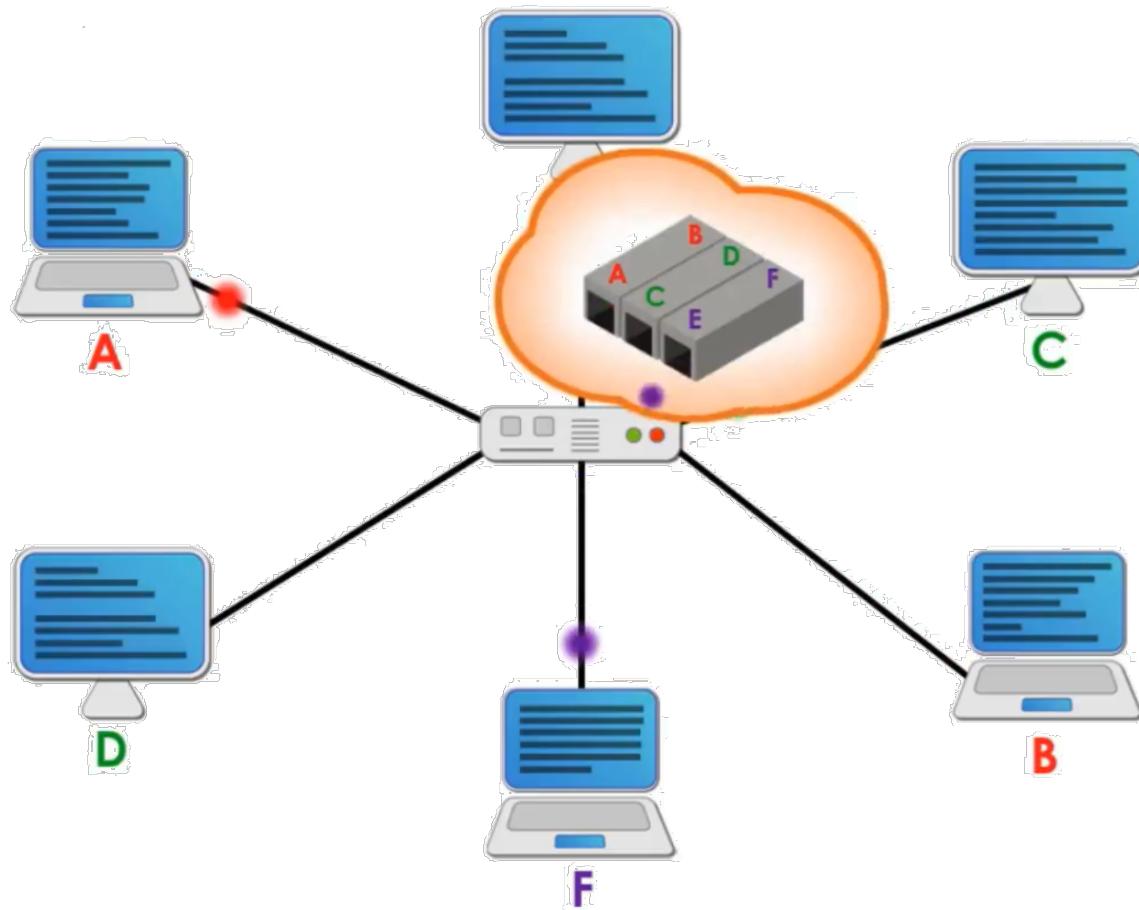
- Fewer Ports
- Replaced by Switch

- Segments a LAN

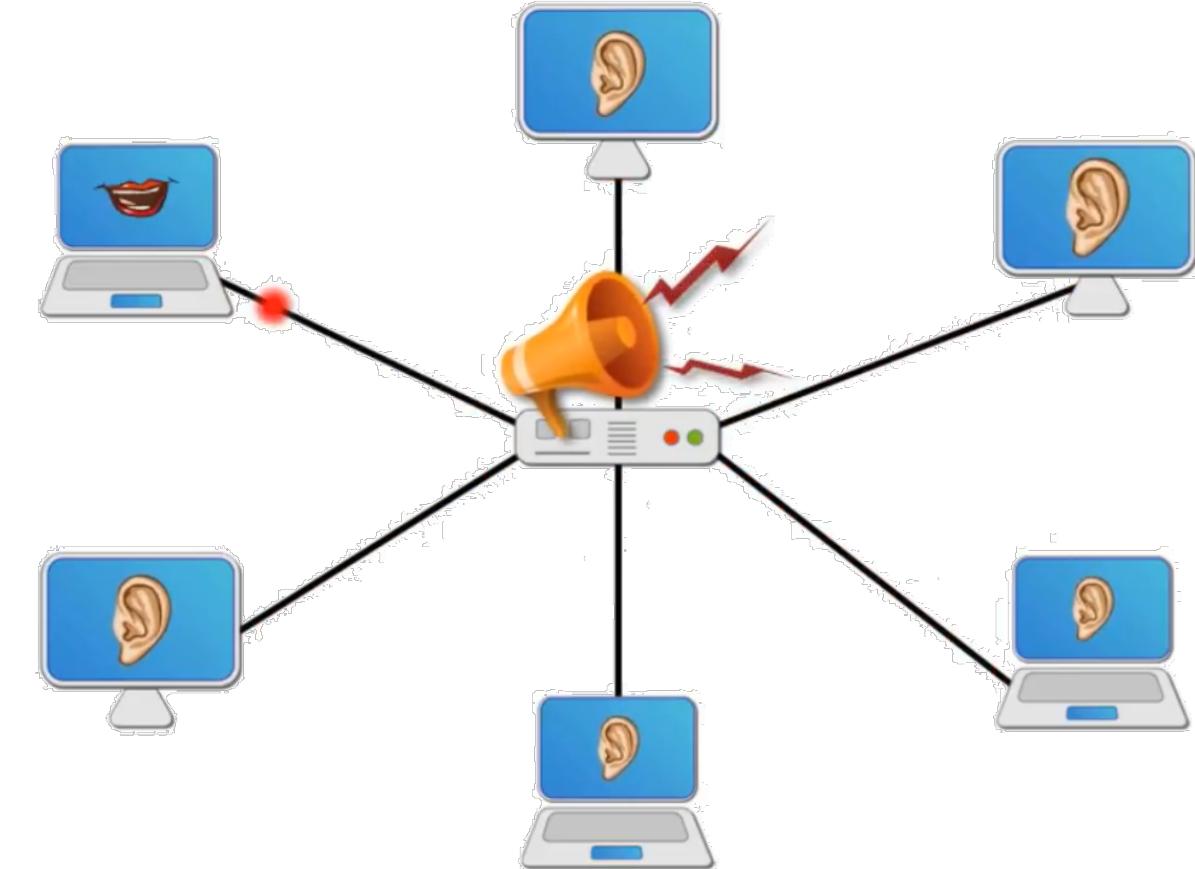
- MAC address filtering
- Local traffic stays in own LAN



Layer 02: Switch



Each port is a collision domain

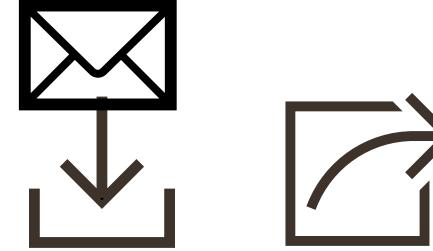


One broadcast domain

Can operate in **Full Duplex** Mode

Layer 02: Switch – cont.

- **Switch buffers** network **frames**. (**store-and-forward**)
- **Forward later** when the egress link is idle



This decoupling of receive and transmit operations enables a network that works with flows that are largely independent from each other and only compete for link bandwidth.

- **On a collision:**

- ✓ Any collision will not propagate across the switch
- ✓ Switch will retransmit the buffered frame later-on

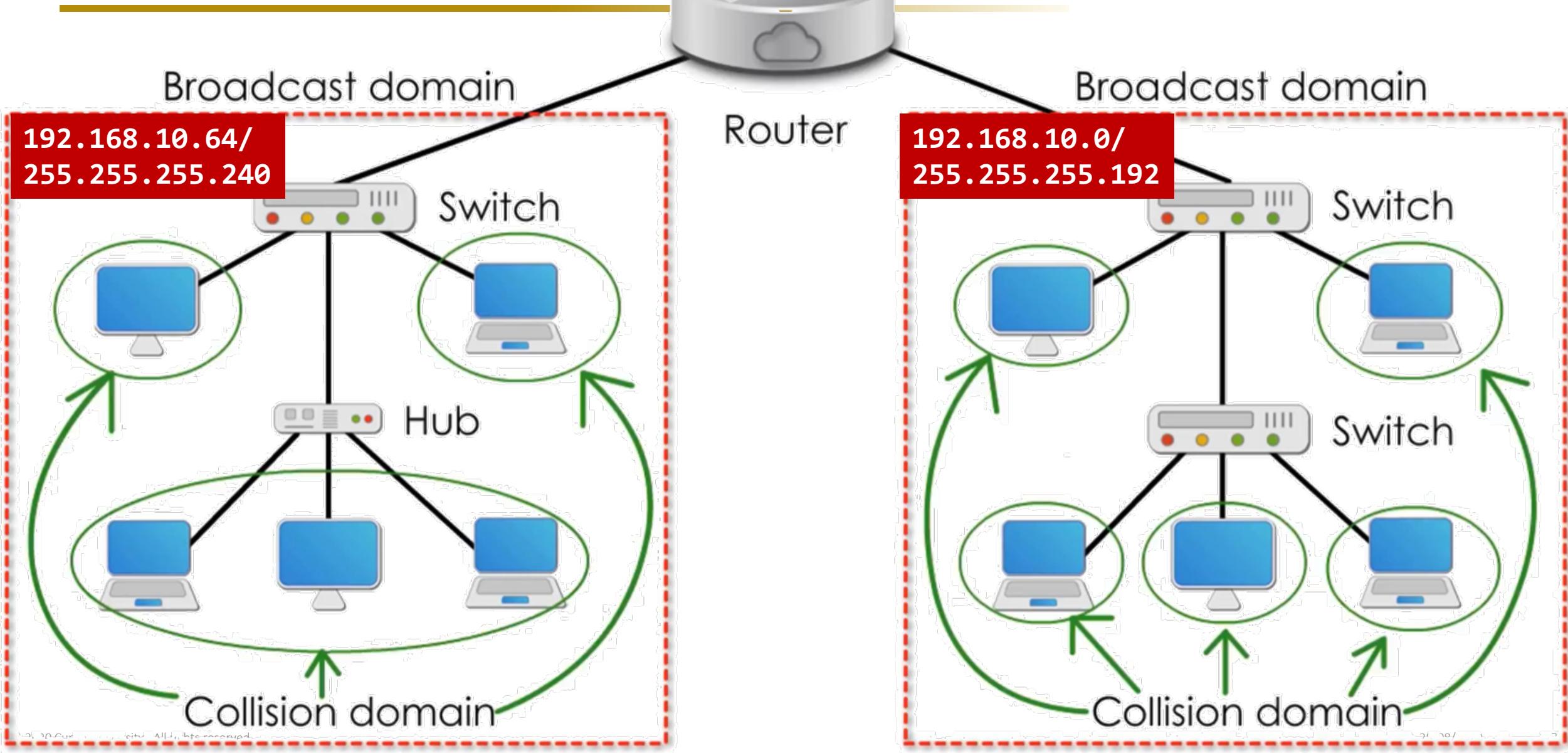
Switch segregates collision domains (on half-duplex links) or removes them completely on full-duplex links

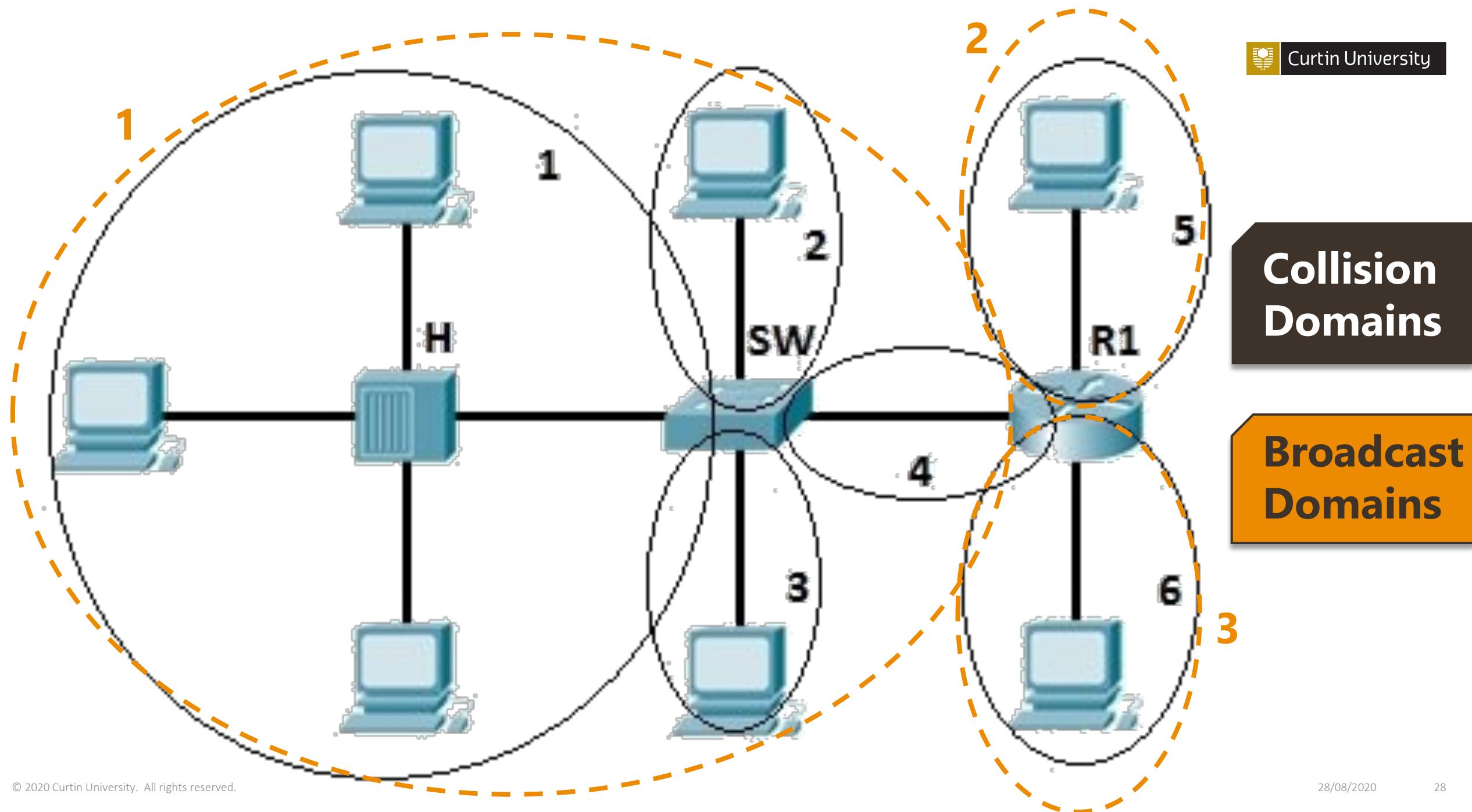
Layer 02: Switch – cont.

In contrast:

- **Hubs** repeat incoming bits as they are received (**No Buffering**)
- **On a collision:**
 - ✓ A collision on an egress interface will disrupt reception on the ingress interface
 - ✓ The hub needs to propagate an upstream collision back to the source
 - ✓ All nodes connected to a hub (or potentially chained hubs) form a single, common collision domain
- **Only one** node at a time **can transmit**

Layer 03: Router





**Collision
Domains**

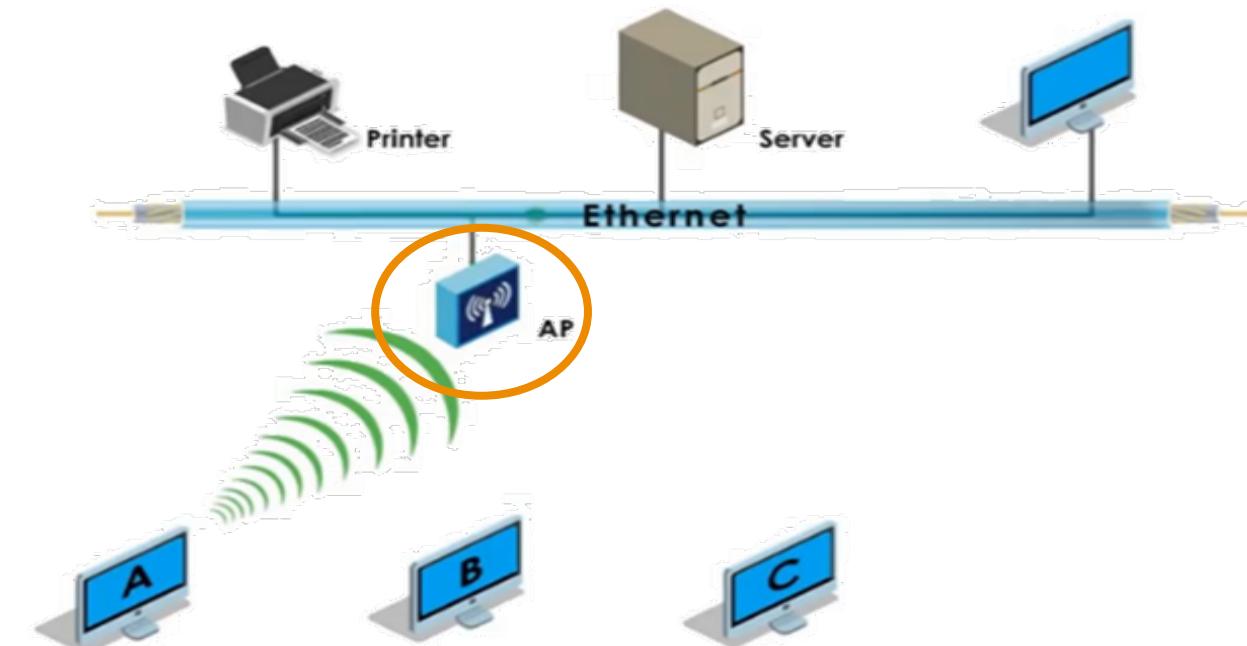
**Broadcast
Domains**

Collision/Broadcast Domains: Wi-Fi

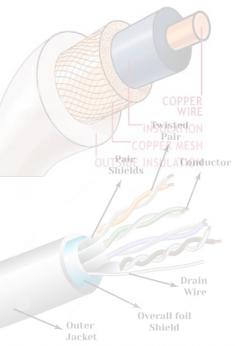
- In Wi-Fi, all devices share the same medium

- Hence collisions can occur during transmission
- Wi-Fi uses **CSMA/CA – Collision Avoidance (not CSMA/CD)** due to the **difficulty of detecting collisions**

- Wireless Access Point acts as a bridge between wired and wireless network
 - Collision / Broadcast domains are similar to bridge's we saw earlier



Full/Half Duplex and Collision Domains

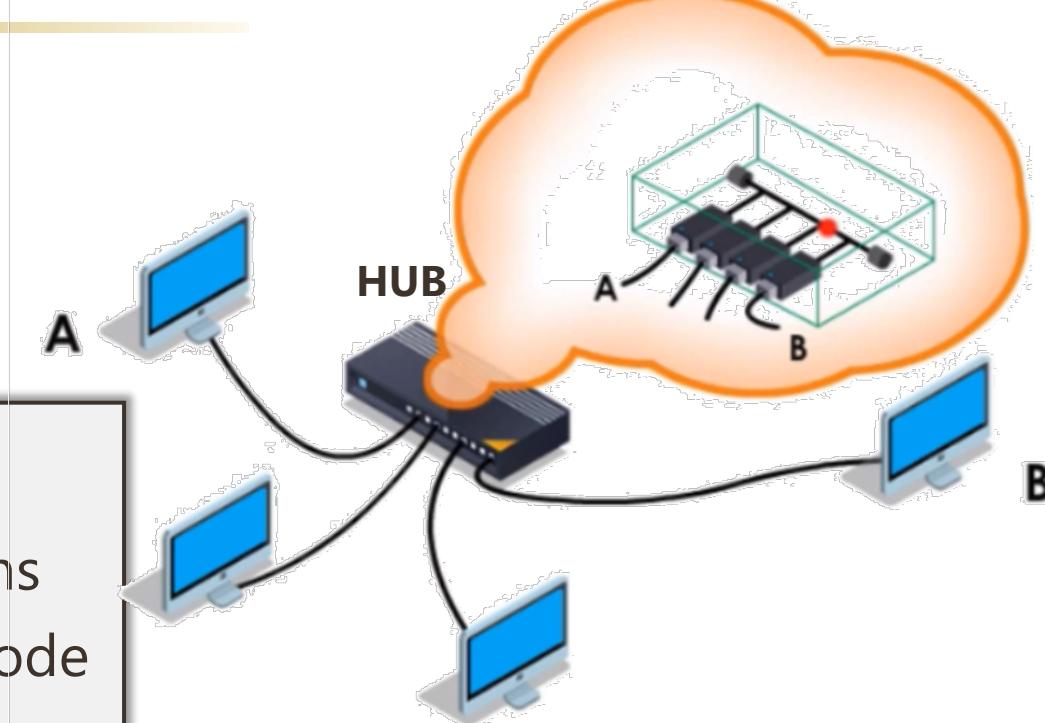
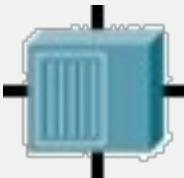


Coaxial cable: Half duplex

Fiber/UTP cable: Full duplex

▪ Fiber/UTP cable on a hub:

- ✓ Multiple devices share send and receive paths
- ✓ Hence **hubs do not support full-duplex** mode
- ✓ Must use half-duplex with CSMA/CD



▪ Fiber/UTP cable on a switch:

- ✓ Full-duplex Ethernet doesn't use CSMA/CD
- ✓ Separate send and receive paths between two devices (i.e. switch and host)





Wi-Fi Technology

- Fundamentals
- Wi-Fi Modes
- Wi-Fi Terms (BSS, BSSID, SSID, ESS)
- Wi-Fi Scanning Methods
- IEEE 802.11 Standards
- MIMO, CSMA/CA, OFDM
- 2.4 Ghz, 5Ghz Bands
- Wi-Fi Frames
- Wi-Fi Threats

Wi-Fi (Wireless Fidelity)

- the **IEEE 802.11 standard** defines the protocols that enable communications with current Wi-Fi-enabled wireless devices

- **What is a wireless access point?**

- Allows wireless devices to connect to the wireless network.

- **What is a mobile hotspot?**

Phone-as-a-modem

- a common feature on smartphones with both **tethered** and **untethered** connections
 - Used to share wireless network connection with other devices

- **What is portable Wi-Fi hotspot?**

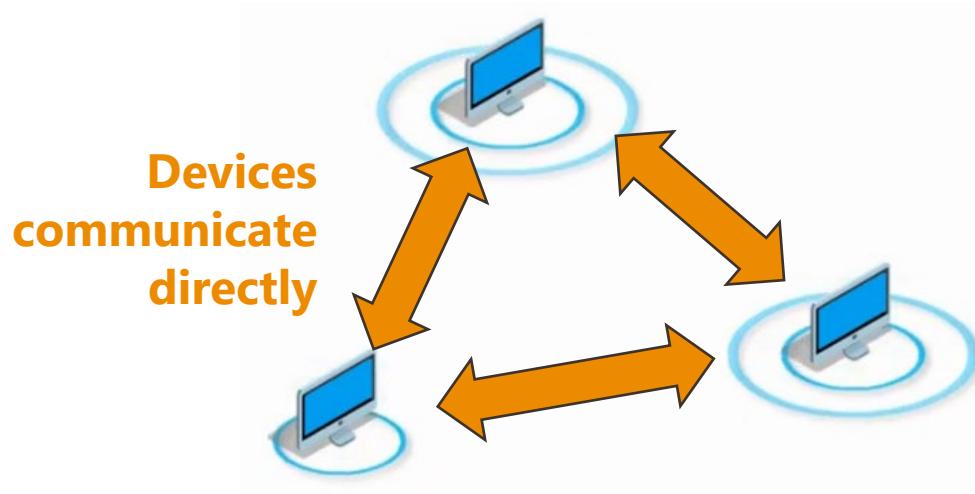
- a small device that uses cellular towers that broadcast high-speed 3G or 4G broadband signals



Wi-Fi Modes

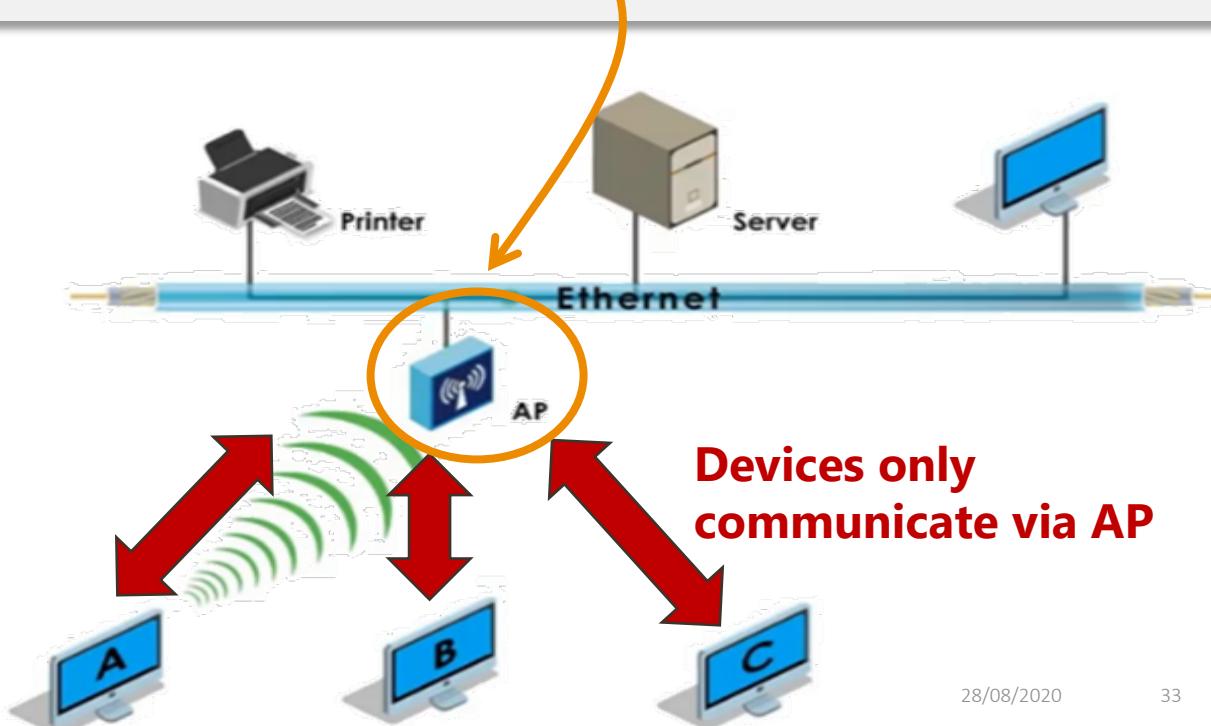
▪ Ad-hoc

- Decentralized (p2p mode)
- Easy/quick to create (no complex setup)
- More users join,
- Performance deteriorate, less secure



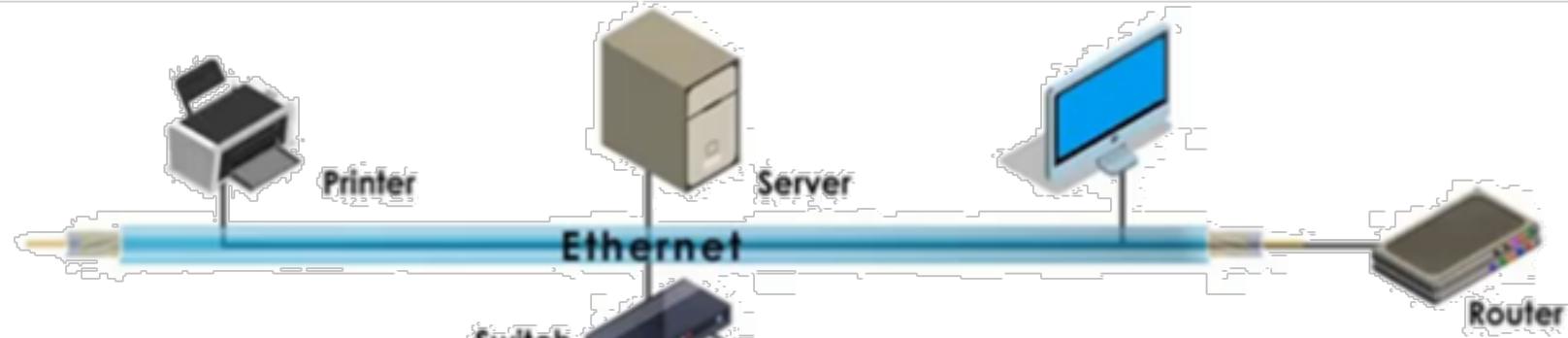
▪ Infrastructure Mode

- Centralized
- Communicate with Access Point (AP)
- AP acts as a bridge between **wireless traffic and wired network**





Wi-Fi Terms



Basic Service Set

Group of wireless devices working with the same AP

SSID: Service Set Identifier

String to Identify an AP

BSSID: AP's MAC Address

In ESS, every AP broadcasts the same SSID

ESS

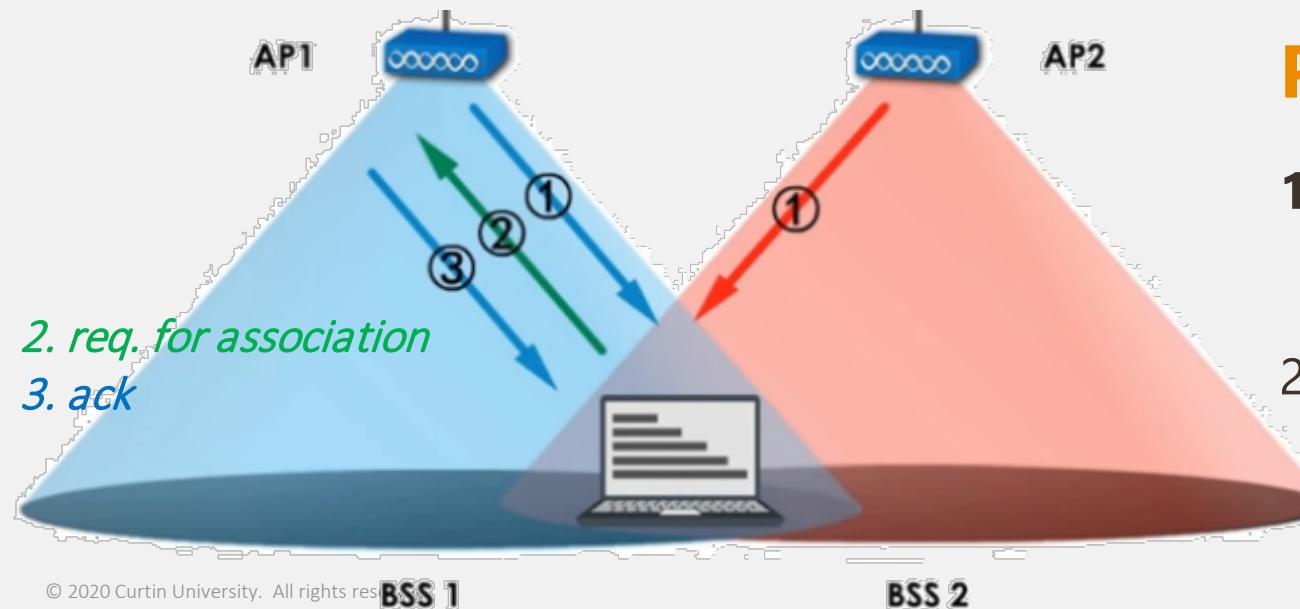
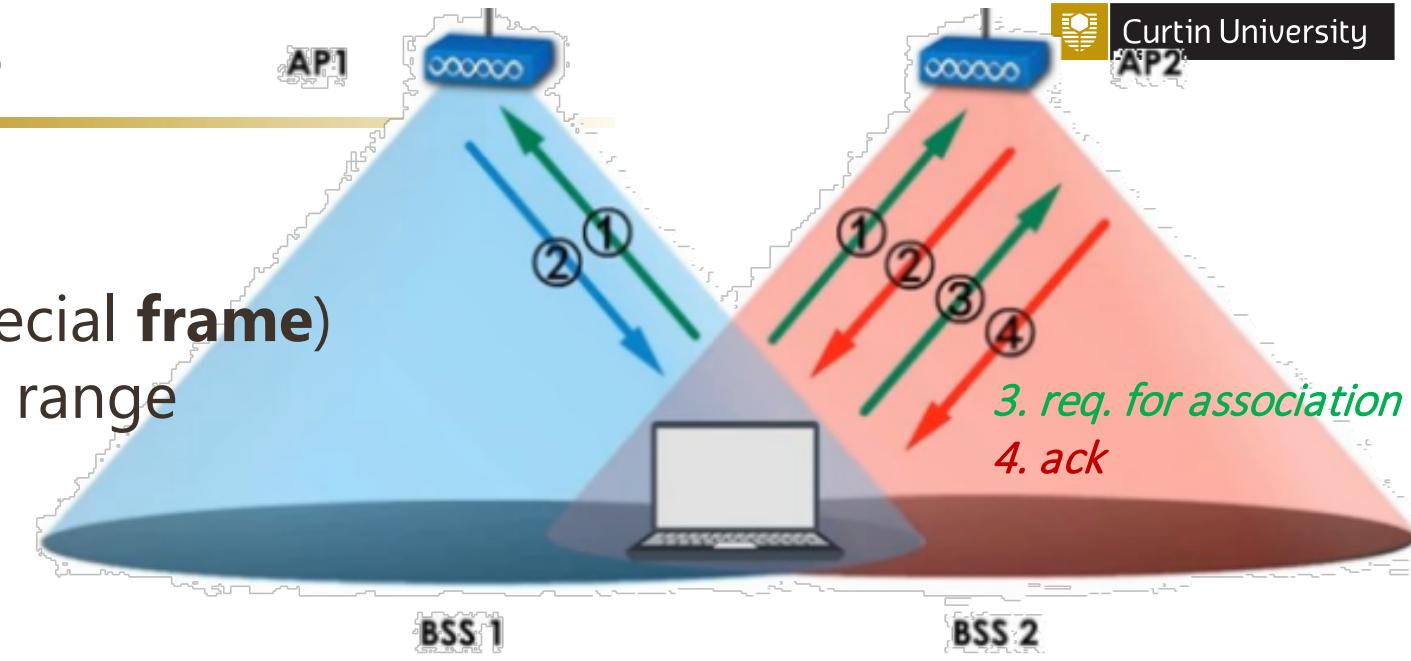
Extended Service Set

A service area can be extended by adding more APs

Scanning Methods

Active Scanning

1. **Device broadcasts a probe (special frame)** to each channel in its frequency range
2. Waits for an AP to respond



Passive Scanning

1. **Device listens** on all available channels within a frequency range
2. AP broadcasts a **beacon frame** continually

Contains AP information. i.e. SSID

Home Router: 5-in-1 magic box

- Integrated Functions

1. Router
2. Switch
3. DHCP Server (*for automatic IP assignment*)
4. DNS Server (*for domain name resolution*)
5. Access Point



In a **business setting**,
Router, Switch, DHCP server,
DNS server **are separated**

IEEE 802.11 Standards



All use **half-duplex** | **CSMA/CA** | **2.4GHz** and/or **5GHz** bands

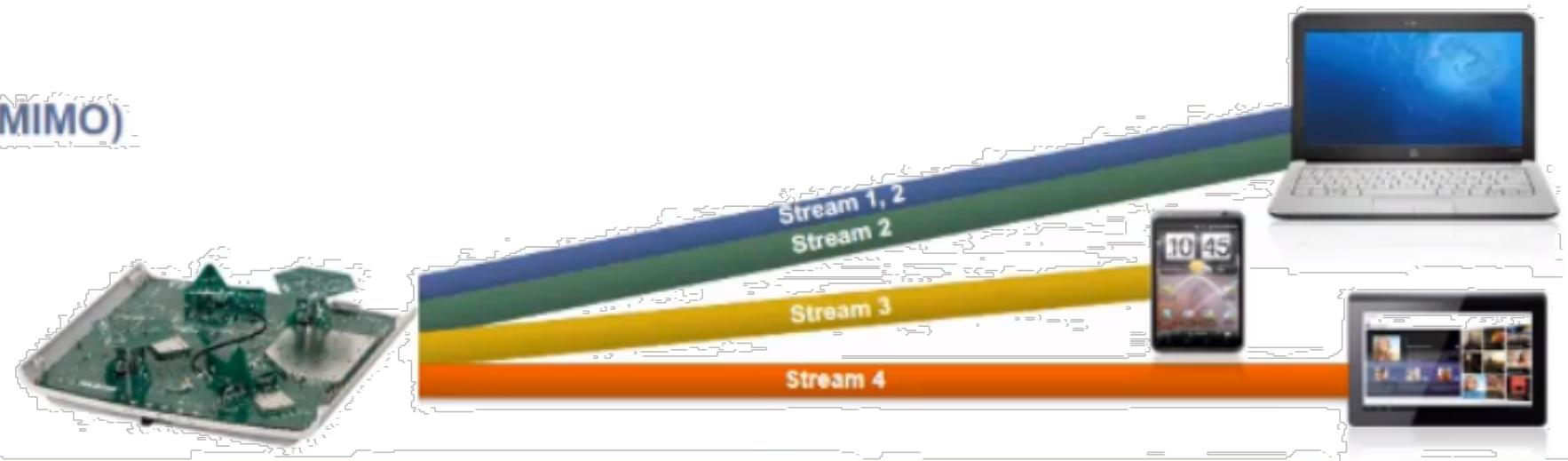
Standard	Year	Frequency Band	Speed	Modulation	Characteristics
802.11	1997	2.4GHz	1-2Mbps	DSSS,FHSS	Base version
802.11b	1999	2.4GHz	11Mbps	DSSS	Oldest, least expensive
802.11a	1999	5GHz	54Mbps	OFDM	Rarely used
802.11g	2003	2.4GHz	54Mbps	OFDM	Compatible with 802.11b networks
802.11n	2009	2.4GHz / 5GHz	65-600Mbps	OFDM	<ul style="list-style-type: none"> Backward compatible with 802.11a, b, g standards MIMO (multiple input-multiple output) Channel bonding: doubles the bandwidth Frame aggregation: reduces overhead
802.11ac WiFi 5	2014	5GHz	Up to 7Gbps	MIMO-OFDM	<ul style="list-style-type: none"> Gigabit Wi-Fi MU-MIMO (Multi User MIMO) Wave 1 (2014) vs. Wave 2 (2016)
802.11ax	2017	Wi-Fi 6 (802.11ax has 802.11ac features and more, 4x faster)			

Multi Input Multi Output (**MIMO**)

Single user MIMO (SU-MIMO)

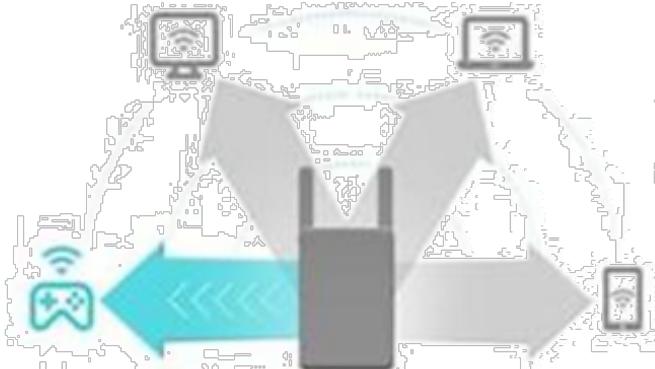


Multi-user MIMO (MU-MIMO)



More Connections and Faster for Everyone

MU-MIMO technology allows the RE650 to serve up to four devices at once, reducing wait time and greatly increasing WiFi throughput for every device. With MU-MIMO, the RE650 runs up to 4x faster than traditional AC range extenders.



Traditional RE

Sends data to one device at a time



MU-MIMO RE650

Simultaneously sends data to multiple devices

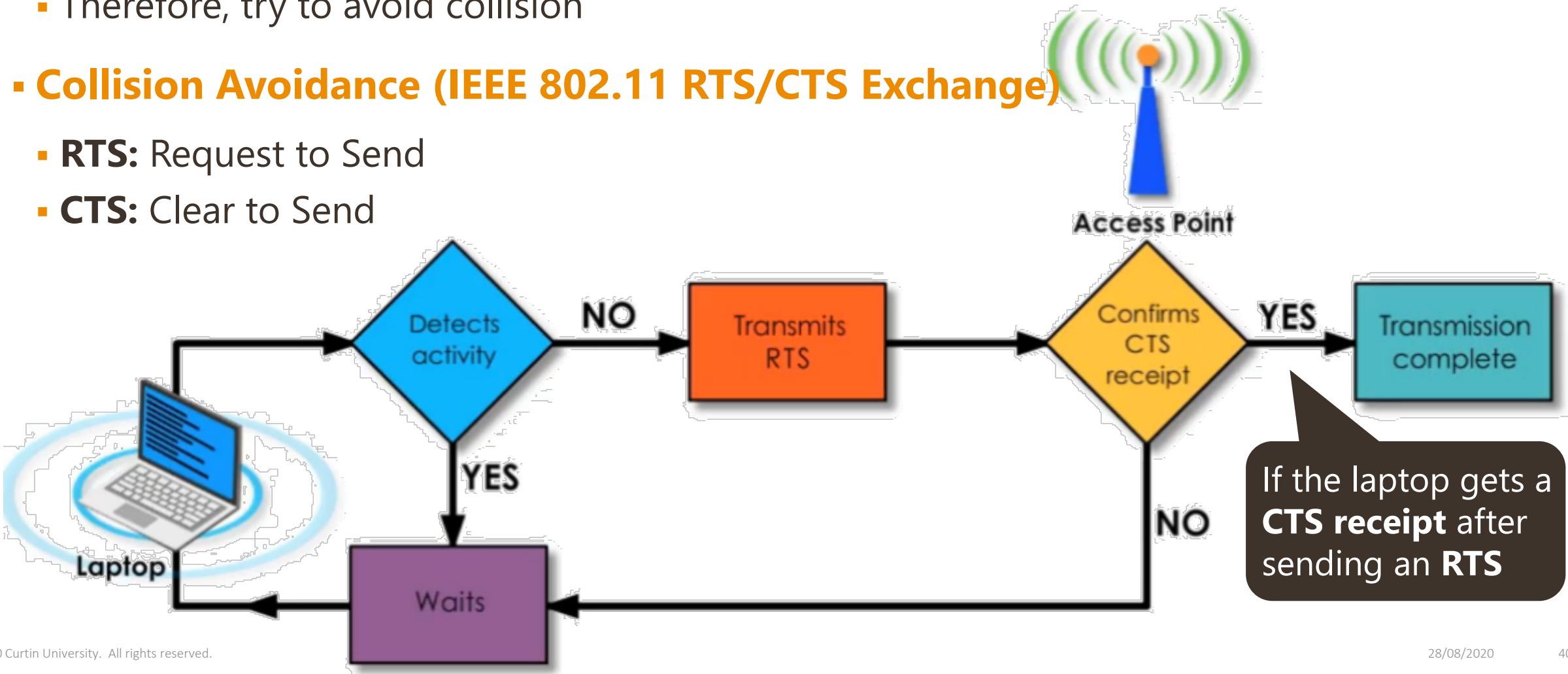
MU-MIMO

IEEE 802.11 - CSMA/CA

- Wireless nature makes harder to detect collision
 - Therefore, try to avoid collision

Collision Avoidance (IEEE 802.11 RTS/CTS Exchange)

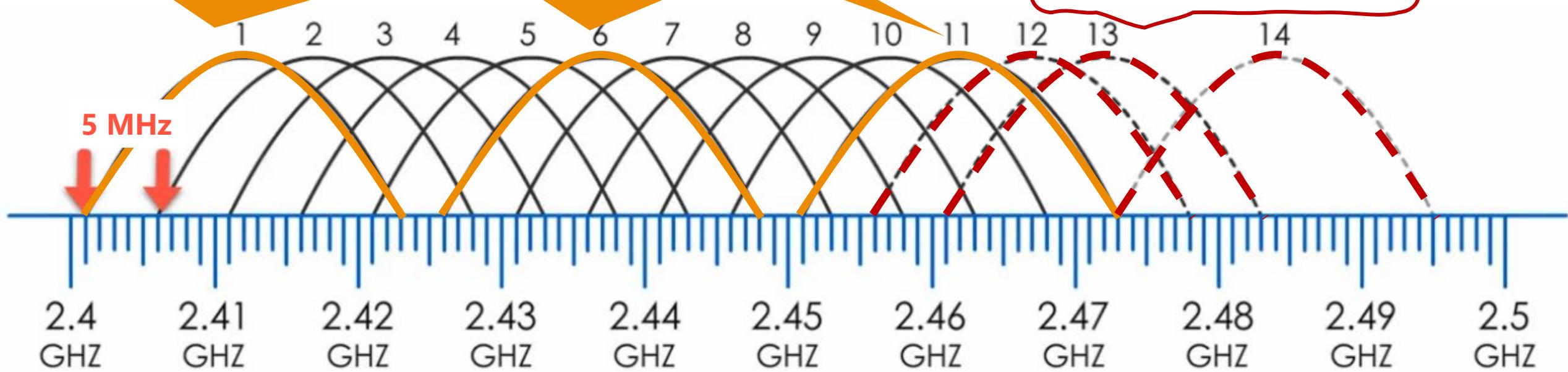
- RTS:** Request to Send
- CTS:** Clear to Send



2.4 GHz Band

Only 3 channels can be active since many overlaps

12, 13, 14 channels are restricted





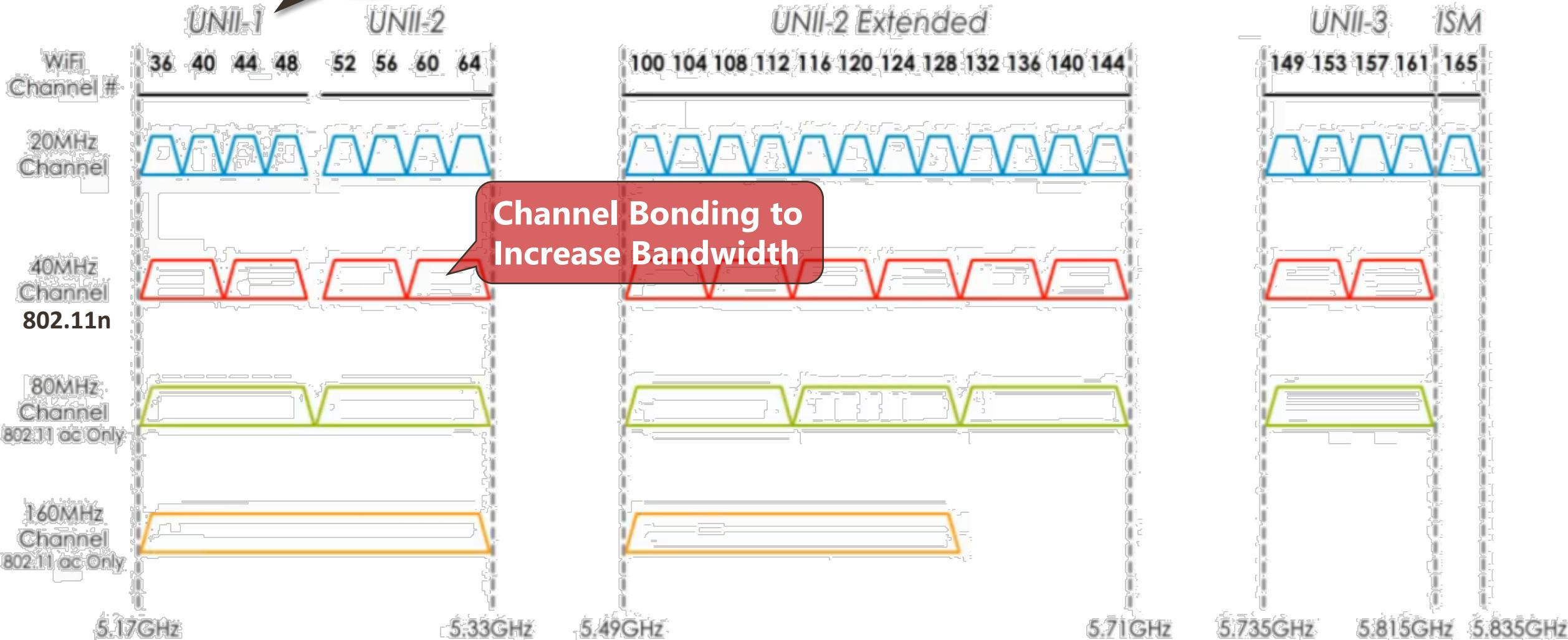
5 GHz Band

For indoor
Wi-Fi

Labels, Specifications, Regulations

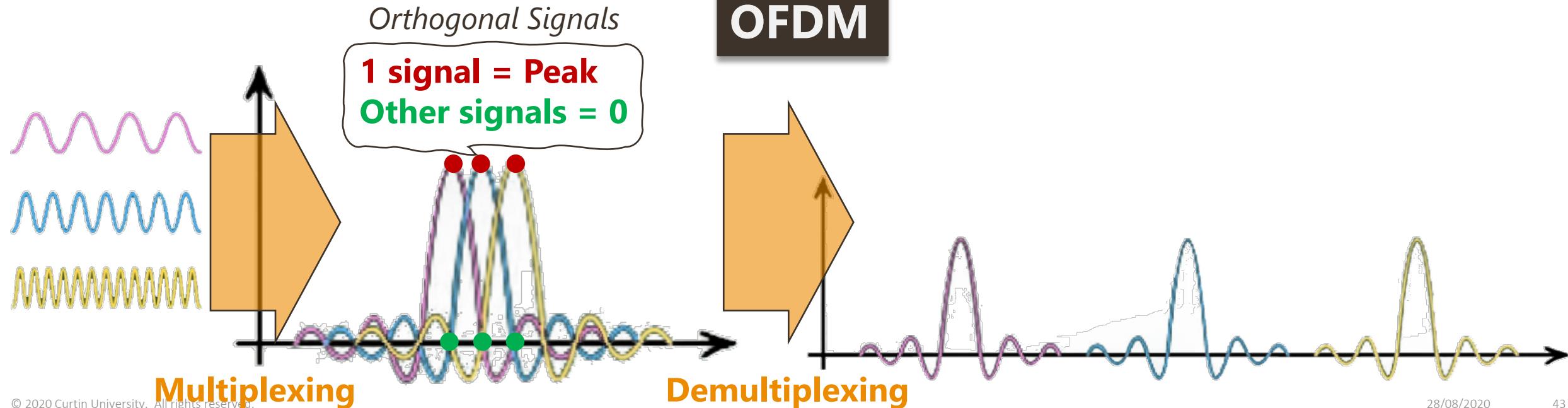
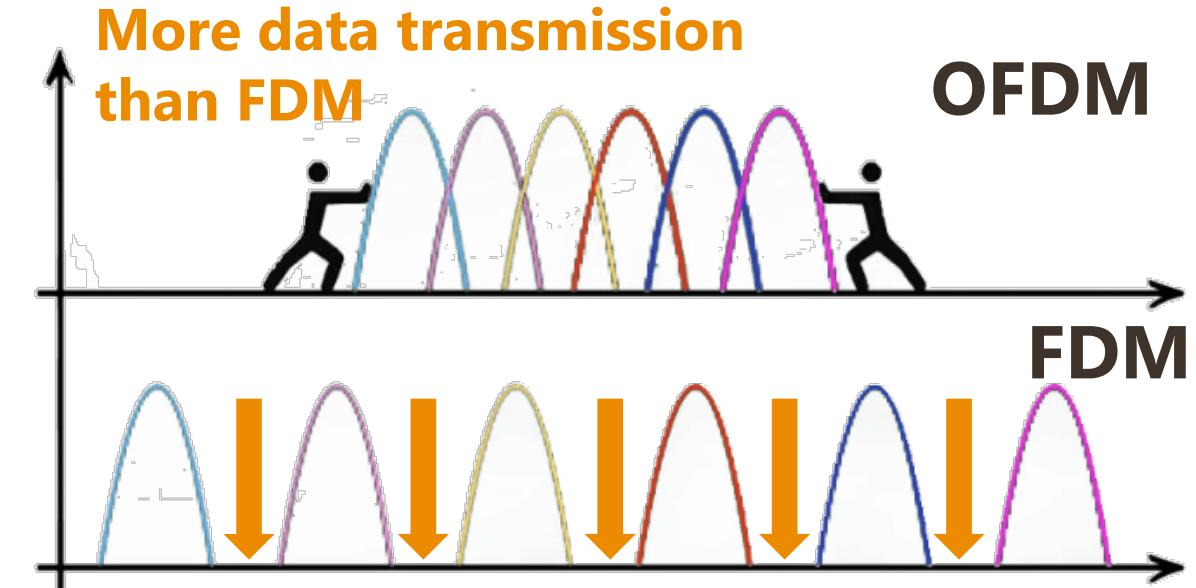
UNII: Unlicensed National Information Infrastructure

ISM: Industrial, Scientific and Medical



OFDM

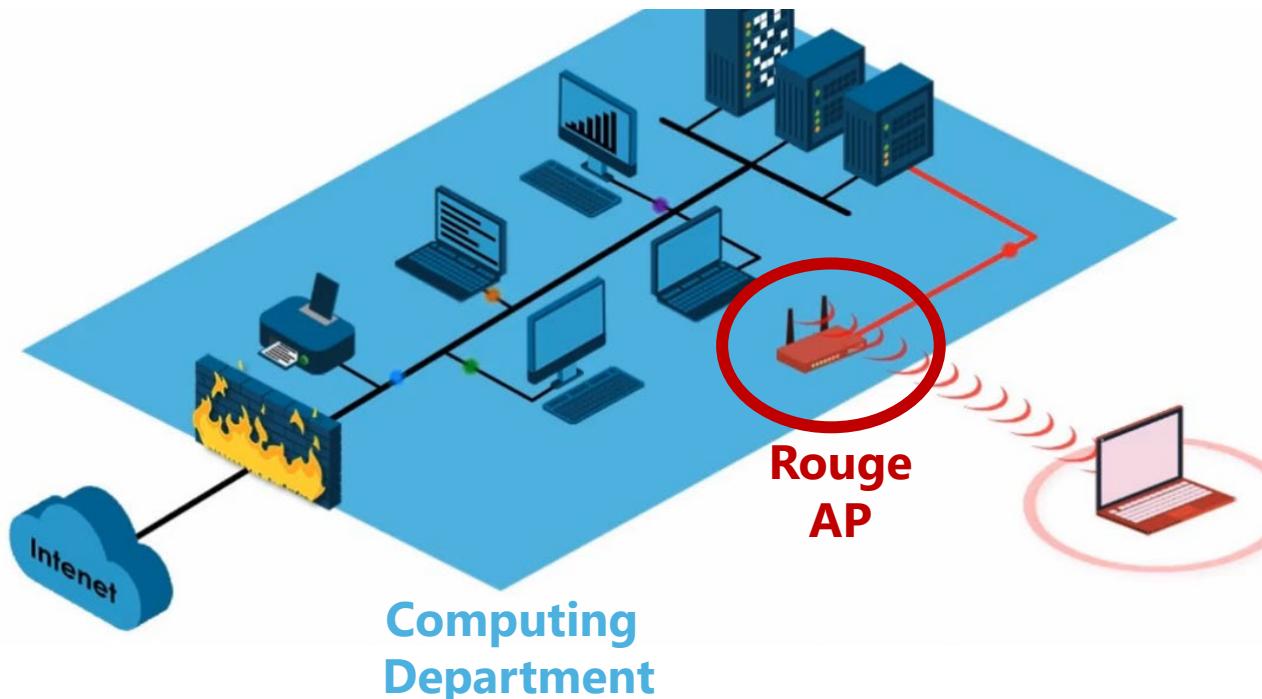
- Orthogonal Frequency Division Multiplexing
- Used by Wi-Fi 802.11ac/ax (Wi-Fi 6), 4G, 5G, WiMAX, Satellite, etc.



Wireless Threats

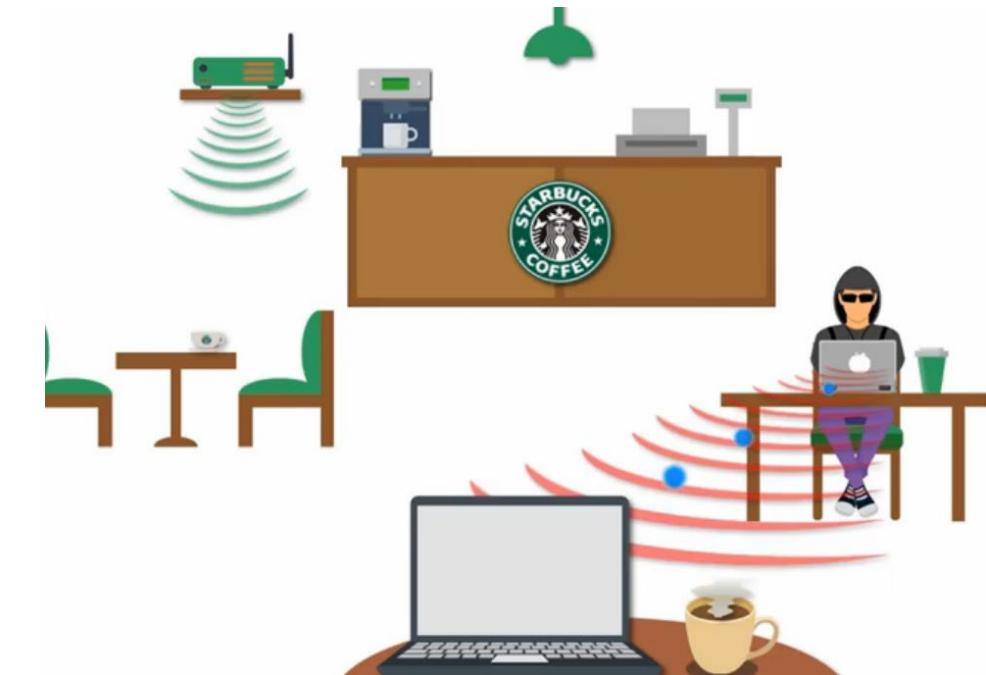
Rouge Access Point

- Threat to the **private network**
- Wireless backdoor
- By-pass firewall



Evil Twin

- Threat to the **end-user**
- Poses as a legitimate Wi-Fi access point



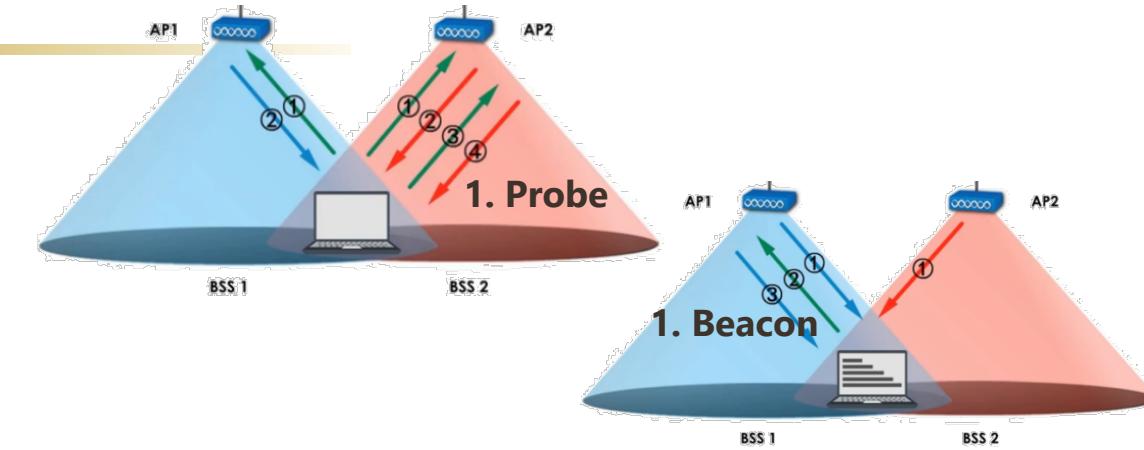
IEEE 802.3 Ethernet vs 802.11 Wi-Fi

Basis for Comparison	IEEE 802.3 Ethernet	IEEE 802.11 Wi-Fi
Layer 2 Frame:	Data Frame	<ol style="list-style-type: none">Management FrameControl FrameData Frame
Layer 2 Devices:	Switch	Access Point (AP) (bridge between Ethernet & Wi-Fi)
Data Link Layer:	CSMA/ CD	CSMA/ CA
Layer 1 Topology:	Star (most popular)	Star or Mesh
Layer 1 Medium:	UTP, Fibre Optics, Coaxial	Air

Wi-Fi Frames

1. Management Frames involve in:

- ✓ Wireless association
- ✓ Re-association
- ✓ Probe
- ✓ Beacon

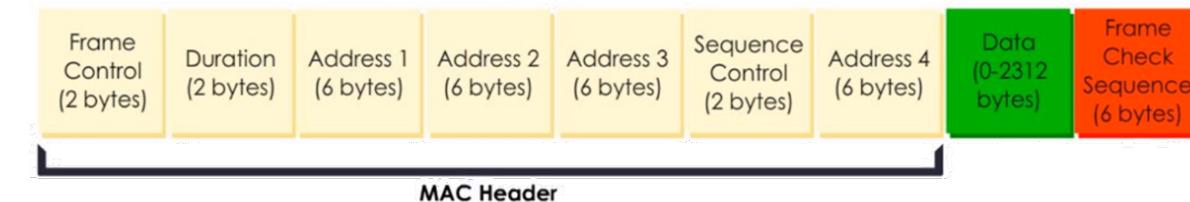


2. Control Frames involve in:

- ✓ Media Access and Data Delivery
- ✓ i.e. RTS/CTS frames in CSMA/CA

3. Data Frames involve in:

- ✓ Carry data (sender <-> receiver)



802.11 Wi-Fi Data Frame

Field transmission duration

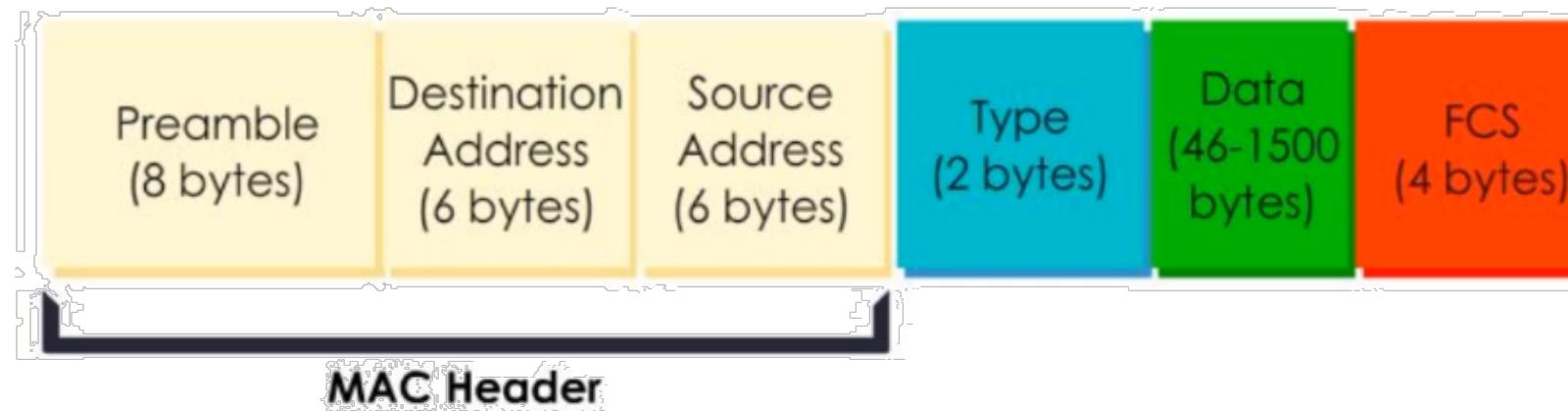
So other devices know when the channel will be available again

Transmitter /
AP MAC
(BSSID)

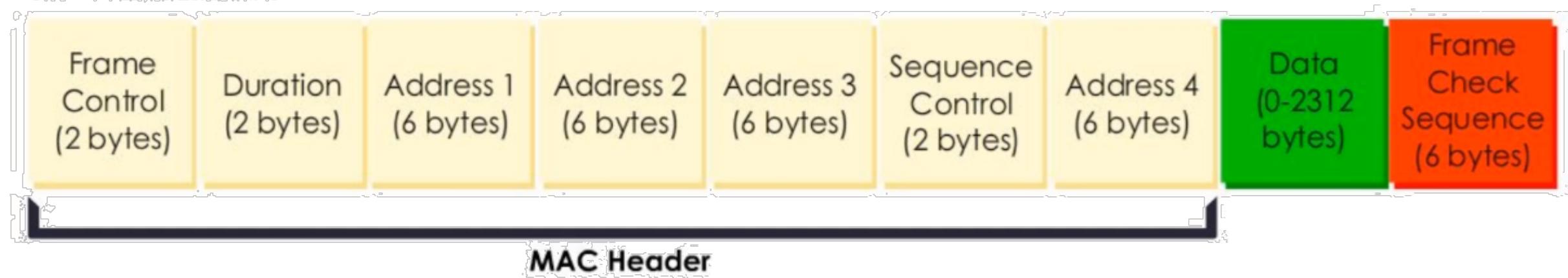
How a large packet
is fragmented



MAC Header

Ethernet Data Frame:

Ethernet vs Wi-Fi Data Frame

Wi-Fi Data Frame:

Wi-Fi (large header) traffic is not as efficient as Ethernet

Latest 802.11ac Wi-Fi uses **frame aggregation** to reduce overhead

5G vs WiFi6





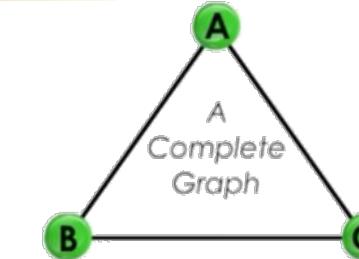
STP: Spanning Tree Protocol

- Fundamentals
- Protocol Brief

Complete Graph vs Spanning Tree

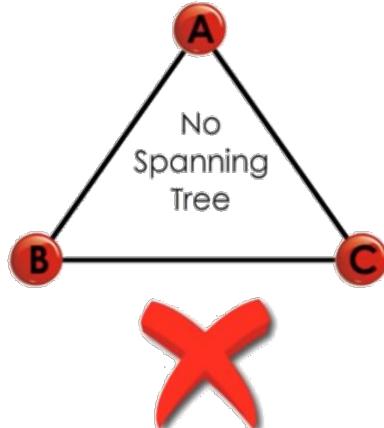
■ Complete Graph:

Each pair of vertices connected by a line
i.e. Fully Meshed Network

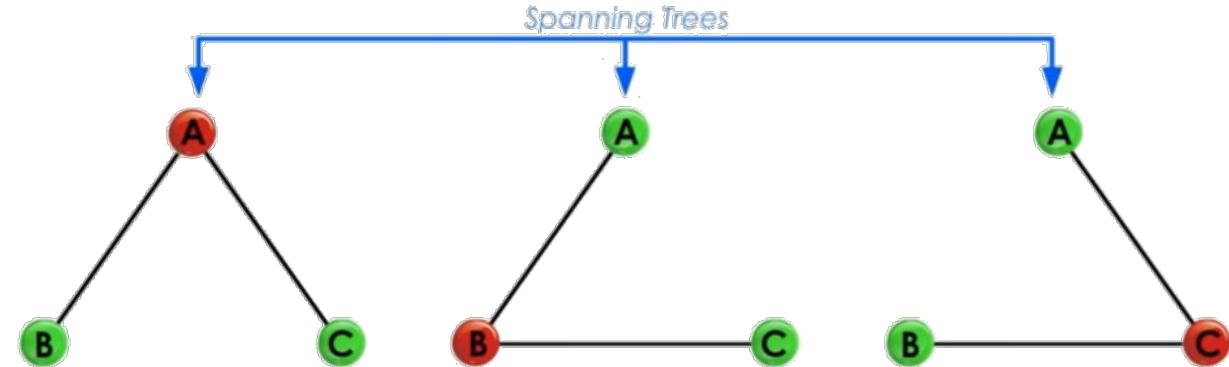


■ Spanning Tree:

Each pair of vertices connected by a line



1) A spanning tree has no loop

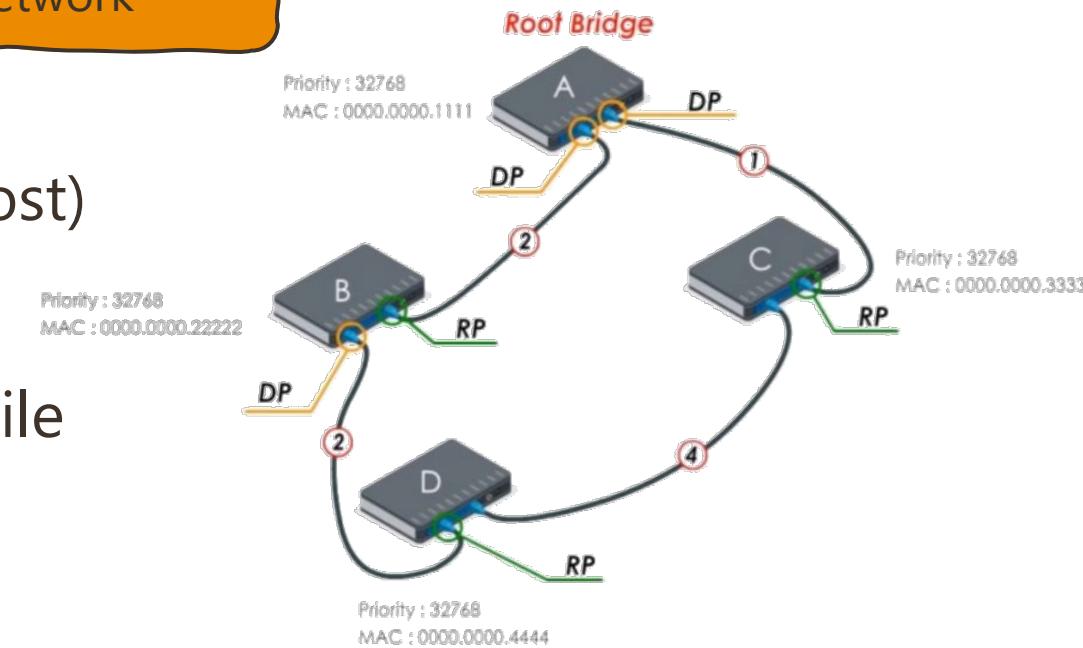


Spanning Tree Protocol (STP)

- Layer 2 protocol runs on bridges and switches
- Used to build a loop-free logical topology

▪ Steps

- 1 Select one switch as **root bridge**
 - 2 It chooses the **shortest path** (the least cost) **from a switch to root bridge**
 - 3 It **blocks** links that could cause **loops** while maintaining these links as backups (fault tolerance)
- is the central point on the network





VLAN

- Fundamentals
- VLAN Types
- IEEE802.11q Standard
 - Access Ports
 - VLAN Tag
- Default VLAN
- Native VLAN

Virtual LAN (VLAN)

- VLAN is a logical network that group devices/users regardless of their different physical locations
- **VLAN is created at the switch**

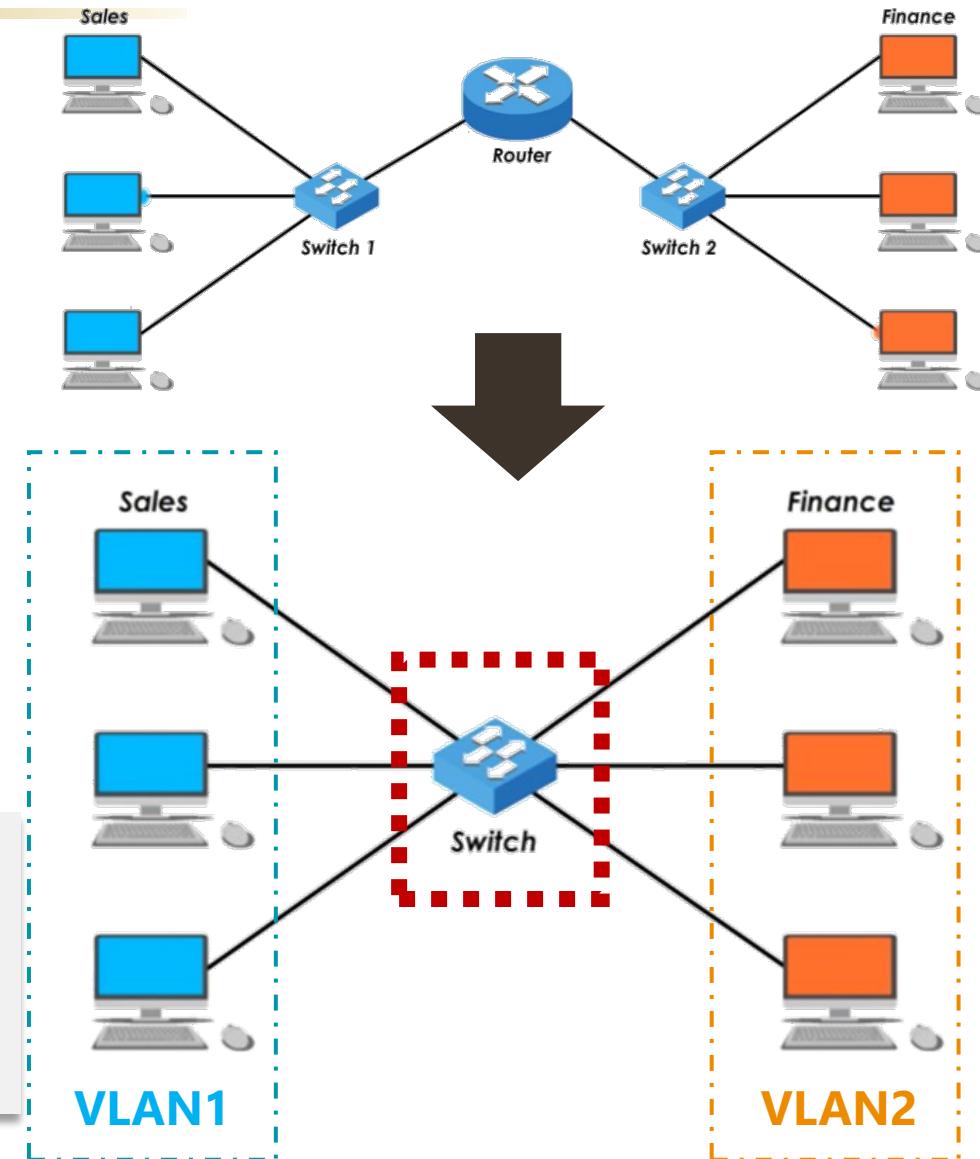
▪ Why VLAN

- Network Segmentation
- Security
- Traffic Prioritizing
- Network Management

Network Design, Deployment,
Troubleshooting without
affecting other VLANs

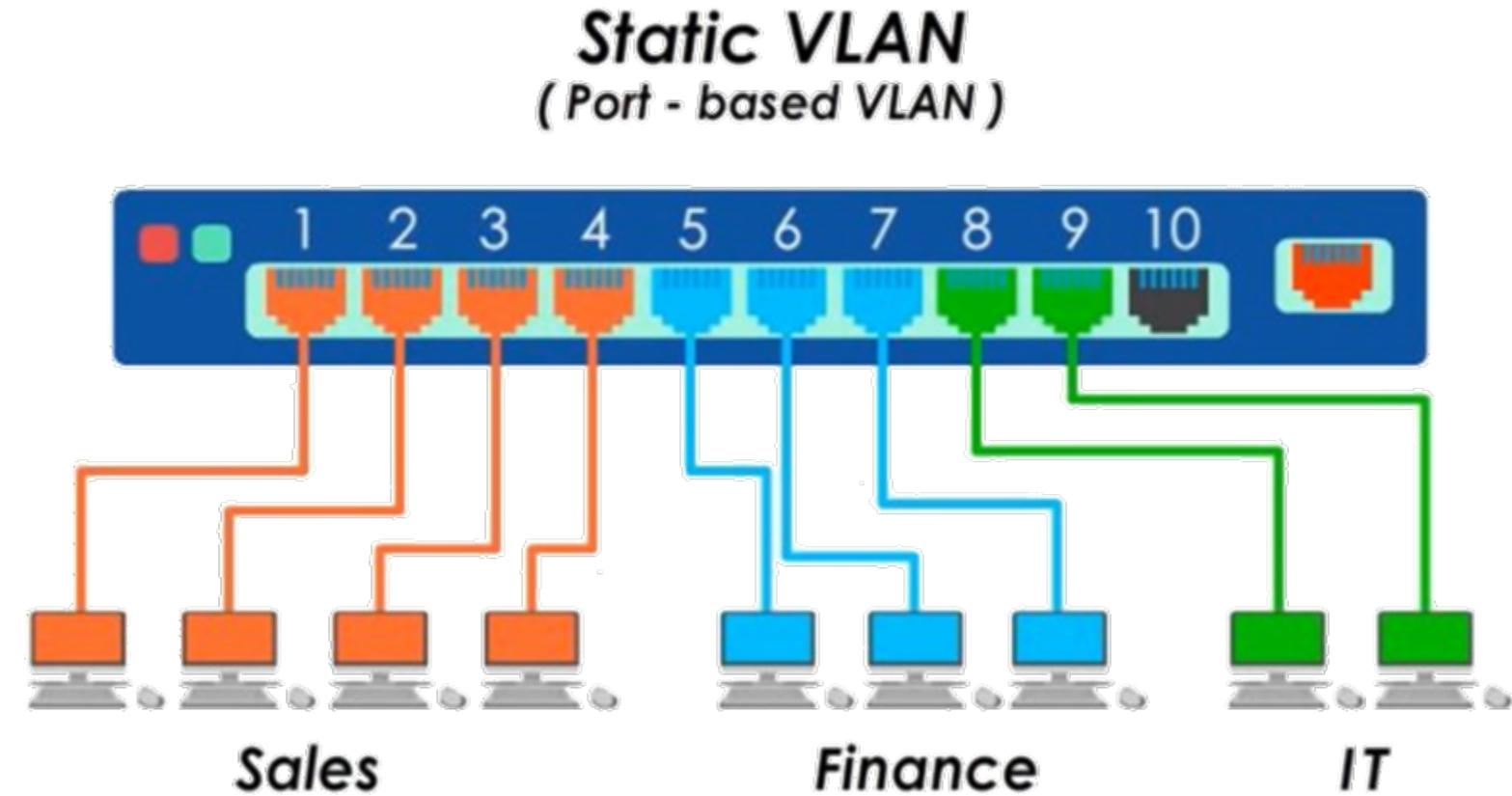
▪ VLAN Types:

1. **Static VLAN**
2. **Dynamic VLAN**



Type: Static VLAN

- Port based
- Manual Assignment
of individual ports on
a switch to a VLAN



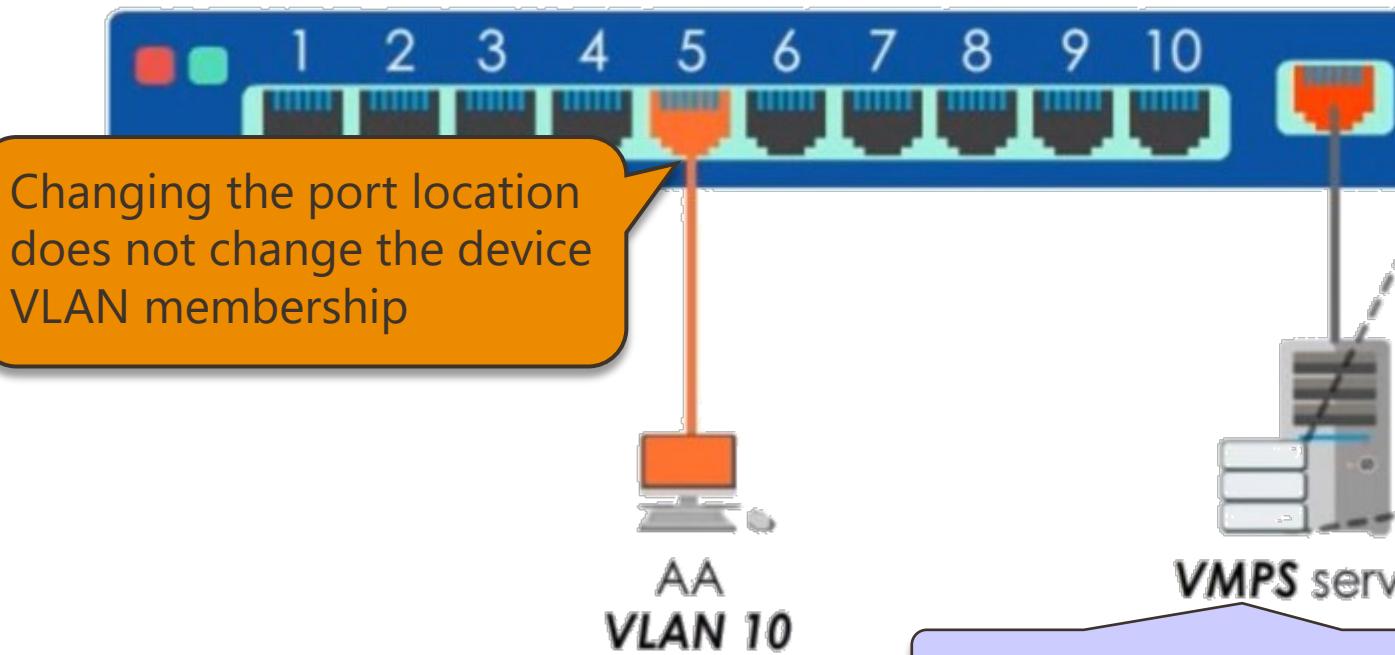
Sales VLAN cannot communicate with Finance or IT VLAN vice versa

Type: Dynamic VLAN

- MAC based or IP based
- Defined based on a device instead of a port location

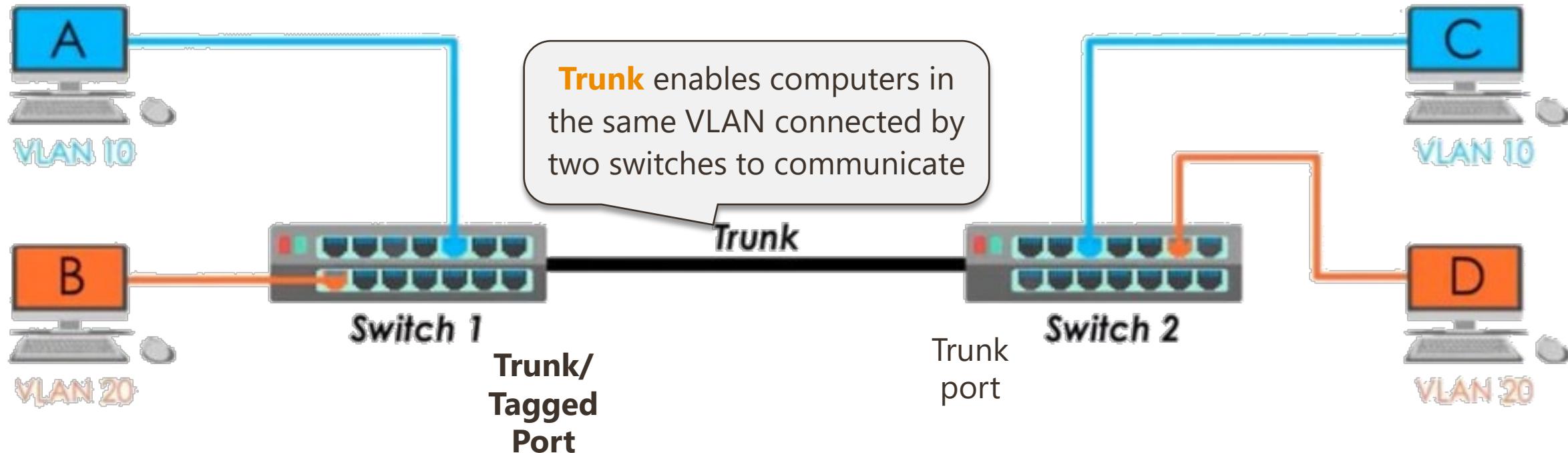
+VE
Flexibility
Security

Dynamic VLAN
(MAC based)



IEEE802.1Q Standard

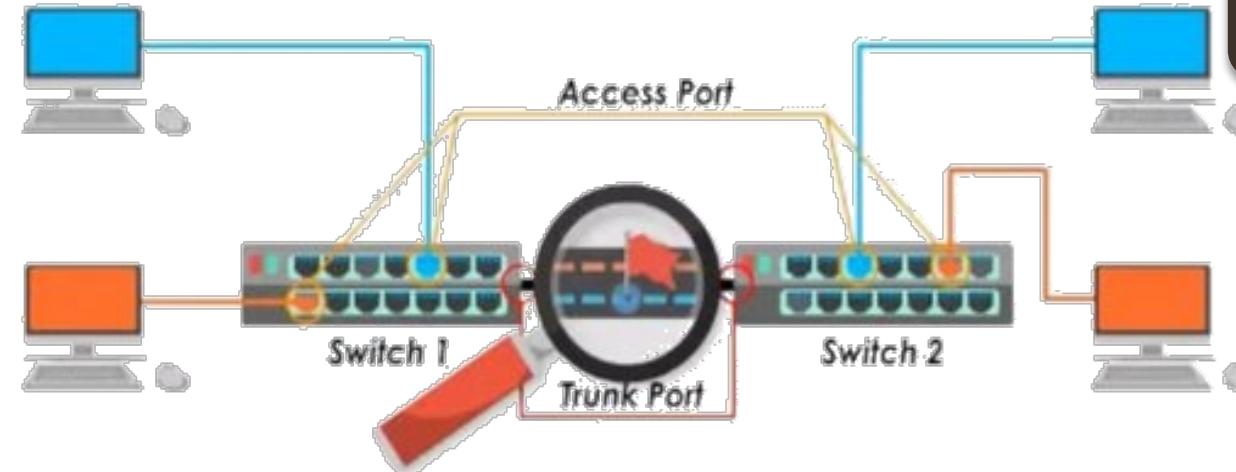
- supports VLANs on the Ethernet network
- defines a **method of tagging traffic between two switches or switch-router to tell which traffic belongs to which VLAN**



Ports

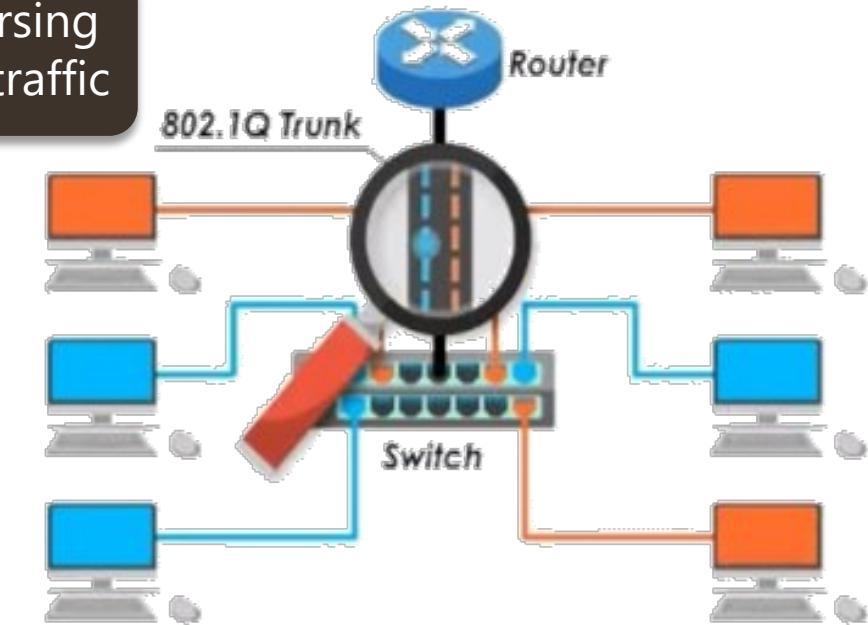
- **Trunk/Tagged Port:** Send and expect traffic with VLAN tag
- **Access Port:** Send and expect traffic with no VLAN tag

Trunking between two switches

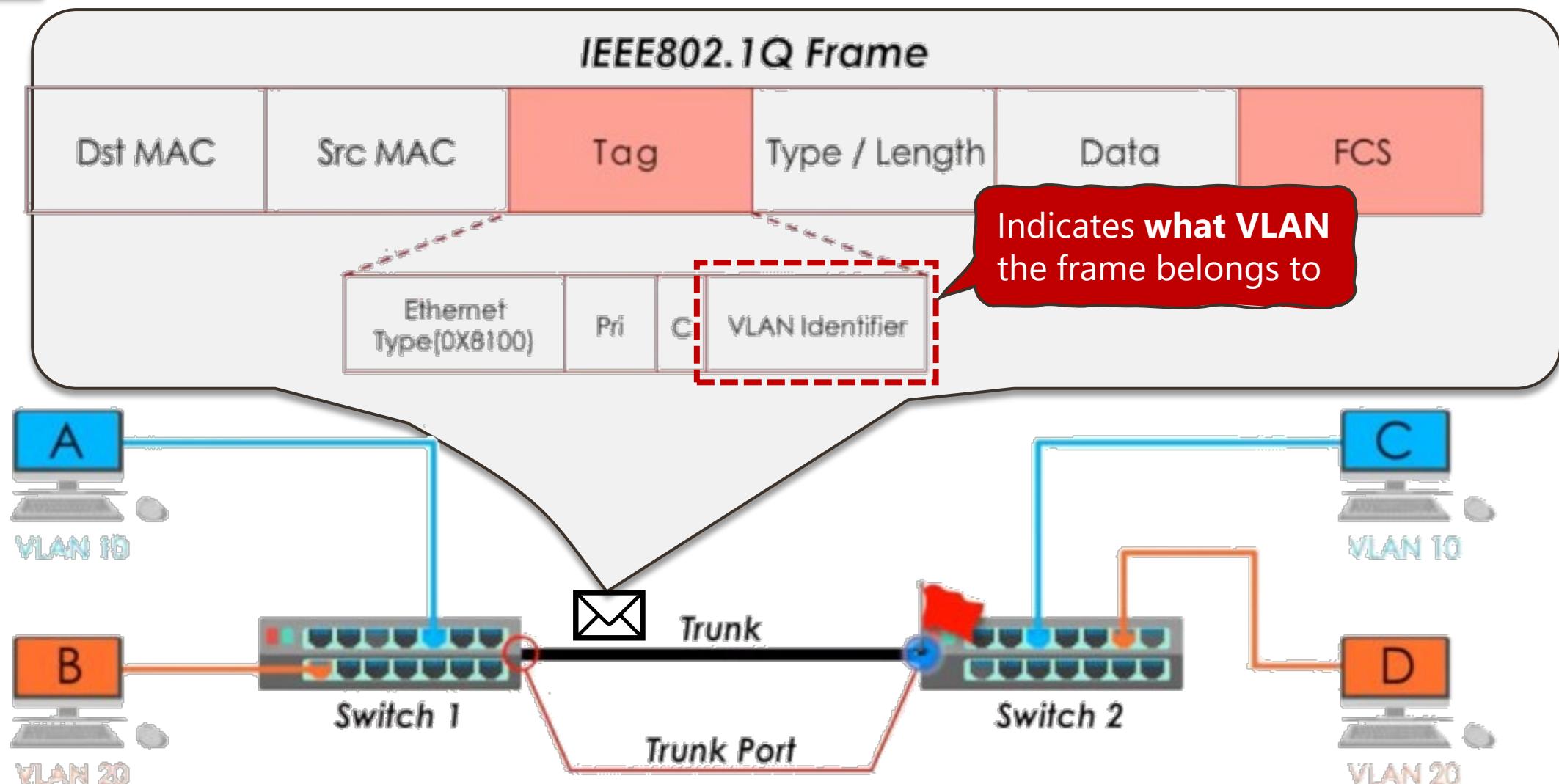


Process of traversing
different VLAN traffic

Trunking between a switch and a router



VLAN Tag



Default VLAN: VLAN 1

- **Cannot change, or delete** default VLAN
- Not intended to be used as a standard data VLAN

Switch1#show vlan brief				
VLAN Name	Status	Ports		
1 default	active	Fa0/1, Fa0/2, Fa0/3 Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2		
10 Engineering	active			
20 Finance	active			
30 Management	active			
40 Marketing	active			
50 Sales	active			
1002 rcd1-default	act/unsup			
1003 token-ring-default	act/unsup			
1004 fddinet-default	act/unsup			
1005 trnet-default	act/unsup			

Default setting
on cisco switches
for all ports

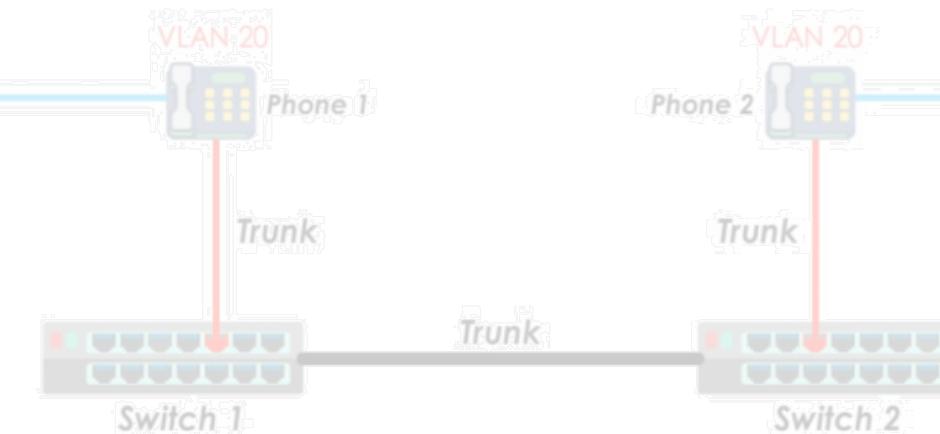
Native VLAN

Devices that do not support VLANs

1. Provides **backward compatibility** for **old devices** that do not speak 802.1q standard

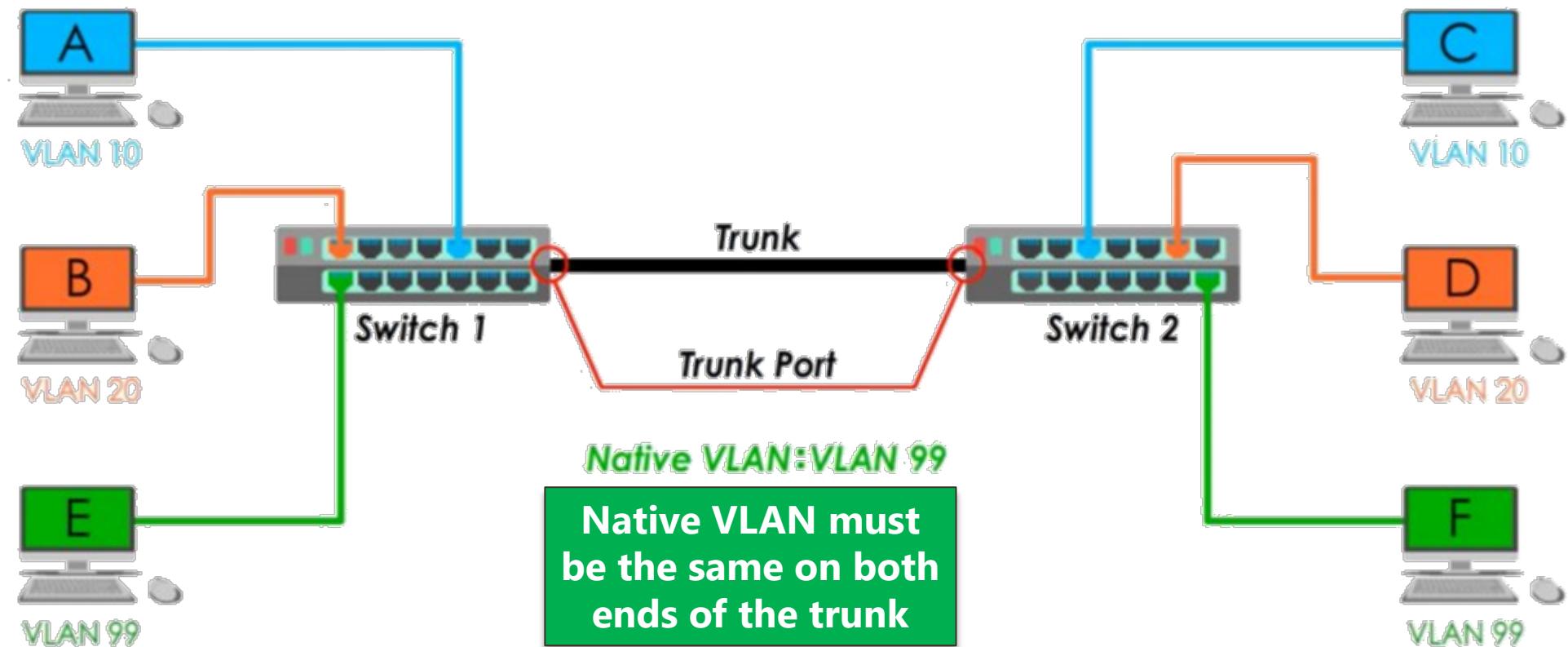
2. Used by the switch to **carry** specific **control and management protocol traffic**
 - i.e. Cisco Discovery Protocol (CDP),
 - VLAN Trunking protocol (VTP),
 - Spanning Tree Protocol (STP)

3. **Useful in VOIP** (Voice Over IP)



Native VLAN

Native VLAN is a special VLAN whose **traffic** traverses **on 802.1q trunk without the VLAN tag**



By default,

Native VLAN = Default VLAN = **VLAN 1**



■ Ethernet

- LAN, Ethernet Fundamentals
- History and Evolution
- Ethernet, Fast Ethernet, Gigabit Ethernet and Cabling
- IEEE802.1 Ethernet DLL
 - LLC: Services
 - MAC: Data Frame
 - MAC: Protocol
- Deployment
- Adaptive Learning

■ Collision vs Broadcast Domain

- Fundamentals
- Hub, Bridge, Switch, Router

■ Wi-Fi

- Fundamentals
- Modes (adhoc, infrastructure)
- Terms (BSS, BSSID, SSID, ESS)
- Scanning Methods (active, passive) Applications
- IEEE802.11 Standards (b/a/g/n/ac/ax, Wi-Fi 6)
- MIMO (SU-MIMO, MU-MIMO)
- CSMA/CA
- 2.4 Ghz, 5Ghz Bands
- OFDM
- Wireless Threats
- Ethernet vs Wi-Fi Technology
- IEEE802.11 Wi-Fi Frames
 - Management Frame
 - Control Frame
 - Data Frame
- Ethernet vs Wi-Fi Data Frame

■ Spanning-Tree Protocol

- Fundamentals
- Protocol Brief

■ VLAN

- Fundamentals
- VLAN Types
- IEEE802.11q Standard
 - Access Ports
 - VLAN Tag
- Default VLAN
- Native VLAN

THANK YOU

Make tomorrow better.



Curtin University

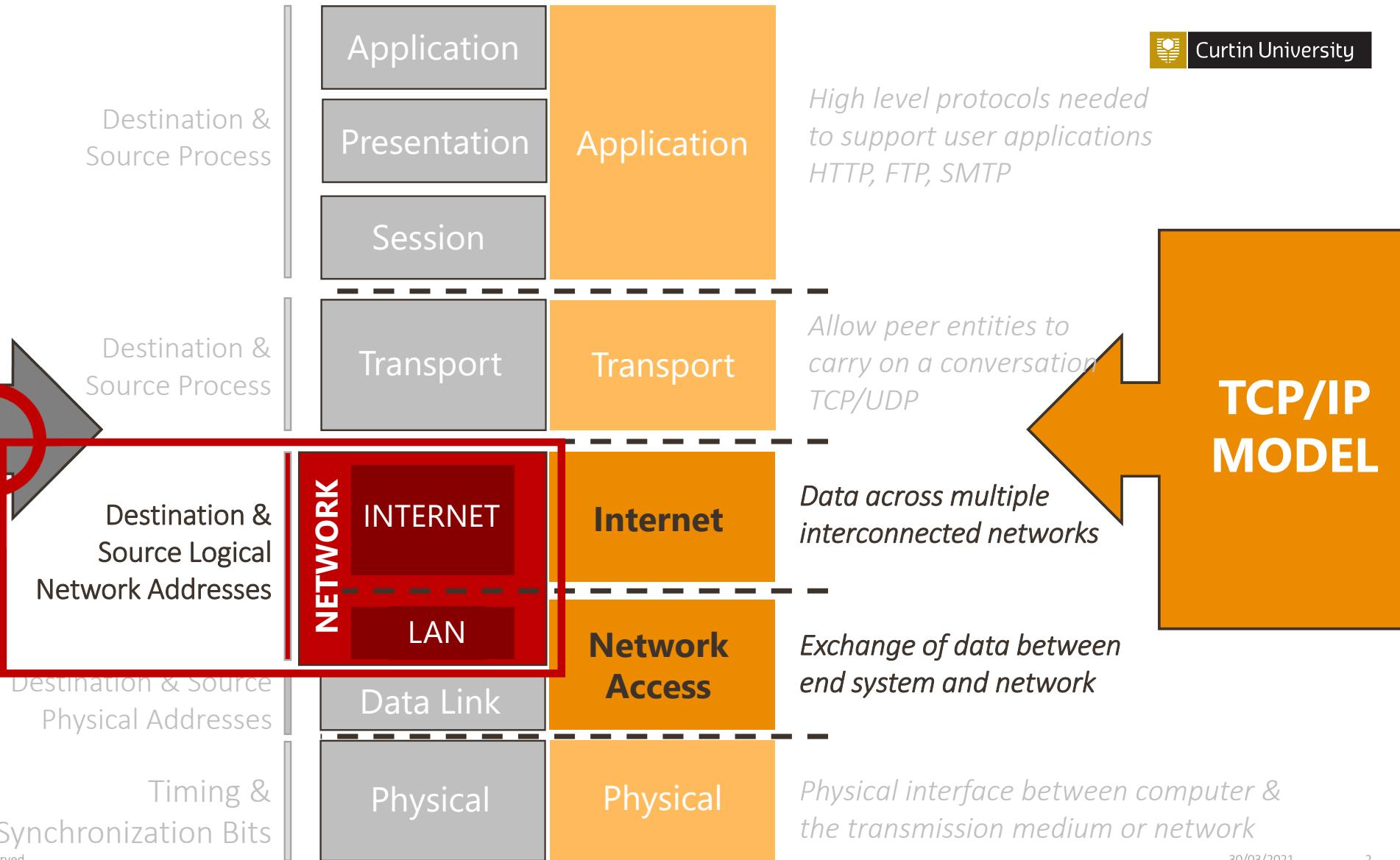
Network Layer I

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 05

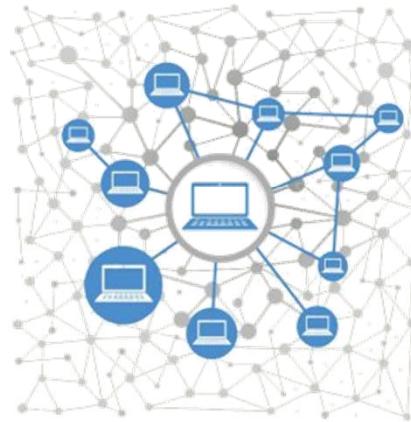
Semester 1, 2021

A GLOBAL UNIVERSITY

WESTERN AUSTRALIA | DUBAI | MALAYSIA | MAURITIUS | SINGAPORE



Network Layer



Provides the means of transferring packets from a **source to a destination** host via one or more networks

Network Layer

- Transport segment from sending to receiving host

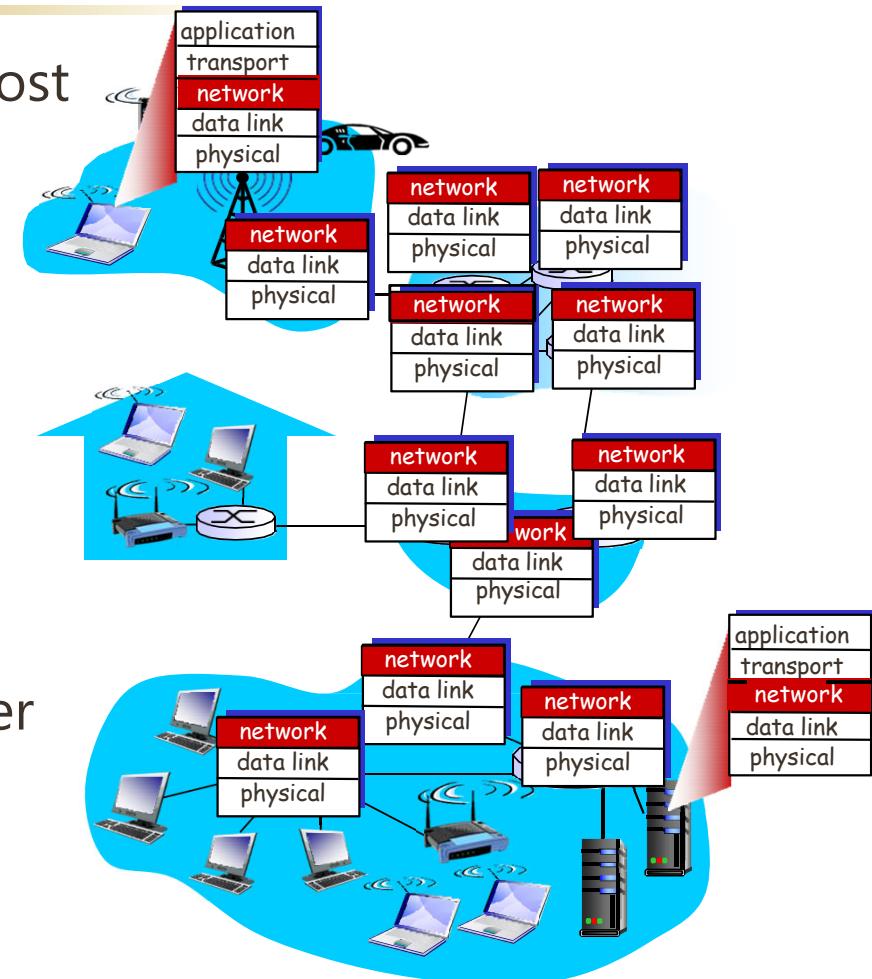
- Sending Side:

- Encapsulates segments into datagrams

- Receiving Side:

- Delivers segments to transport layer

- Network layer protocols in every host and router
(layer-3 devices)



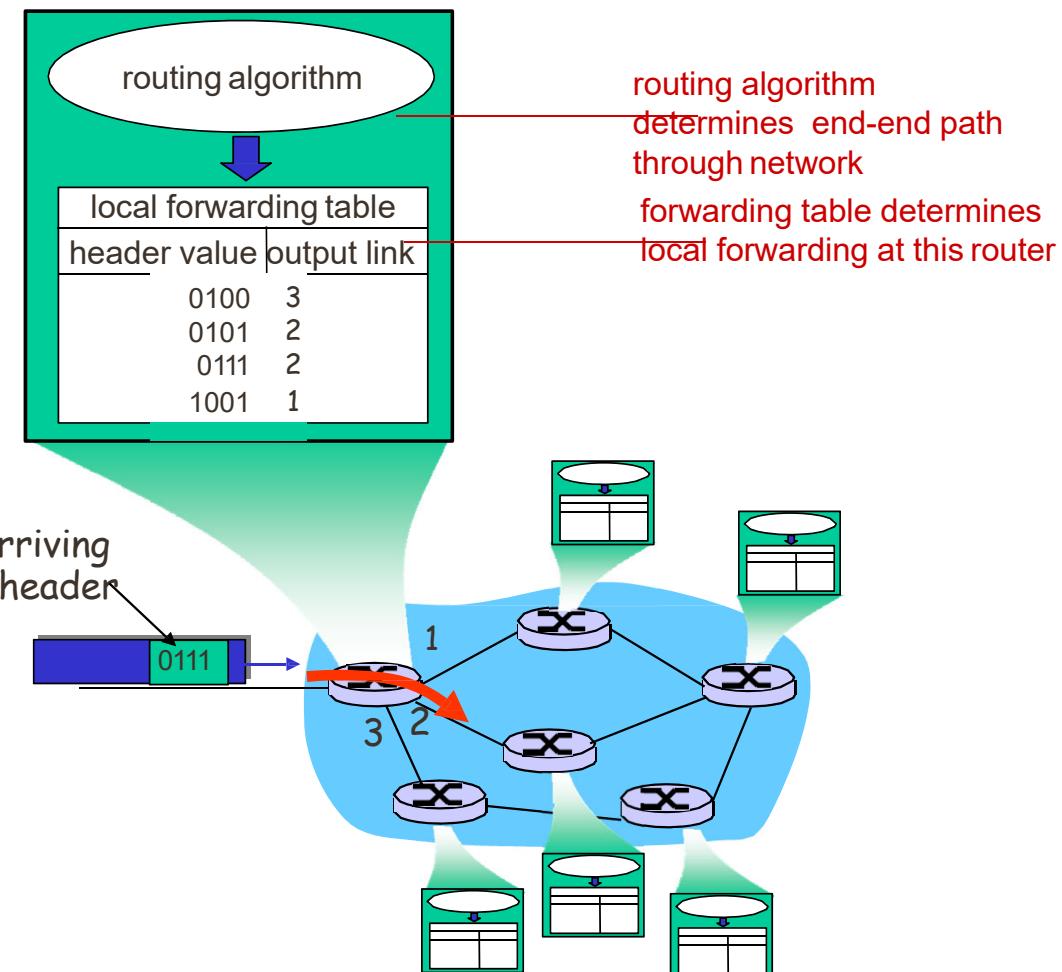
Key Functions

1. Store-and-forward

- Move packets from router's input -> appropriate router output
 - **Analogy:** process of getting through single interchange

2. Routing

- Determine route taken by packets from source to destination
 - **Analogy:** process of planning trip from source to destination (via routing algorithms)



Key Functions – cont.

3. Connection Setup

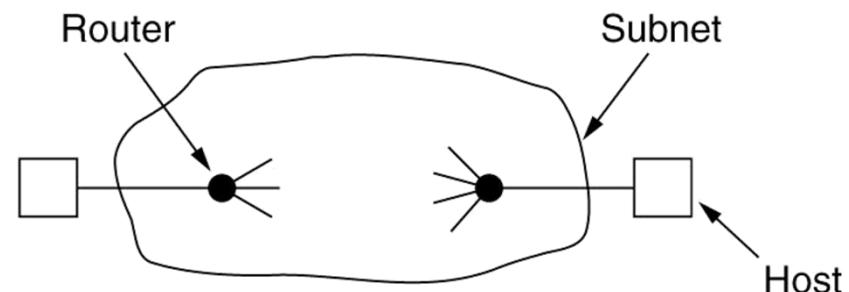
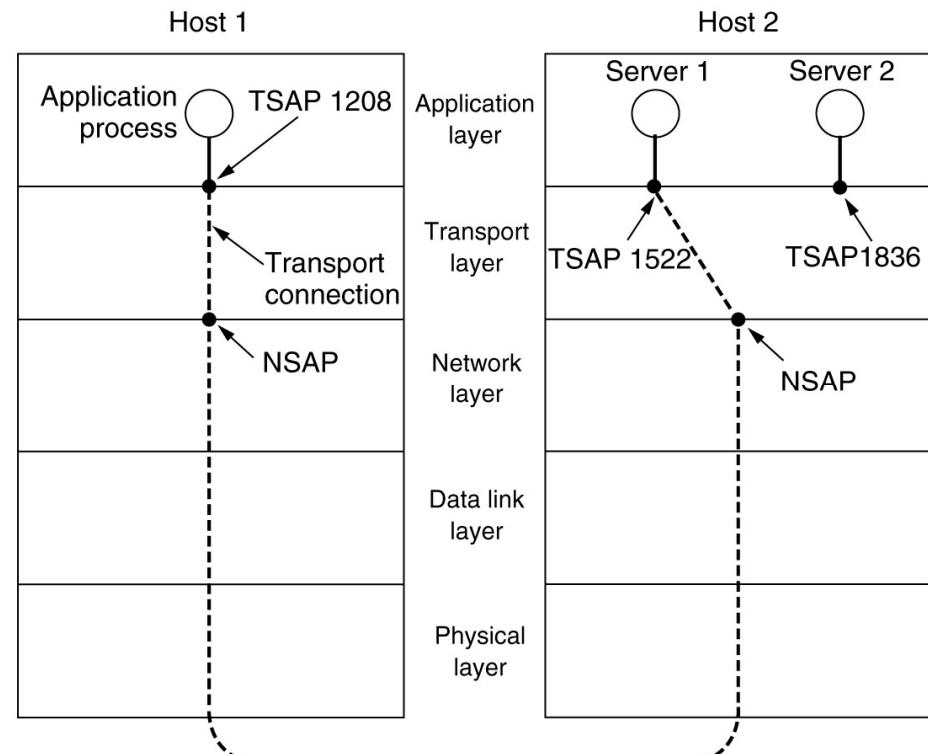
- Only in some network architectures

(i.e. ATM, frame relay, X.25)

- Before datagrams flow: Two end hosts and intervening routers establish *virtual-circuit*

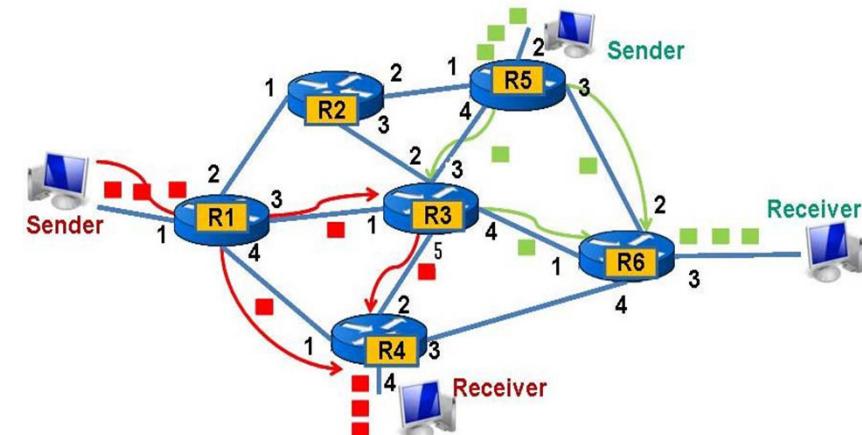
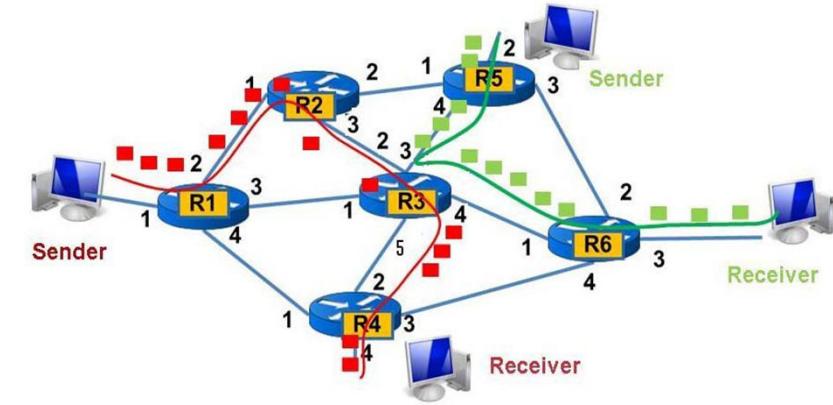
▪ Network vs Transport Layer connection service:

- **Network:** between two hosts (may also involve intervening routers in case of VCs)
- **Transport:** between two processes

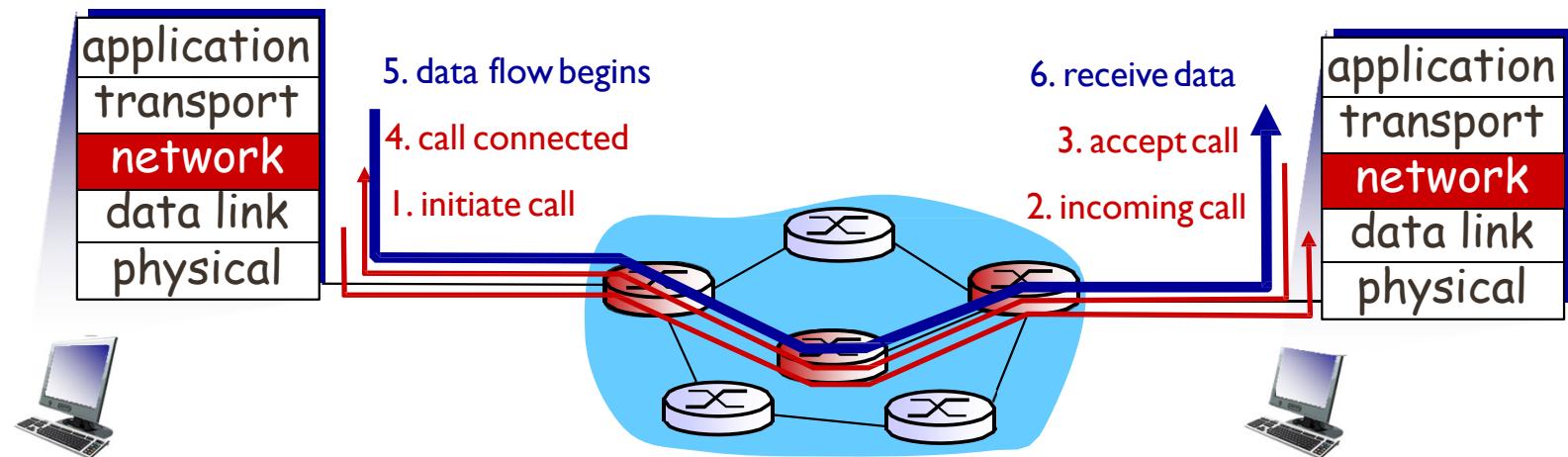


Connection/Connectionless Service

- **Virtual-circuit (PS) Network** provides network-layer **connection service**
- **Datagram (PS) Network** provides network-layer **connectionless service**
- Analogous to **TCP/UDP**, but:
 - **service:** host-to-host
 - **no choice:** network provides one or the other
 - **implementation:** in network core



Virtual Circuit Network

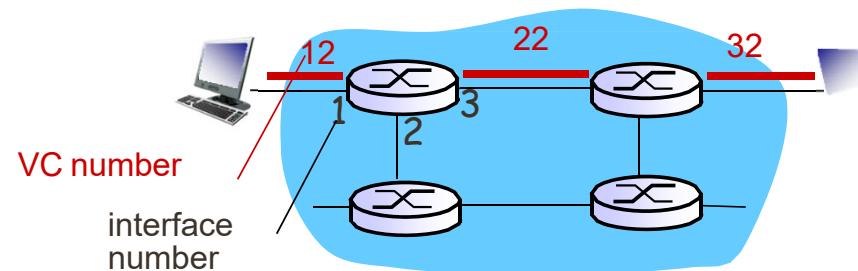


▪ Steps:

- Call setup, teardown for each call before data can flow
- Each packet carries **VC identifier** (*not dest. host address*)
- Every router on source-destination path maintains "state" for each passing connection
- Link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)

"source-to-destination path behaves much like telephone circuit"

VC Forwarding Table



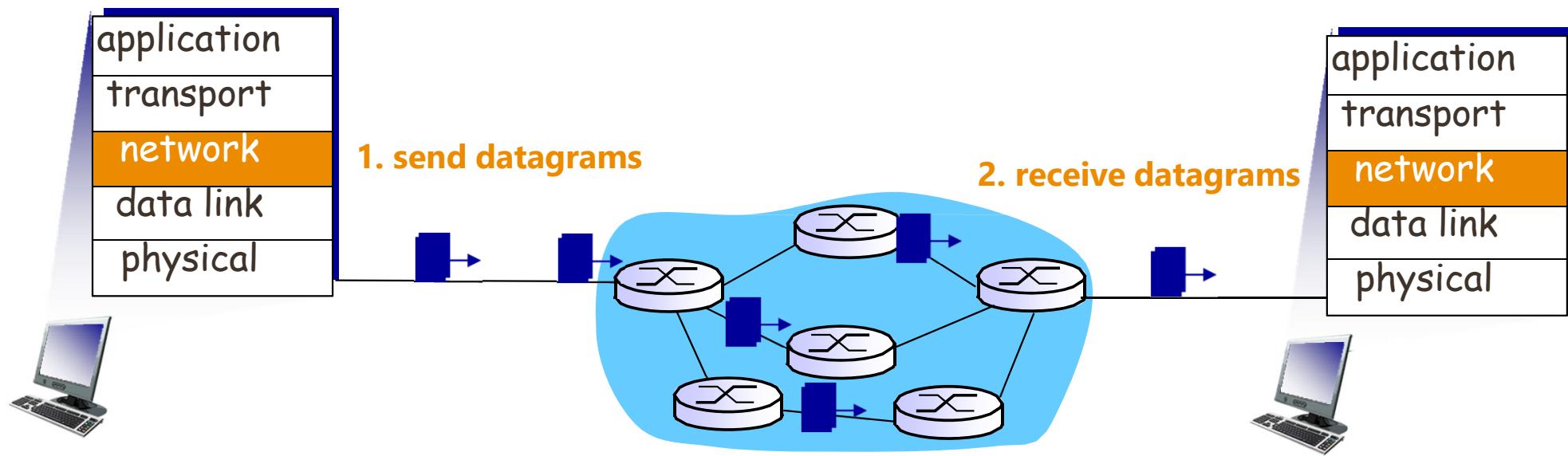
Forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

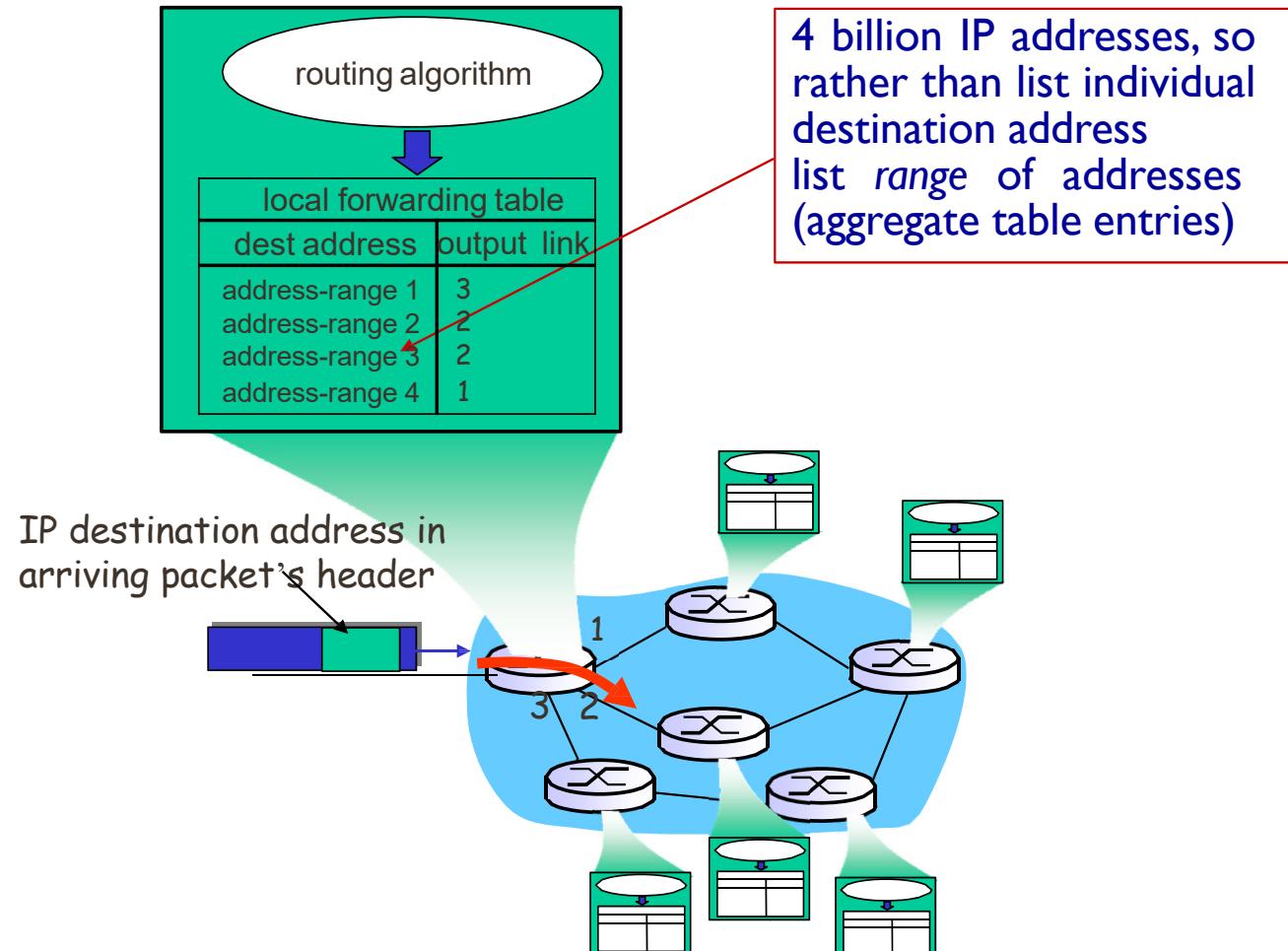
VC routers maintain connection state information!

Datagram Network

- **No call setup** at network layer
- **Routers:** no state about end-to-end connections
- **Packets Forwarding** via destination host address



Datagram Forwarding Table



Datagram Network or VC: why?

▪ Internet (datagram)

- ✓ data exchange among computers
 - “elastic” service, no strict timing req.

▪ many link types

- ✓ different characteristics
- ✓ uniform service difficult

▪ “smart” end systems (computers)

- ✓ can adapt, perform control,
- ✓ error recovery

**Simple inside network,
complexity at “edge”**

▪ ATM (VC)

- ✓ Evolved from telephony
- ✓ Human conversation
 - Strict timing, reliability requirements
 - Need for guaranteed service

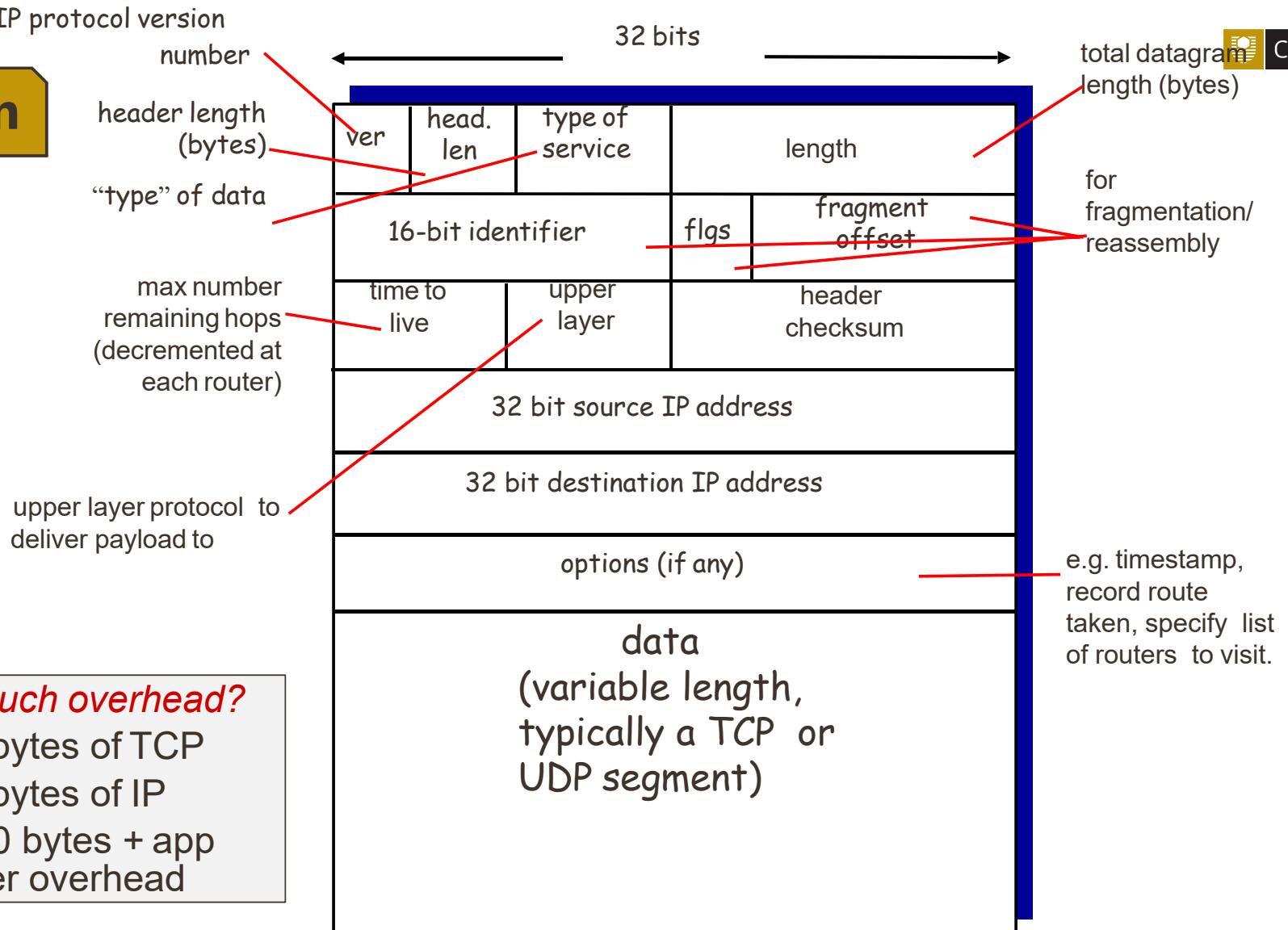
▪ “dumb” end systems (computers)

- ✓ can adapt, perform control,
- ✓ error recovery

Complexity inside network

IP Datagram

Header

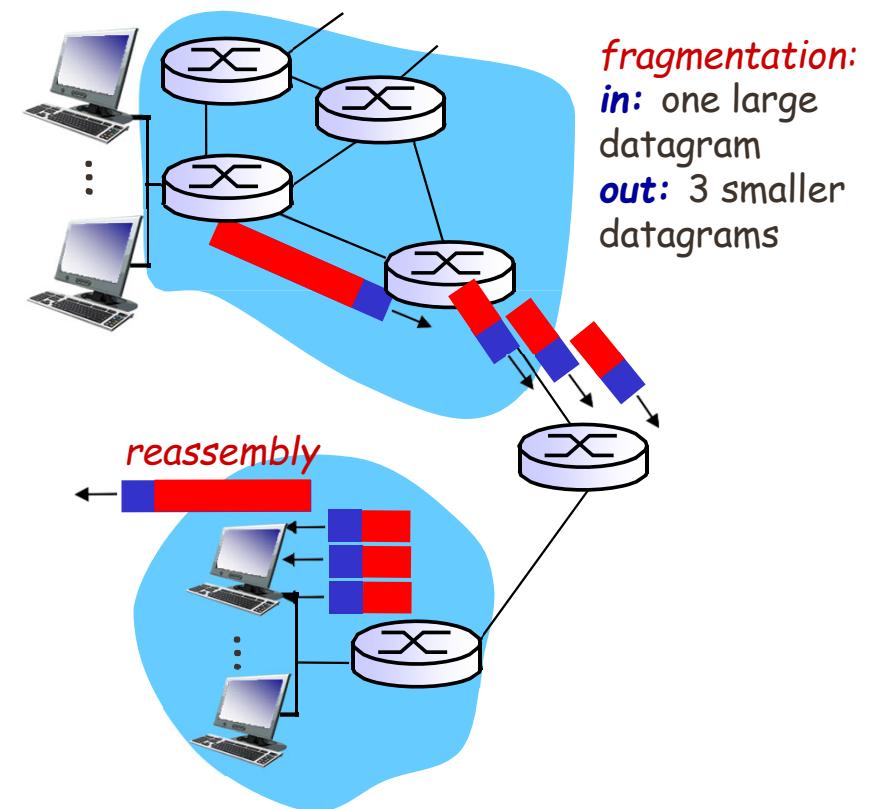


how much overhead?

- ❖ 20 bytes of TCP
- ❖ 20 bytes of IP
- ❖ = 40 bytes + app layer overhead

IP Fragmentation, Reassembly

- Network links have **MTU** (Max Transfer Size)
 - Largest possible link-level frame
 - Different link types, different MTUs
- **Fragmentation:** Large IP datagram divided within network
 - One datagram becomes several datagrams
- **Reassembly:** Only at final destination
 - IP header bits used to identify & order related fragments





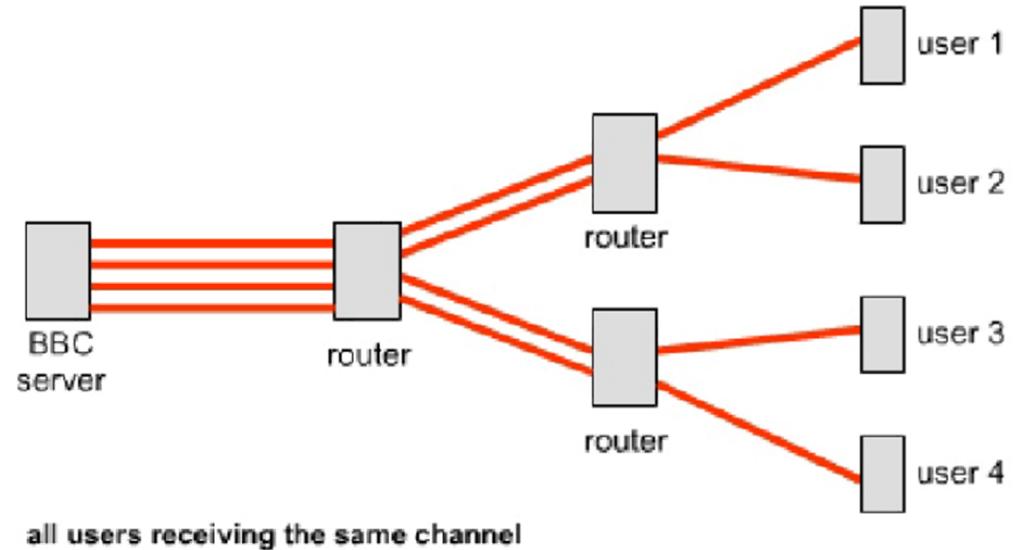
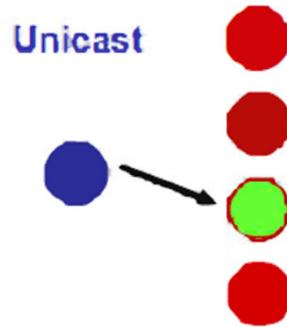
Address Types

- Unicast
- Multicast
- Broadcast
- Anycast
- Geocast

Unicast

- Use IP delivery methods (session-based protocols) such as:

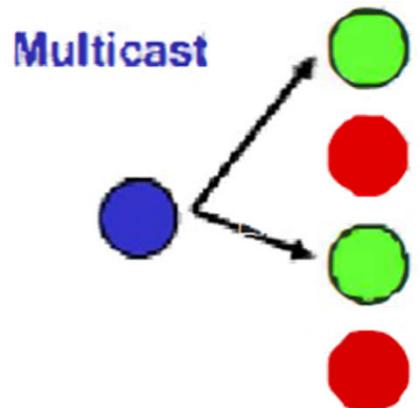
- ✓ Transmission Control Protocol (TCP)
- ✓ User Datagram Protocol (UDP)



- Each unicast client that connects to the server takes up additional bandwidth
- **Client** has a **direct relationship** to the **server**

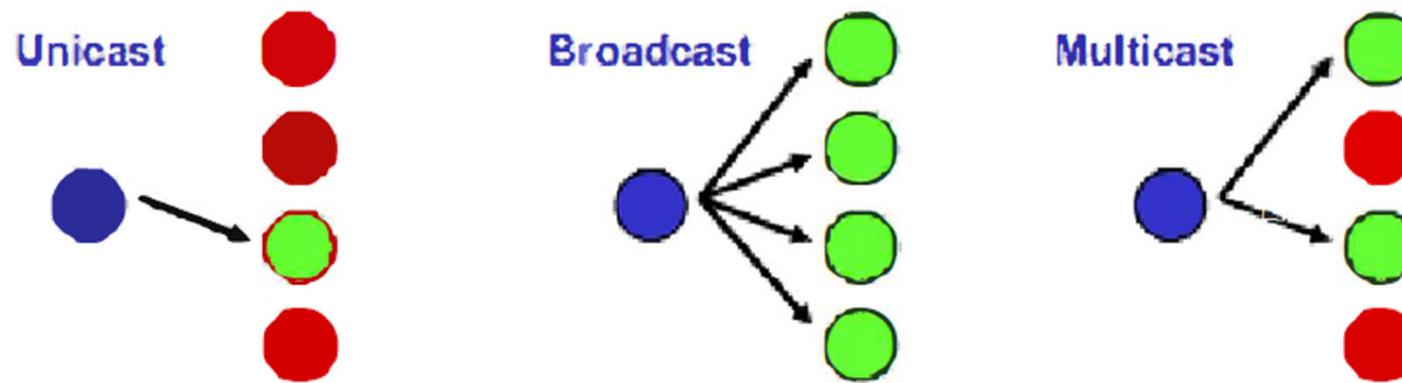
Multicast

- A multicasting is associated with a group of **interested receivers**
- **No direct relationship** between the clients and server
- Clients are connected to the multicast address, **no additional overhead** on the server.
 - i.e. server sends out only one stream per multicast station. same load is experienced on the server whether only one client or 1,000 clients are listening
- Can be used across a **WAN**



Multicast / Broadcast

- **One-to-many** communications



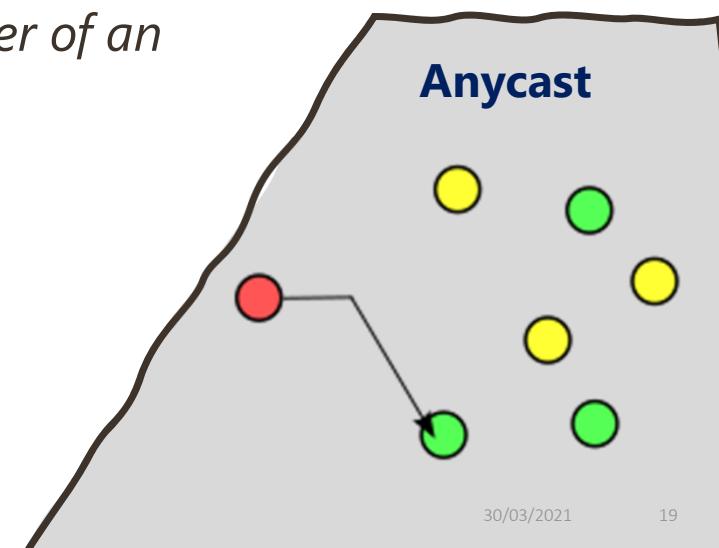
- IP Multicasting refers to implementation of multicast communication in the internet

- **Multicast is driven by receivers:**

- Receivers indicate interest in receiving data

Anycast [RFC: 1546]

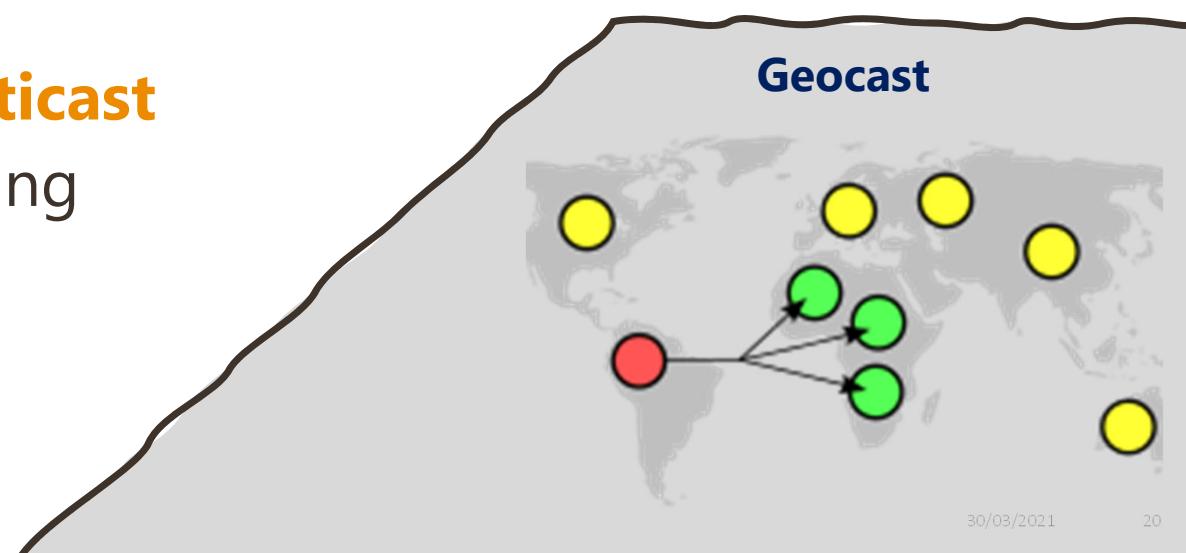
- Anycast addressing is a **one-to-one-of-many** association where datagrams are routed to any single member of a group of potential receivers **that are all identified by the same destination address**
- The routing algorithm selects the single receiver from the group **based on least-expensive routing metric**
 - *i.e. the packets are routed to the topologically-nearest member of an anycast group*
- **Widely used** for Content Delivery Network (**CDN**) products to bring their content closer to the end user



Geocast

- Geocast refers to the delivery of information to a group of **destinations in a network identified by their geographical locations**

- It is a **specialized form of multicast addressing** used by some routing protocols for mobile ad-hoc networks.





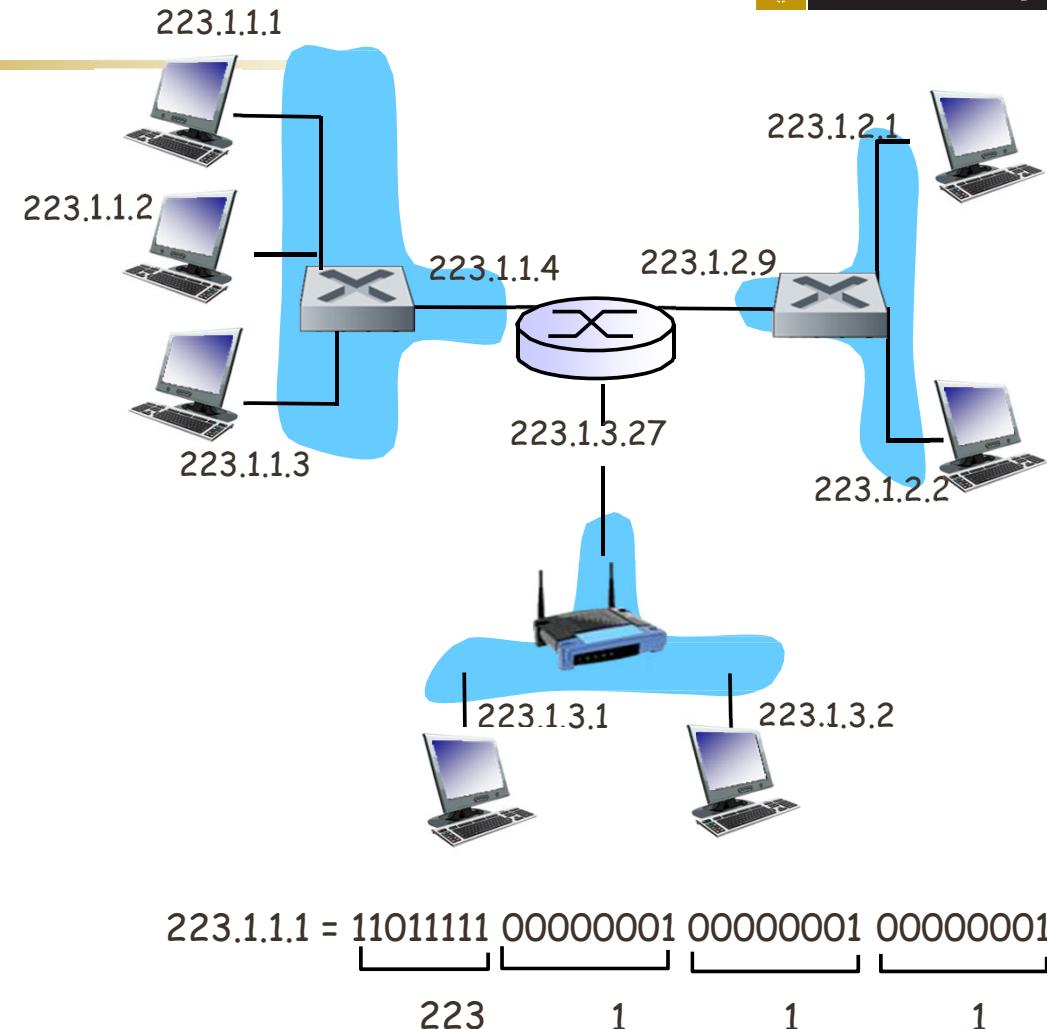
IPv4 Addressing

- Fundamentals
- Subnets
- Classful, Classless (CIDR) IP
- Static vs Dynamic IP
- Obtaining a Global IP Address
 - ISP Address Allocation
 - Hierarchical Addressing
 - ICANN
- Special IPv4 Addresses

IP Addressing

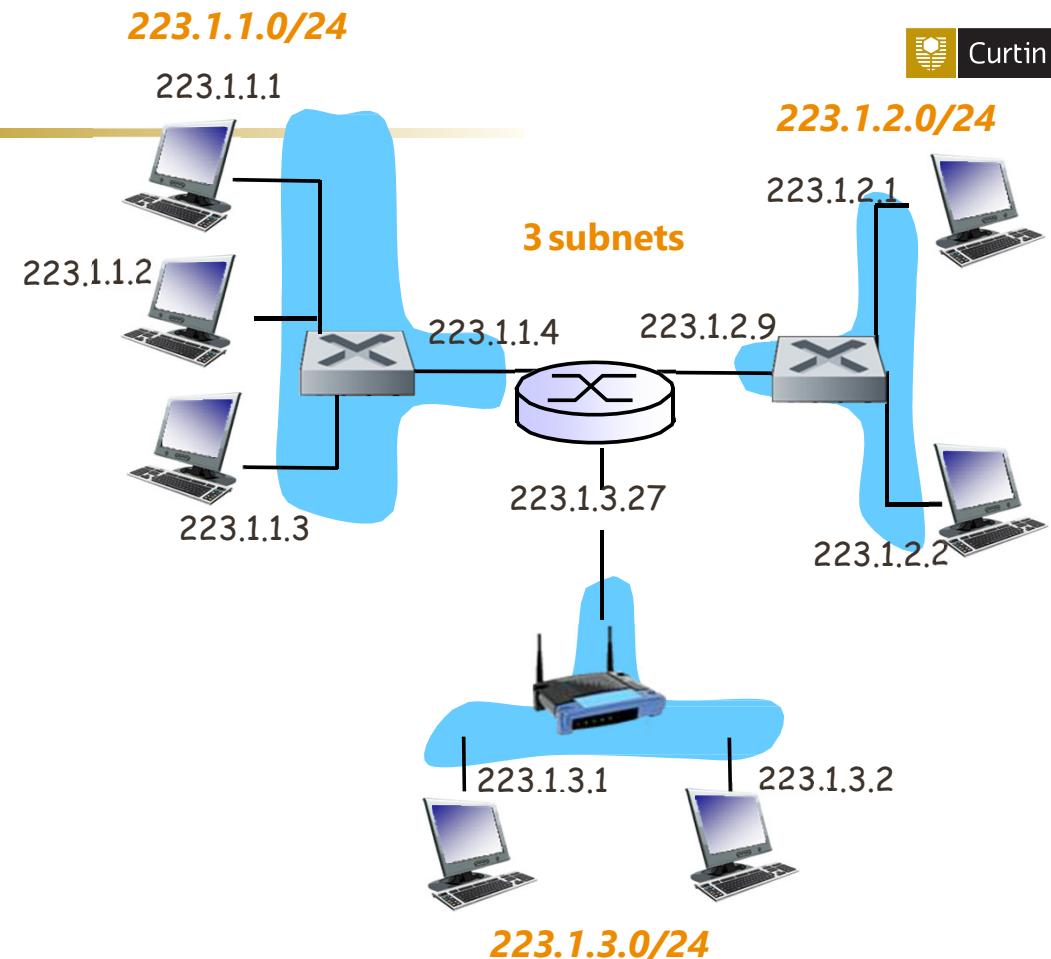
- **IP address:** 32-bit identifier for host and router interface
- **Interface:** connection between host/router and physical link
 - **Routers:** multiple interfaces
 - **Host:** one or two interfaces
(e.g., wired Ethernet, wireless 802.11)

"IP addresses for each interface"



Subnets

- A **subnetwork** or **subnet** is a logical subdivision of an IP network
- Isolated networks with **Subnet ID**
 - Can physically reach each other in same sub-network without intervening router
- **IP address:**
 - **Subnet ID** - high order bits
 - Host ID - low order bits



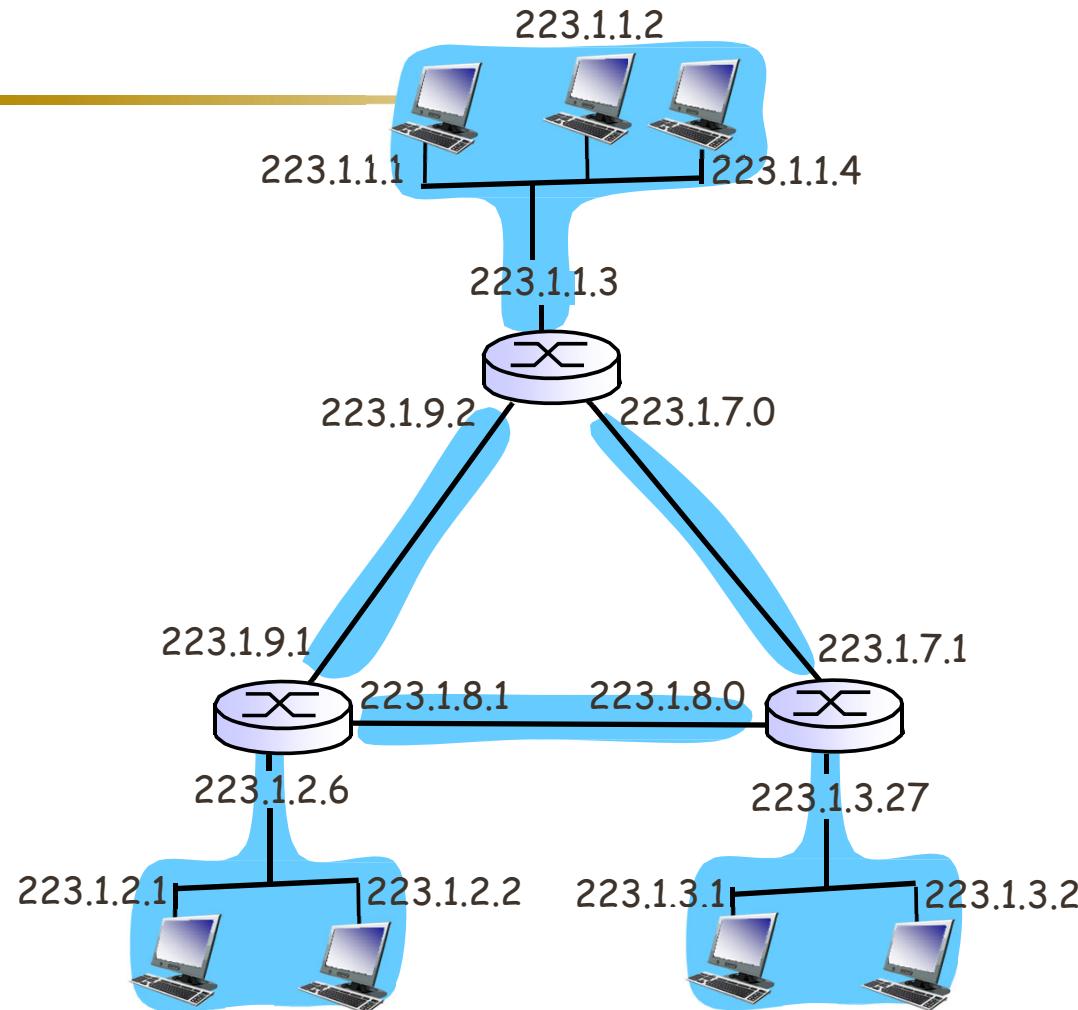
$223.1.1.1/24 = \underline{11011111} \underline{00000001} \underline{00000001} \underline{00000001}$

223 1 1 1

Subnet Mask: /24

Subnets

- How many hosts?
- Each subnet has **equal number of IPv4 addresses**



IP Address Types

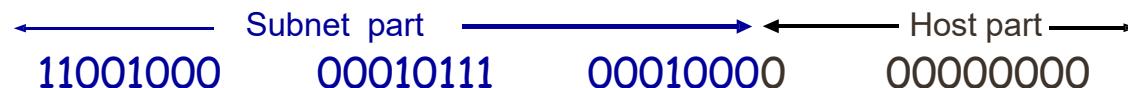
- Classful

Class	Start bits	No. of netwk bits	No. of host bits	Range	
A	0	8	24	1.0.0.1 to 126.255.255.254	Supports 16 million hosts on each of 127 networks.
B	10	16	16	128.1.0.1 to 191.255.255.254	Supports 65,000 hosts on each of 16,000 networks.
C	110	24	8	192.0.1.1 to 223.255.254.254	Supports 254 hosts on each of 2 million networks.
D	1110	undef	undef	224.0.0.0 to 239.255.255.255	Reserved for multicast groups.
E	1111	undef	undef	240.0.0.0 to 254.255.255.254	Reserved for future use, or research and development purposes.

IP Address Types

▪ CIDR: Classless InterDomain Routing

- Subnet portion of address of arbitrary length
- Address format: $a.b.c.d/x$, where x is # bits in subnet portion of address
- Possible to split host part further to create additional subnetting



200.23.16.0/23
Subnet mask is /23 or
255.255.254.0

Subnet Mask example

▪ IP address 192.168.40.28/21 - what's the network ID?

- Subnet Mask /21: 255.255.248.0
- 40 = 00101000 with mask 11111000
- Network ID is 192.168.40.0 /21

▪ 192.168.45.52 /21 - what's the network ID?

- Subnet Mask /21: 255.255.248.0
- 45 = 00101101 with mask 11111000
- Network ID is 192.168.40.0 /21

Variable Length Subnet Masking (**VLSM**)

- Use more than 1 subnet masks in the same network to create subnets with **unequal number of IPv4 addresses (hosts)**.
- **E.g. Divide 192.168.10.0 (a Class C network) into**
 - Subnet 1 : 126 IPv4 Addresses (7 bits required)
 - Subnet 2 : 62 IPv4 Addresses (6 bits required)
 - Subnet 3 : 30 IPv4 Addresses (5 bits required)
 - Subnet 4 : 30 IPv4 Addresses (5 bits required)
- Hence:
 - 192.168.10.0 /25 (255.255.255.128): 126 (128-2) usable IPv4 addresses
 - 192.168.10.128 /26 (255.255.255.192): 62 (64-2) usable IPv4 addresses
 - 192.168.10.192 /27 (255.255.255.224): 30 (32-2) usable IPv4 addresses
 - 192.168.10.224 / 27 (255.255.255.224): 30 (32-2) usable IPv4 addresses

Static vs Dynamic IP Addresses

▪ **Statically Assigned**

- Hard-coded by system admin in a file
 - **Windows:** control-panel->network->configuration->tcp/ip->properties
 - **UNIX:** /etc/rc.config

▪ **Dynamically Assigned**

- **DHCP:** Dynamic Host Configuration Protocol
- Dynamically get address from a server (or router)
 - Temporarily assigned, or "leased"
 - After a period of time, this lease "expires,"
 - Renews your old address or assigns new IP address

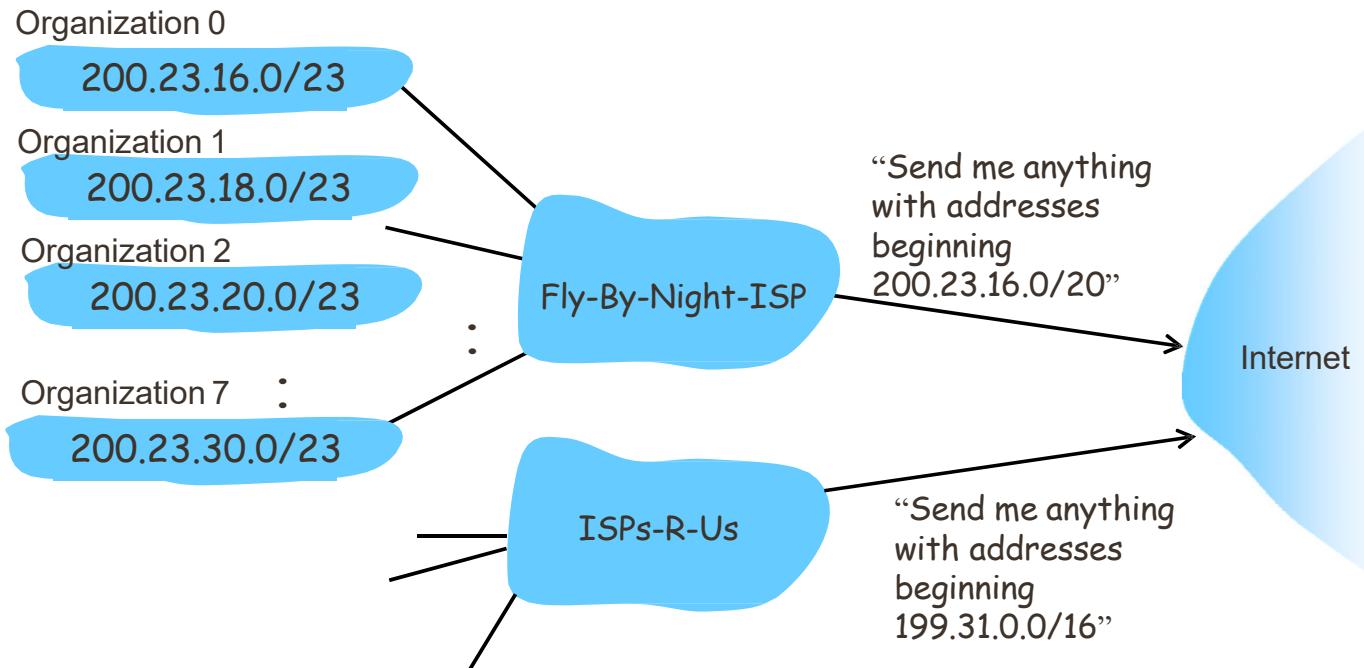
ISP, IP Address Allocation

- ISP (Internet Service Provider), IP address space division

ISP's block	<u>11001000 00010111 00010000 00000000</u>	200.23.16.0/20
Organization 0	<u>11001000 00010111 00010000 00000000</u>	200.23.16.0/23
Organization 1	<u>11001000 00010111 00010010 00000000</u>	200.23.18.0/23
Organization 2	<u>11001000 00010111 00010100 00000000</u>	200.23.20.0/23
...
Organization 7	<u>11001000 00010111 00011110 00000000</u>	200.23.30.0/23

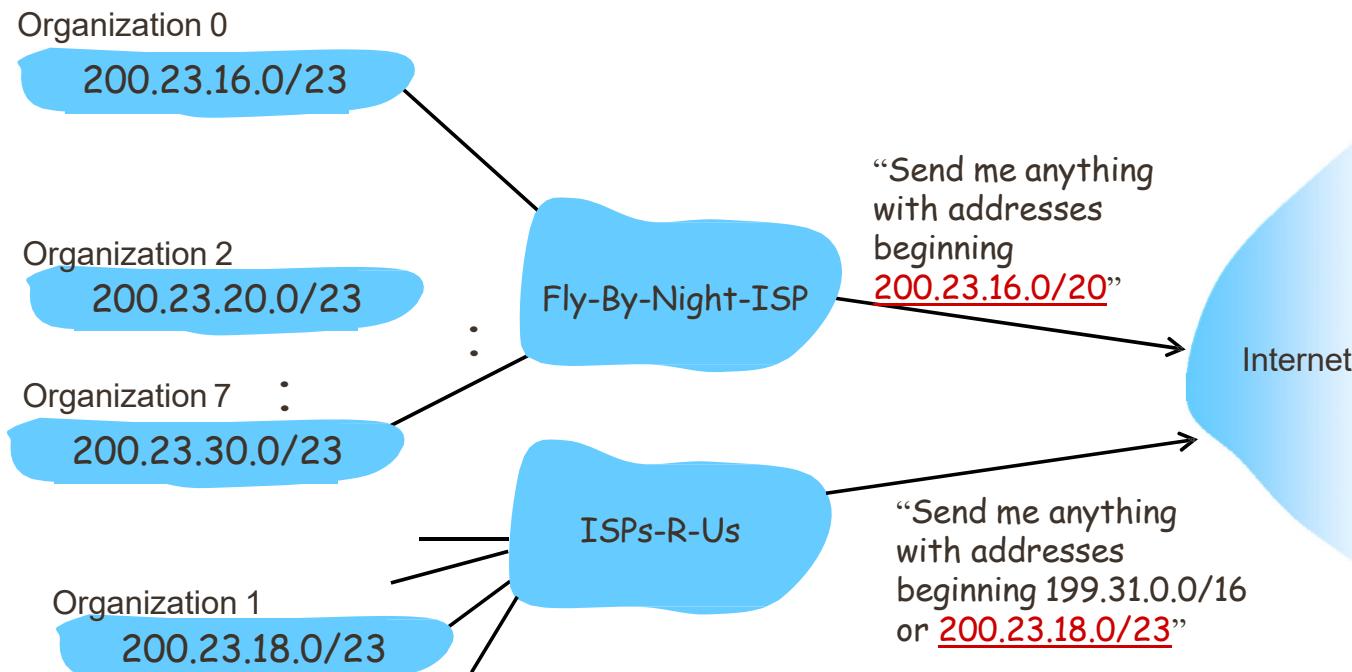
Hierarchical Addressing

- Allows efficient advertisement of routing information:



Hierarchical Addressing – cont.

- ISPs-R-Us has a more **specific route** to Organization 1



ICANN

- How does an ISP get block of addresses?
- **ICANN:** Internet Corporation for Assigned Names and Numbers
<http://www.icann.org/>
 - Allocates addresses
 - Manages DNS
 - Assigns domain names, resolves disputes

Why IP and MAC both?

■ MAC Address Only

- MAC/Physical address is a **globally unique ID** for your device
- Useful and efficient for **local communications (LAN)**
- Broadcast domain – the **internet is not possible**
- MAC address is burnt with hardware NIC card – loss of hardware, loss of all existing connections

tells **who** you are

00:1C:A2:01:A3:45
00:A0:C9:14:C8:29
00:1B:44:11:3A:B7
00:06:5B:BC:7A:C7

■ IP Address Only

- They can **group and organize** different networks (hierarchical)
- They are much like your mailing address
- They are **flexible and changeable** – making a device mobile

tells **where** you are



123.456.789.12

Special IP Addresses

Not assignable to a device or interface

▪ **0.0.0.0 (unspecified address)**

- The address of “this host.” (the primary IP address of the machine executing the instruction)
- In DHCP, when a unique address has not yet been determined, 0.0.0.0 used as the Source IP (i.e. DHCP Discover Packet)
- In the context of a Router: 0.0.0.0 represents Default route
- 0.0.0.0 is not assignable to an interface or used as a destination address

▪ **127.0.0.0/8 (or 127.0.0.0 – 127.255.255.255)**

- Send packets back to source. Used to test the protocol stack locally.
- Typically 127.0.0.1

Special IP Address examples

- **0.0.0.18**

The host with the host address 18 on “this local network”

- **255.255.255.255**

Broadcast on this local network

- **161.115.255.255**

Broadcast on target network 161.115.0.0/16

Link-Local IP Address

- **169.254.0.0/16 (or 169.254.0.0 - 169.254.255.255)**
“**valid only for** communications on a **local network segment**”
- **Self-generated automatically**
 - When a host cannot find a DHCP server
 - When communication problems occur between a host and a DHCP server
- **A link local address means**
 - The host cannot access the internet
 - Link local address is not routable
 - The host can communicate with other devices on the same LAN

Routers do not forward packets with link-local addresses

Private IP Addressing

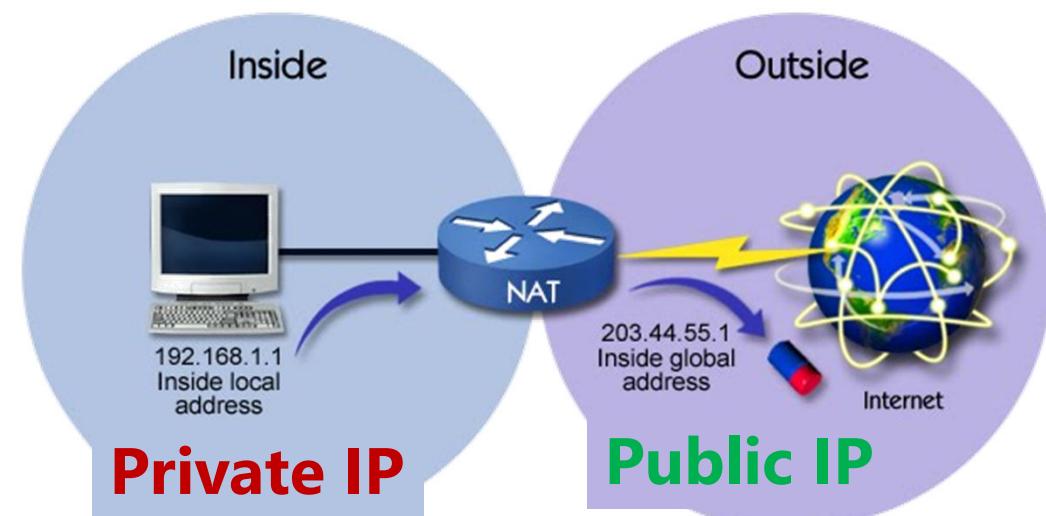
- **What are private IPs?**
 - **Non-routable** addresses **in internet**

- *Routers would not deliver packets with private IP addresses*

- **Free to use** without anyone's permission

▪ Why do we need them?

- There are only 4.3 billion IPv4 addresses
- They have extended IPv4's life



Private IP Addressing – IPv4

Class A	10.0.0.0 to 10.255.255.255	10.0.0.0/8
Class B	172. <u>16</u> .0.0 to 172. <u>31</u> .255.255	172. <u>16</u> .0.0/12
Class C	192.168.0.0 to 192.168.255.255	192.168.0.0/16

Only a portion

Devices with **private IP** address **cannot connect directly to the Internet**

Instead, **access** to the **Internet** must be brokered by a **router** that **supports Network Address Translation (NAT)**

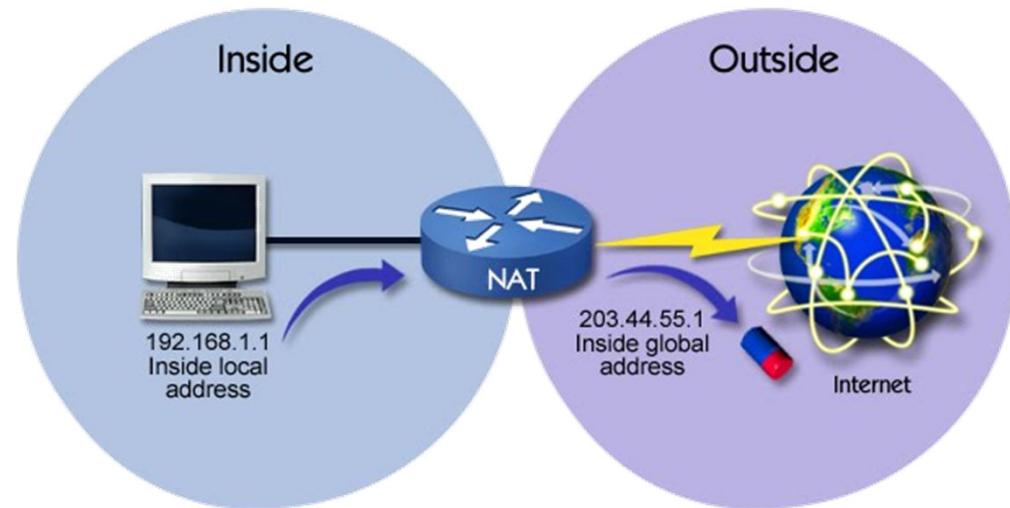


Network Address Translation

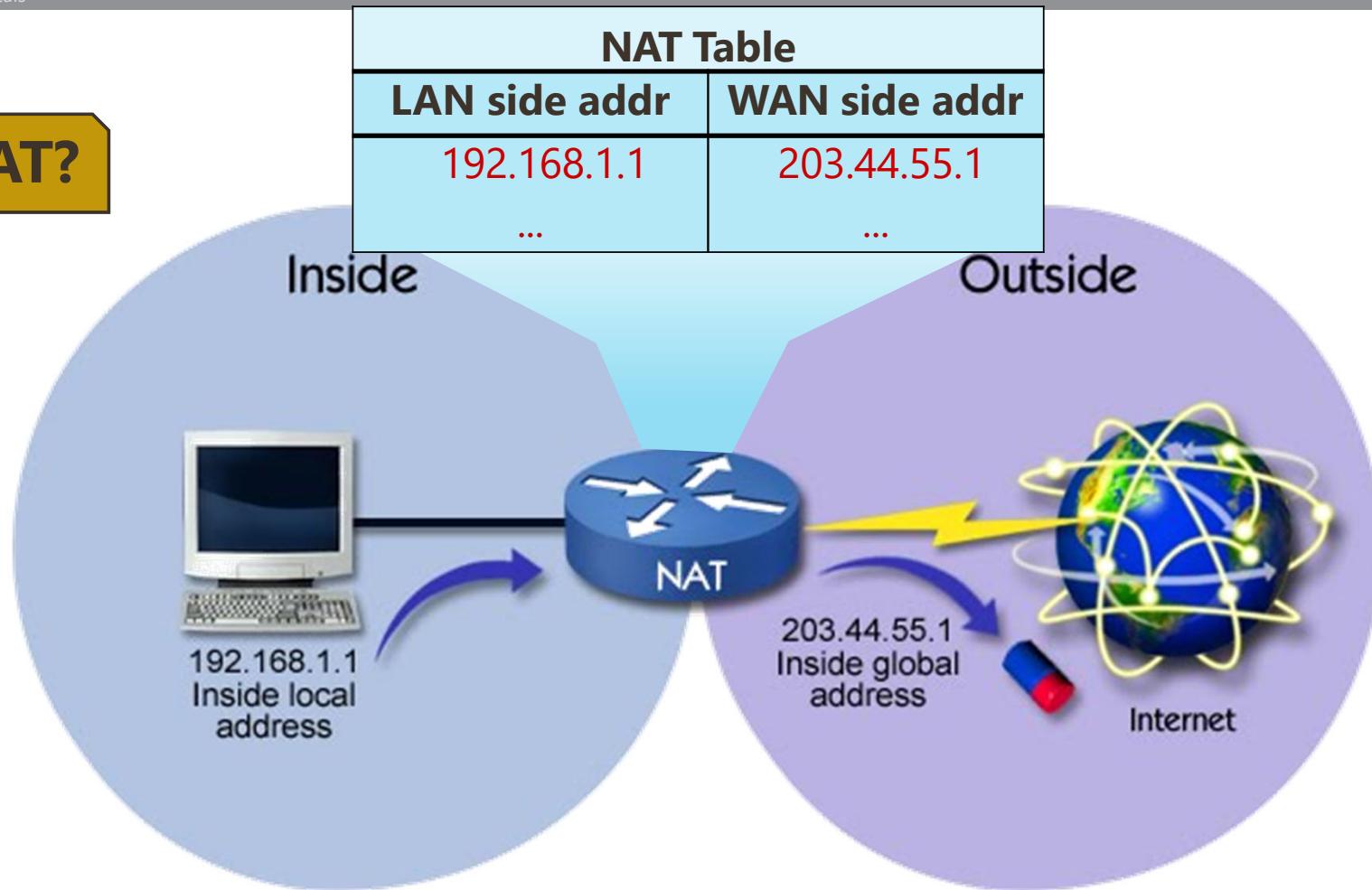
- SNAT
- DNAT
- PAT
- Port Forwarding

Why Network Address Translation?

- **Motivation:** local network uses just one IP address as far as outside world is concerned:
 - range of addresses not needed from ISP: just **one IP address for all devices**
 - **can change addresses of devices** in local network without notifying outside world
 - **can change ISP** without changing addresses of devices in local network
- **Secure:** devices inside local network are not explicitly addressable, visible by outside world



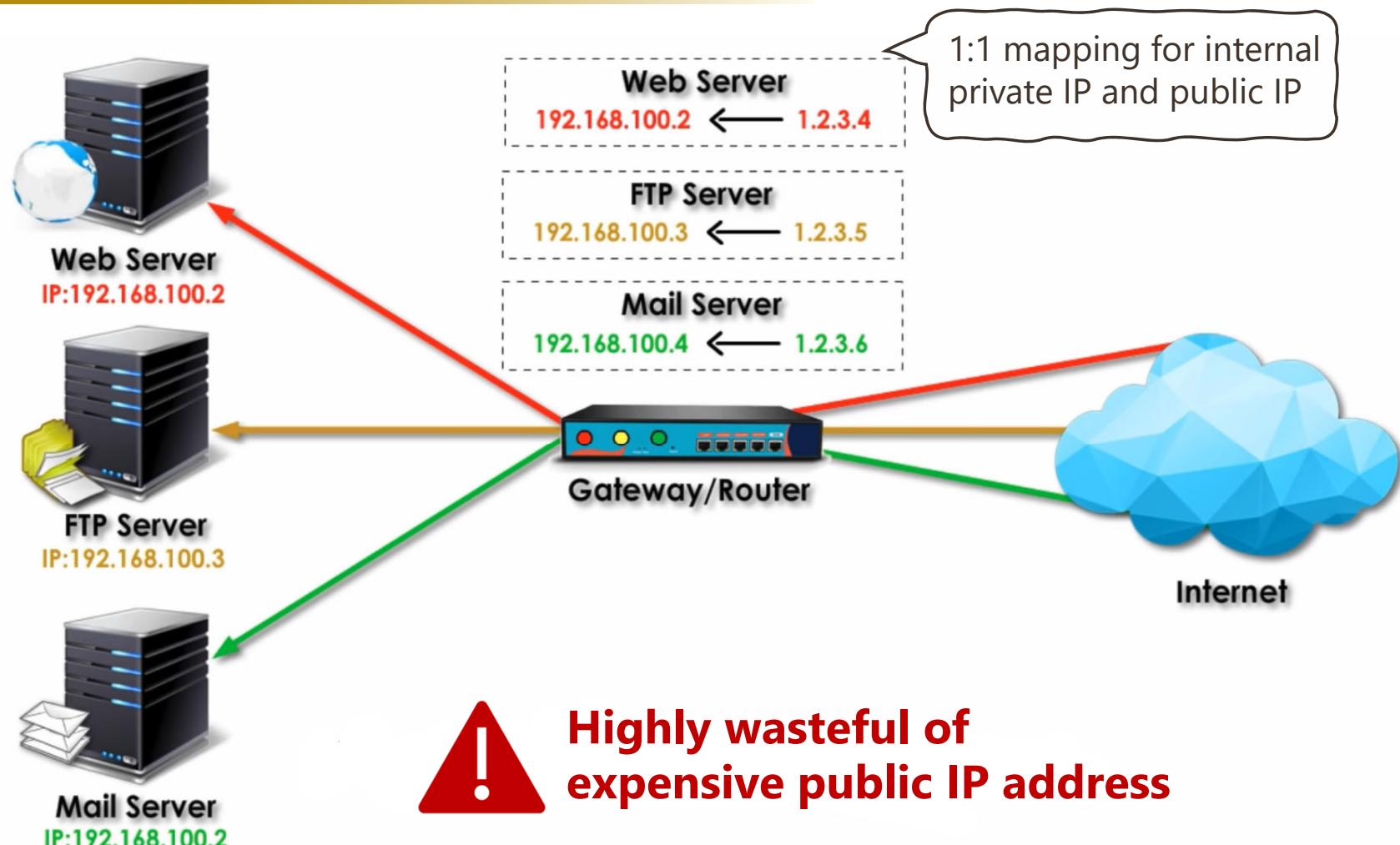
What is NAT?



A router **translate an internal host's private IP** address **into its public IP address** for outgoing/incoming traffic and vice-versa

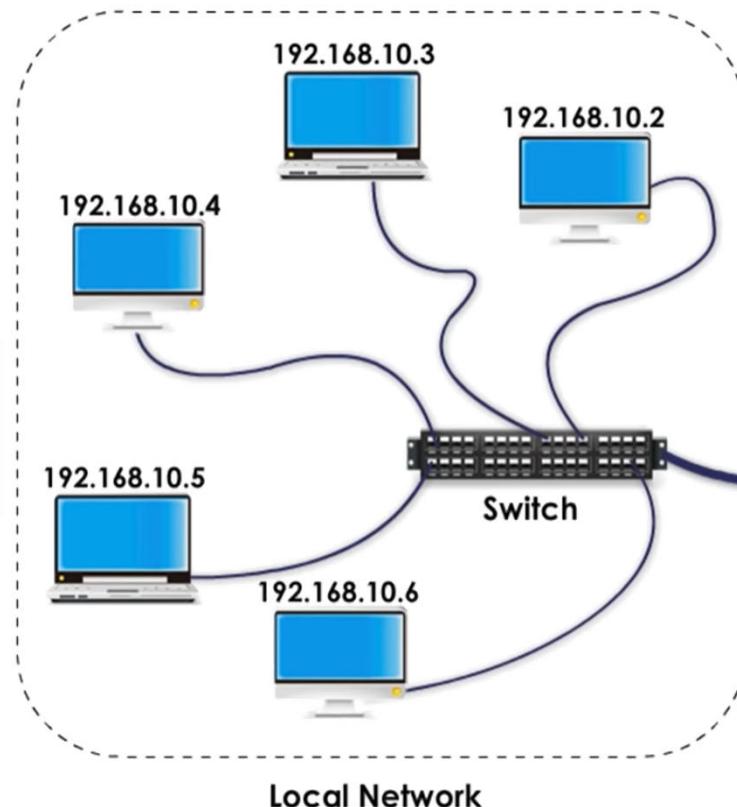
SNAT: Static NAT

deals more with incoming traffic

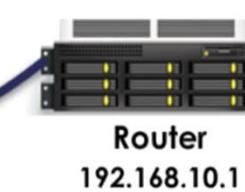


DNAT: Dynamic NAT

deals more with outgoing traffic



Replace internal IP with an IP address of the public IP address pool



Router
192.168.10.1

Internet

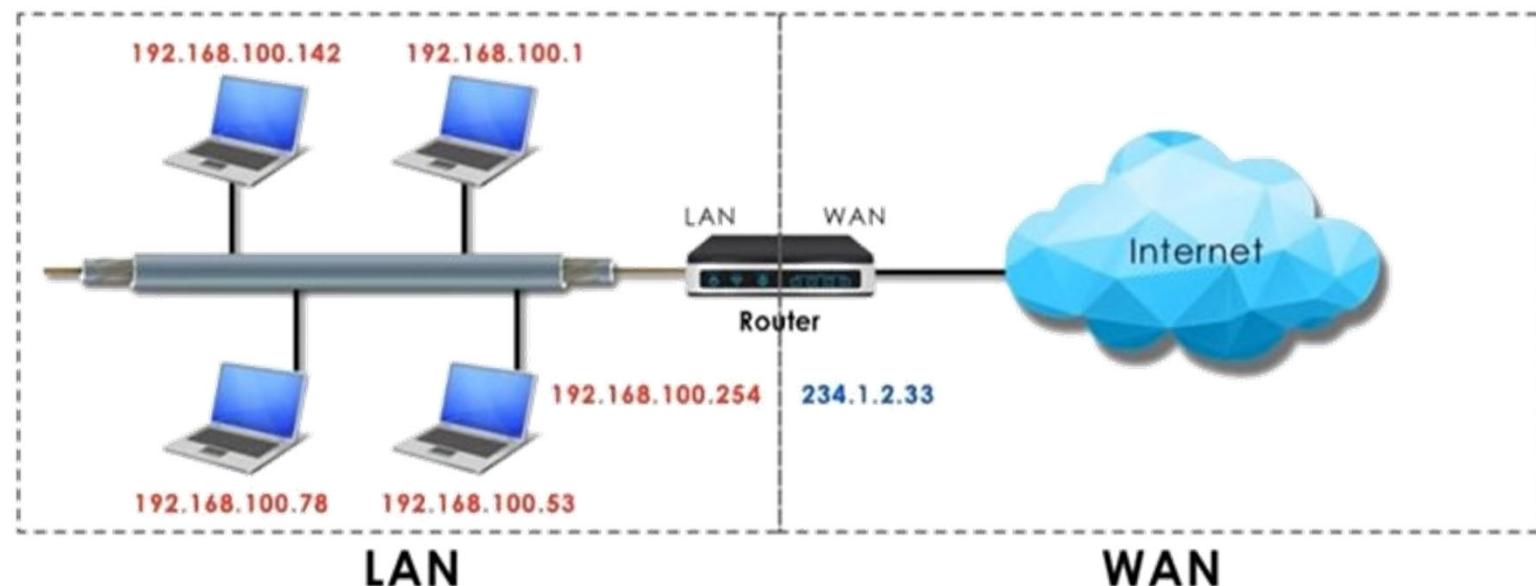


PAT: Port Address Translation

mapping

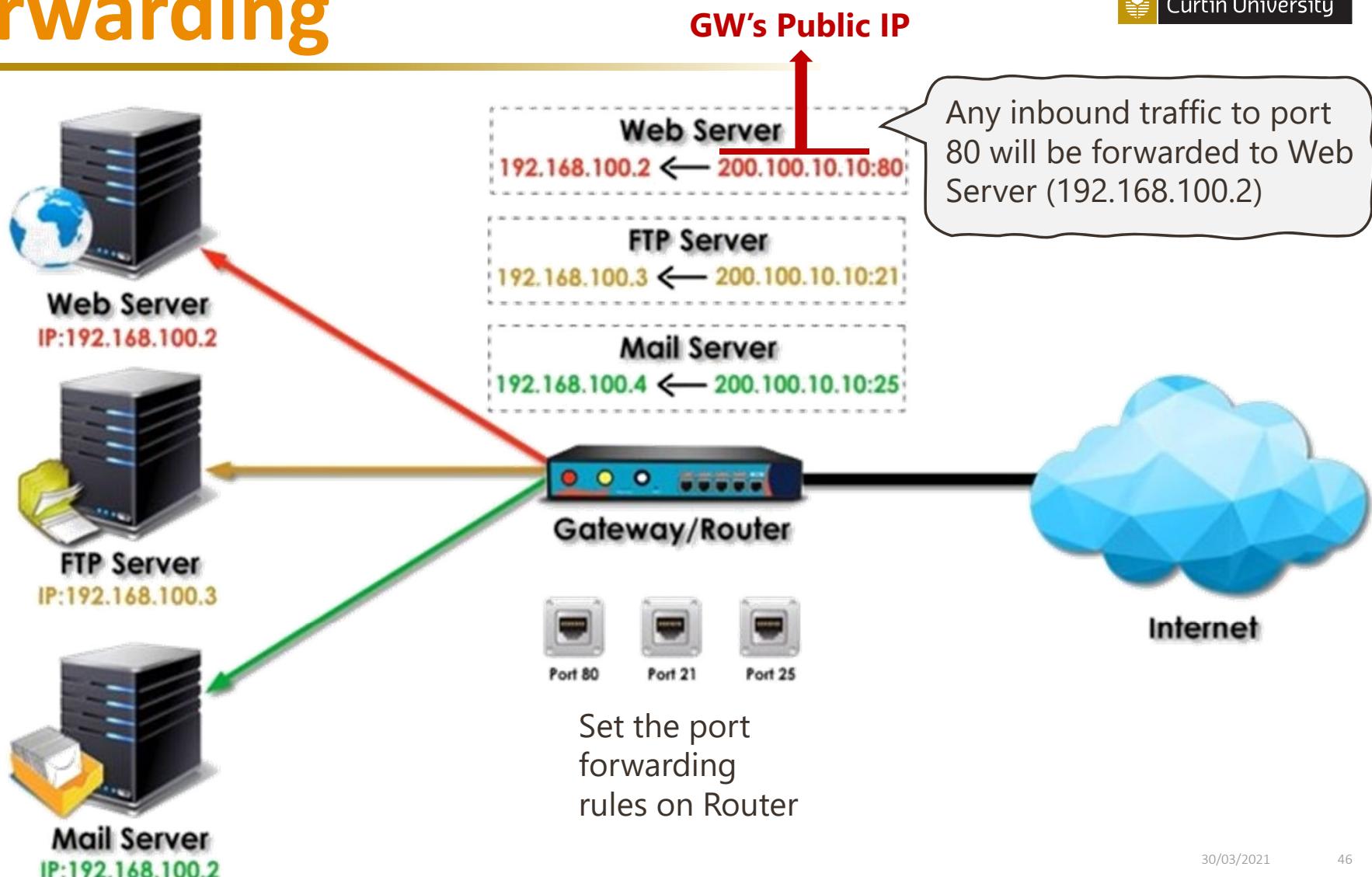
$192.168.100.1:80 = 234.1.2.33:8000$

$192.168.100.78:80 = 234.1.2.33:8001$



Port Forwarding

Port Forwarding
deals more with
incoming traffic





IPv6

- Fundamentals
- Address Space
- IPv6 Datagram
- IPv6 Address Simplification
- Multicast addressing
 - Solicited node address
- Unicast addressing
 - Global Unicast Address

Introduction to IPv6

Initial Motivation:

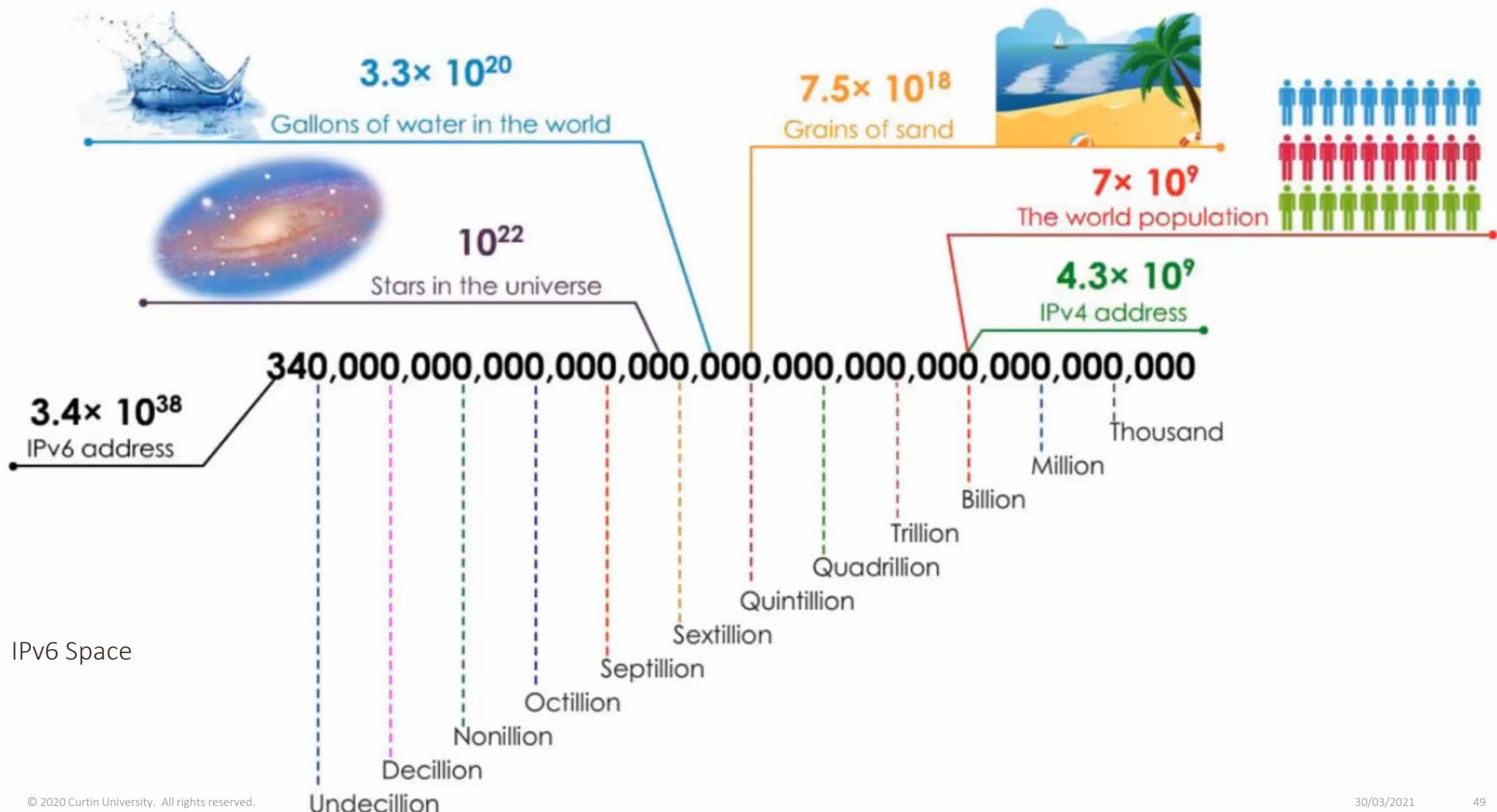
32-bit address space soon to be completely allocated

■ Why IPv6?

- ✓ Supports $2^{128} \sim= 3.4 \times 10^{38}$ addresses

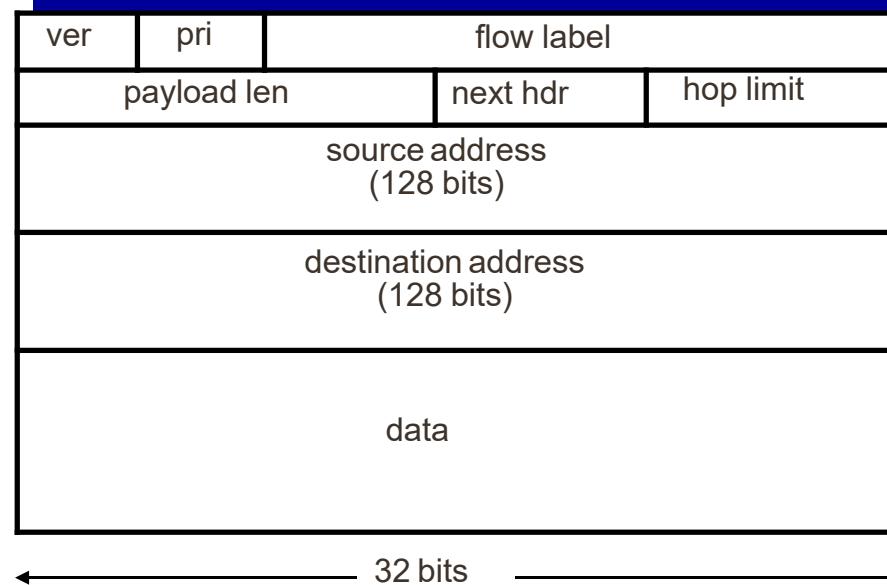
i.e. 2001:000:000:000:0080:ACDE:02CE:1234

- ✓ header format helps speed processing/forwarding
- ✓ header changes to facilitate **QoS**



IPv6 Datagram Format

- **Priority:** identify priority among datagrams in flow
- **Flow Label:** identify datagrams in same “flow.”
(concept of “flow” not well defined)
- **Next Header:** identify upper layer protocol for data



IPv6 Address Simplification

2001:0000:0000:0000:0080:ACDE:02CE:1234

- Leading 0's can be dropped from any group

2001:0:0:0:080:ACDE:02CE:1234 → 2001:0:0:0:80:ACDE:2CE:1234

- Using a pair of colons (::) to represent a string of consecutive groups of 0s

2001:0:0:0:80:ACDE:2CE:1234 → 2001~~0:0:0~~::80:ACDE:02CE:1234

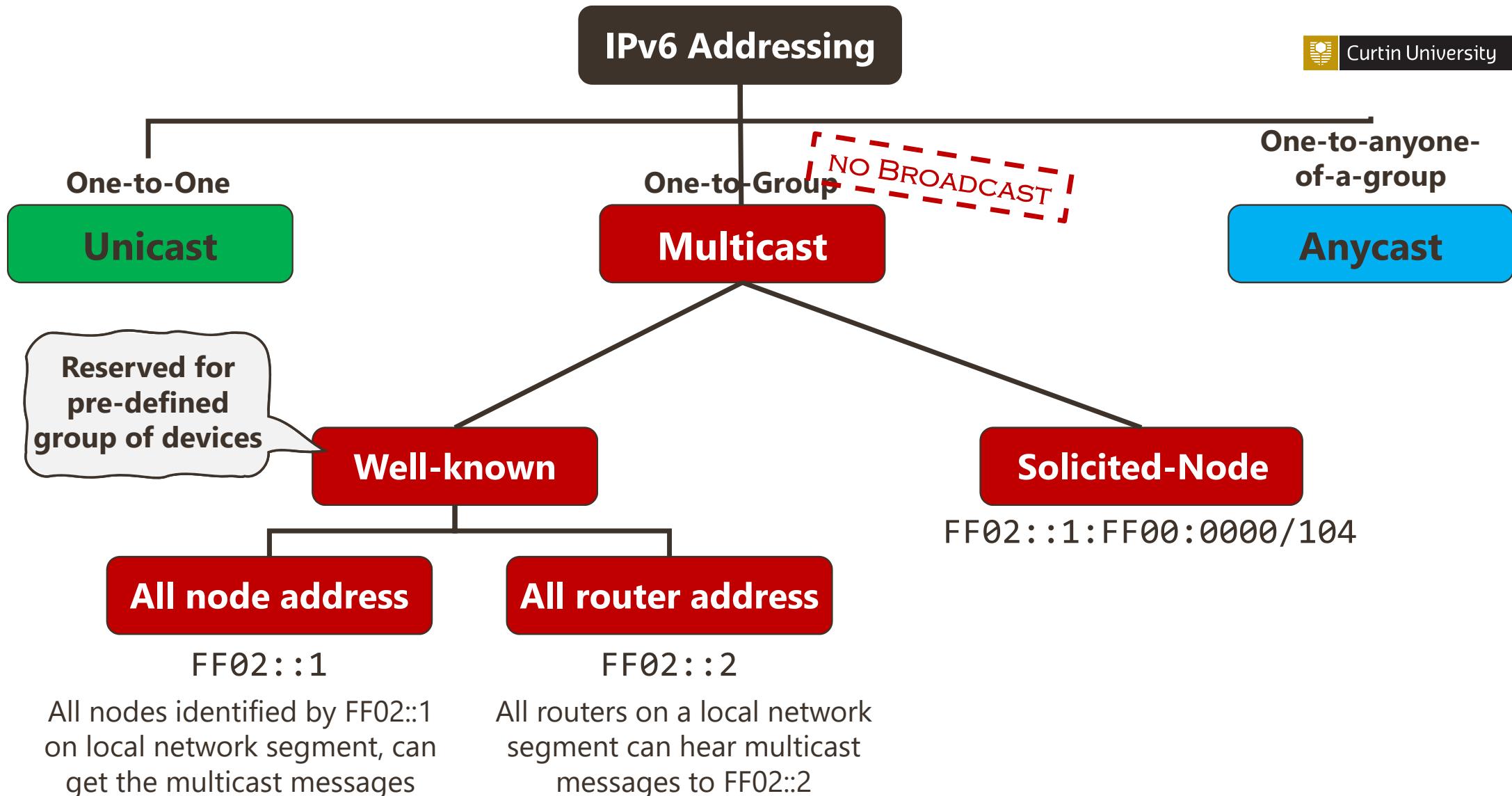
- **Can not use more than one set of colon pairs**

2001:0:0:0:CF:0:0:1234 →

2001::CF:0:0:1234

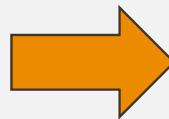
OR

2001:0:0:0:CF::1234



Solicited Node Address

- Created automatically by prepending multicast prefix **FF02::1:FF00:0000/104** to last 24 bits of unicast or anycast address

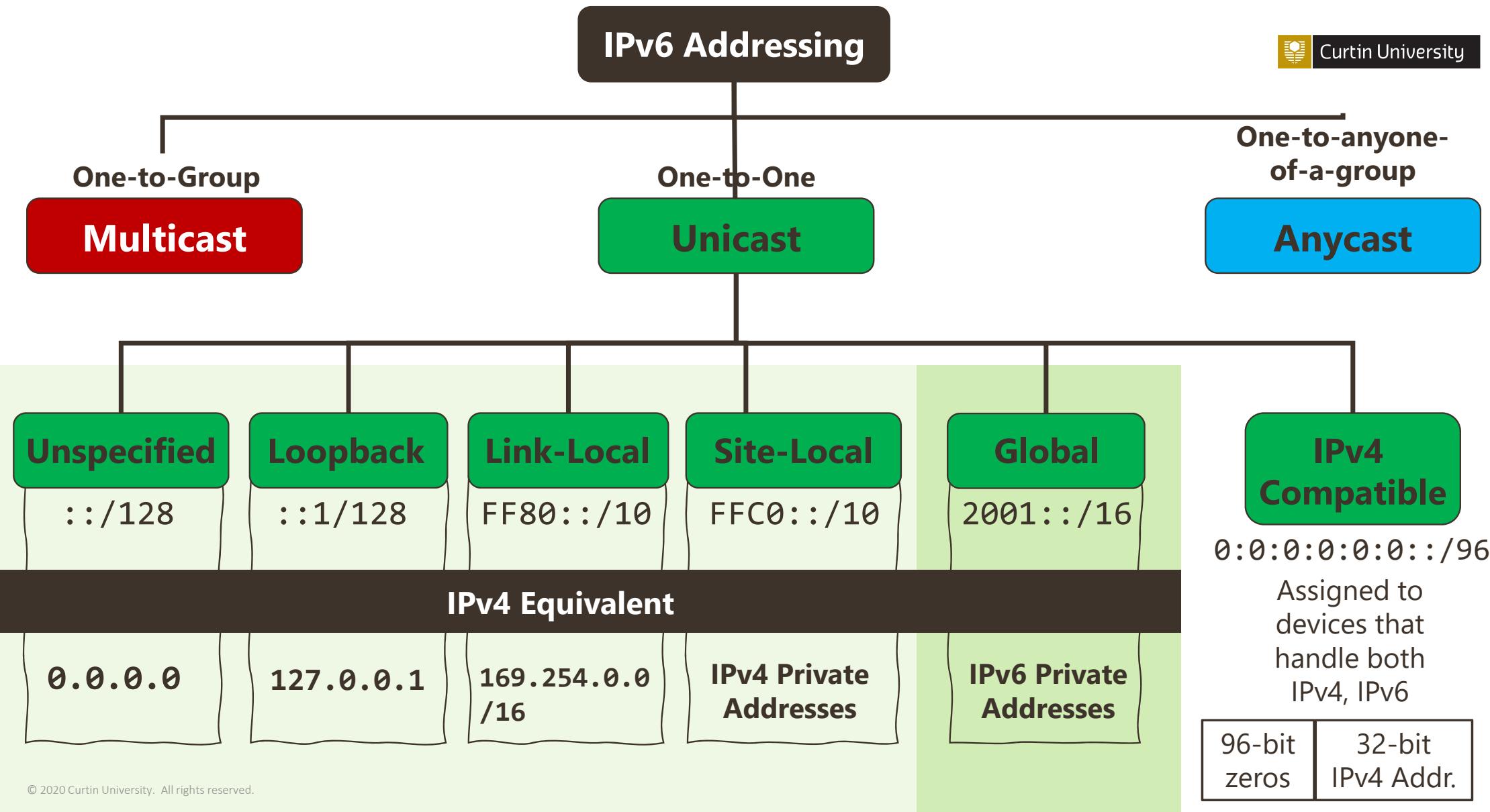
i.e. UnicastIP=2001::01:800:20**0E:8C6C**  FF02::1:**FF0E:8C6C**

- A host is required to join a **solicited-node multicast** group for each of its configured unicast or anycast addresses

IP to MAC Resolution
IPv4 uses ARP,
IPv6 uses **NDP**

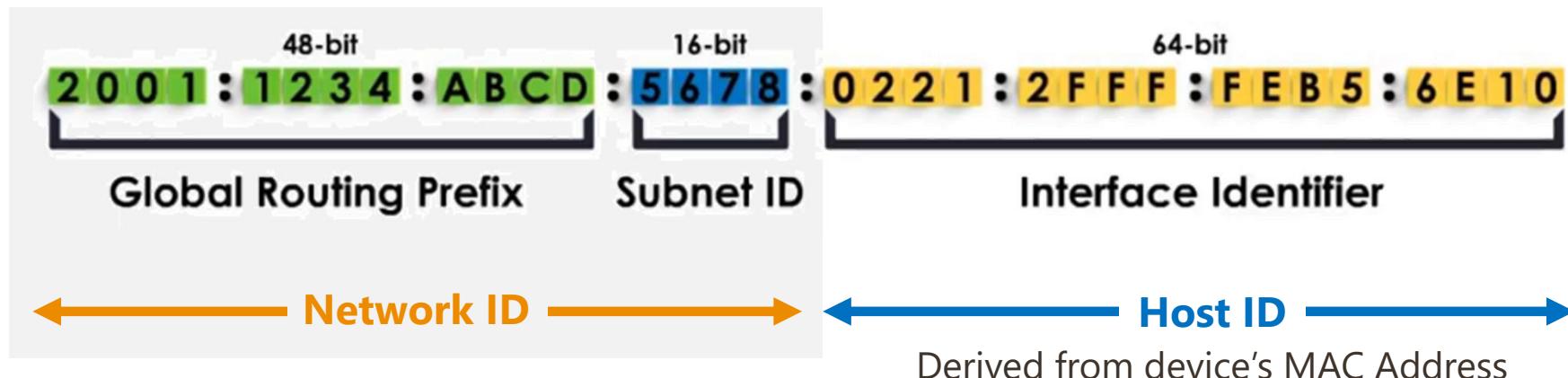
▪ Why Solicited Node Address?

- Useful in link layer address resolution with **Neighbor Discovery Protocol** (NDP) on the link without disturbing all nodes on the local network



Global Unicast

- Similar to IPv4 Public IP Addresses
- Global unicast IPv6 prefixes that are currently allocated by IANA are 2000::/3
 - ✓ They all start with 001 <- binary
 - ✓ At this point, unique global unicast IPv6 address starts with 2001
 - ✓ E.g.



Anyone downstairs will get
2001/16 as routing prefix.

Anyone downstairs will get
2001:1234/32 as routing
prefix.

Anyone downstairs will get
2001:1234:abcd/48 as
routing prefix.

Anyone downstairs will get
2001:1234:abcd:5678/64
as routing prefix.



Network ID Generation

IANA

01

Your regional ISP

02

Your ISP

03

Your company's
default gateway

04





IPv4 to IPv6 Address Transition

- Dual Stack
- Tunnelling
- NAT64 Translation

IPv4 to IPv6 Transition

- IPv4 and IPv6 will coexist for a long time
- *Techniques*

1. Dual Stack

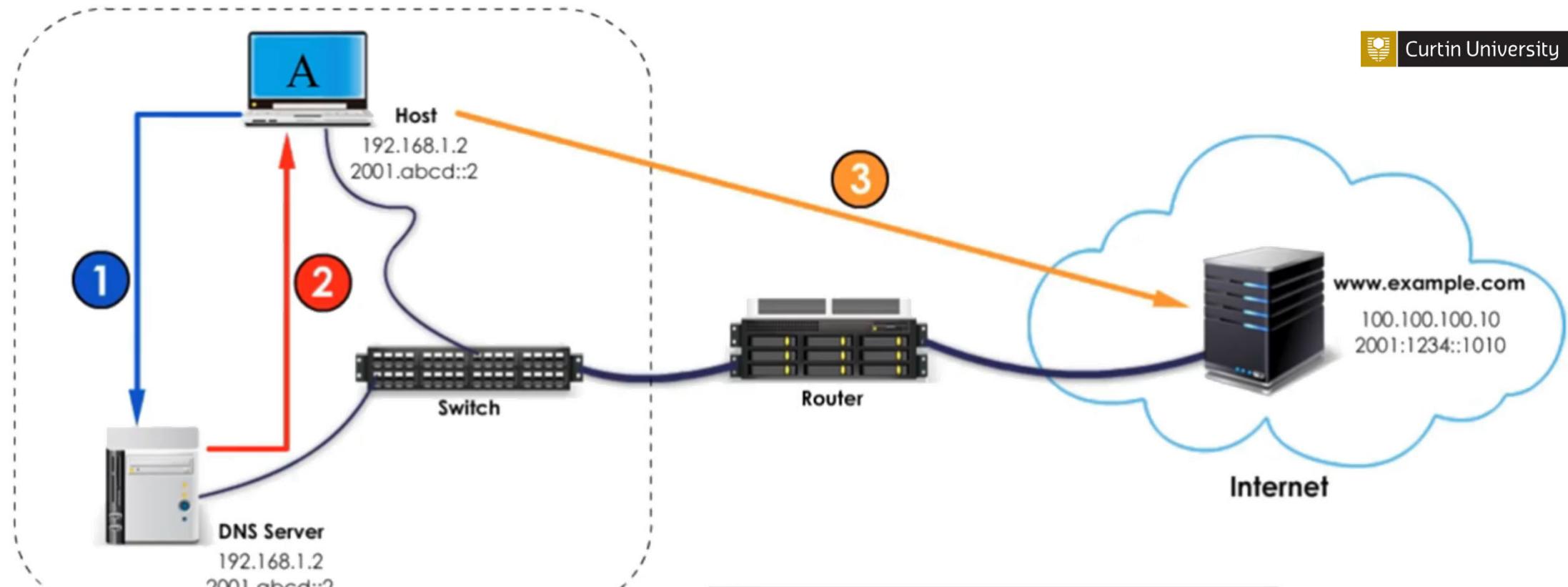
A dual stacked device (PC, Server, Router) supports both IPv4 and IPv6

2. Tunneling

1. Manual Tunneling
2. 6to4
3. ISATAP

3. Translation (NAT64)

Similar to address translation

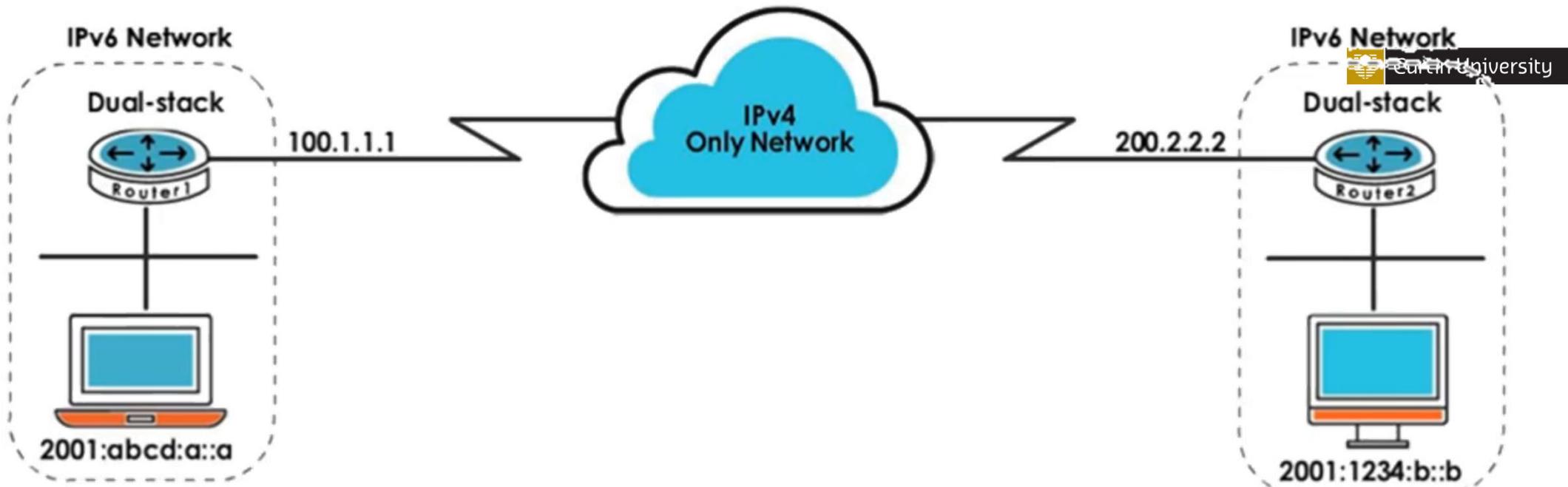


IPv4 and IPv6 Network

Dual Stack

- 1 I need www.example.com IP address
- 2 Type AAAA record: 2001:1234::1010
Type A record: 100.100.100.10
- 3 IPv6 Session with 2001:1234::1010

If failed, IPv4 Session will be established



IPv6 Packet

IPv6 SRC:	IPv6 DST:
2001:abcd:a::a	2001:1234:b::b

Manual Tunnel



IPv6 Packet

IPv6 SRC:	IPv6 DST:
2001:abcd:a::a	2001:1234:b::b

Tunneling

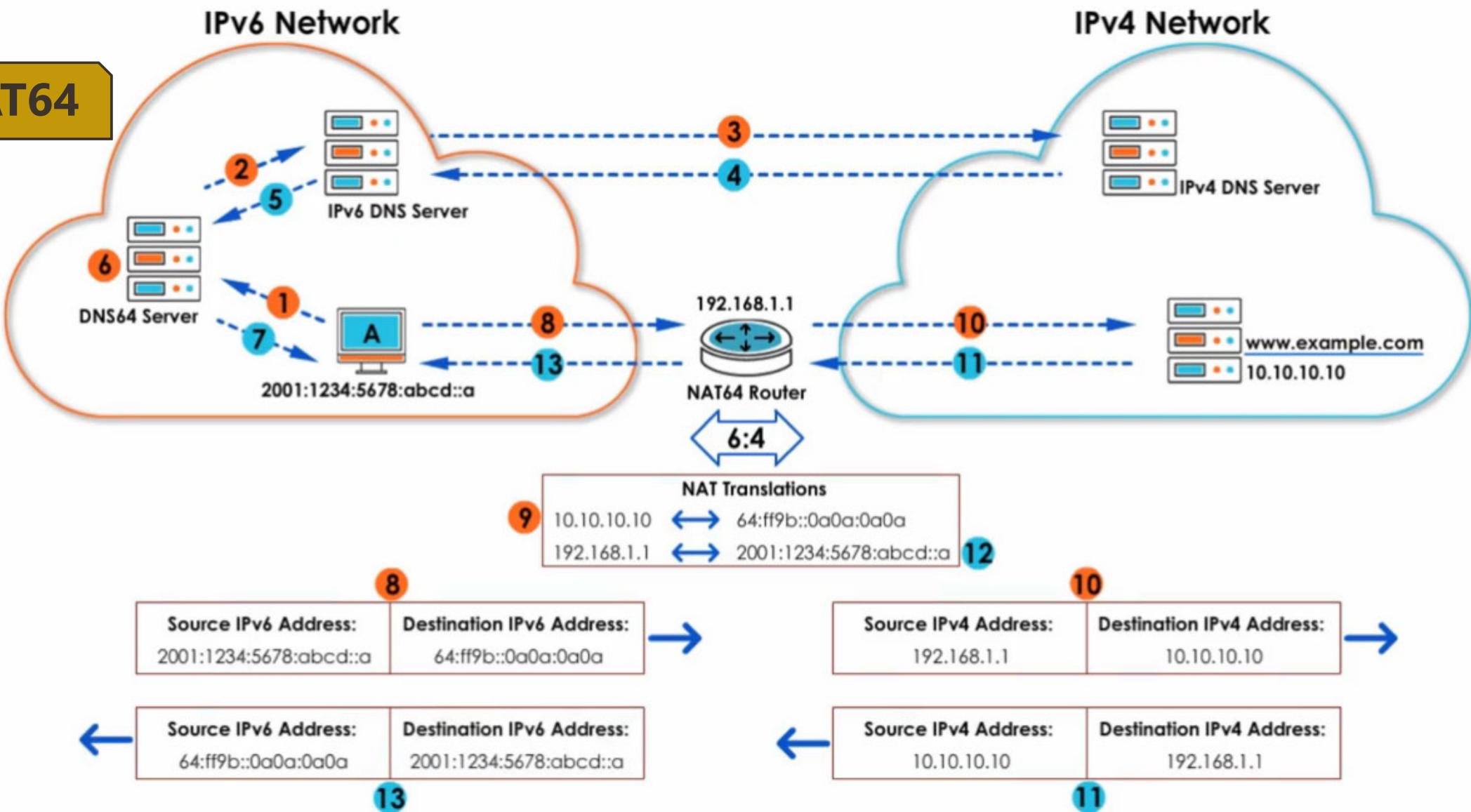
IPv4 SRC: 100.1.1.1
IPv4 DST: 200.2.2.2

41

IPv6 SRC: 2001:abcd:a::a
IPv6 DST: 2001:1234:b::b

Data

Payload – IPv6 Packet



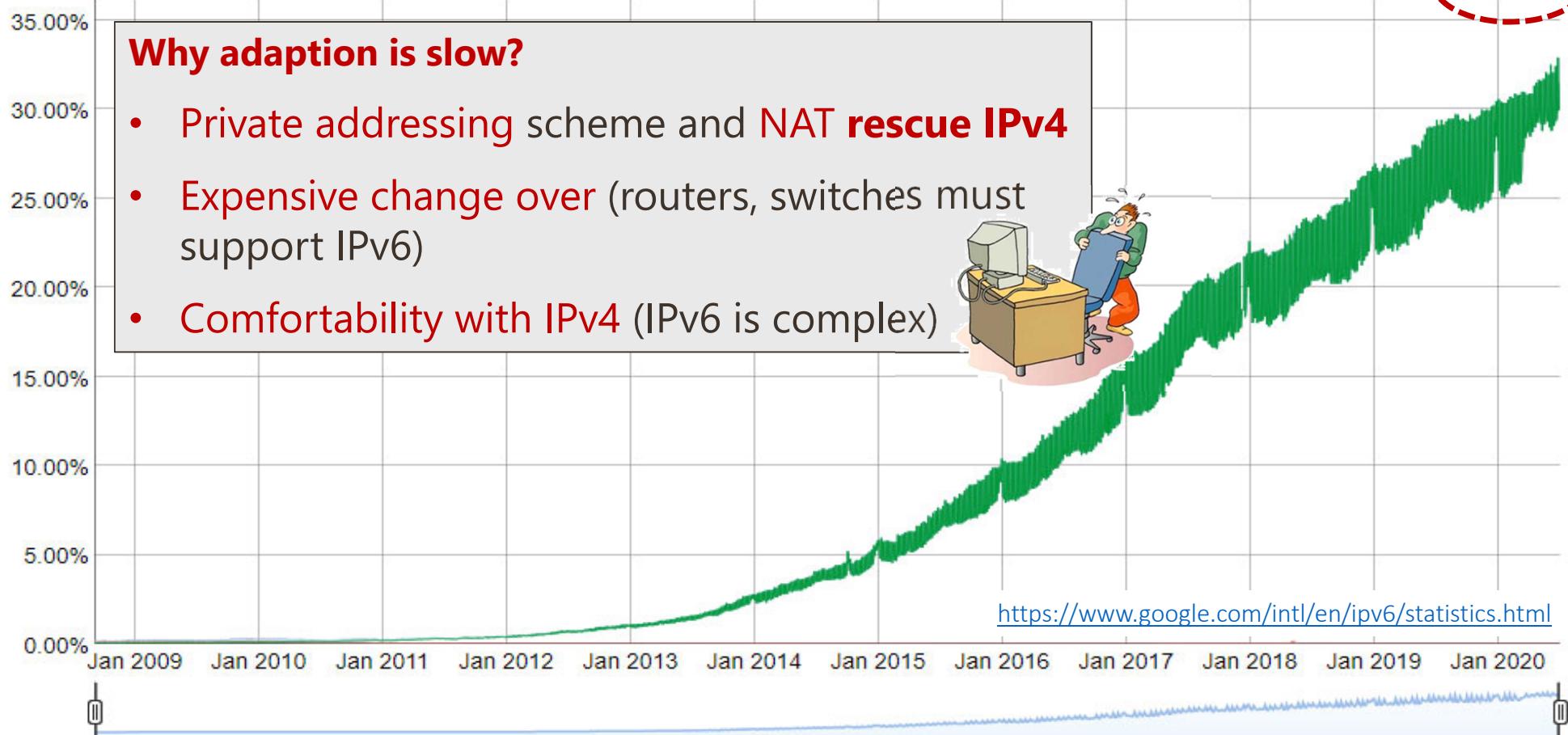
IPv6 Adoption

Per-Country IPv6 adoption

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

Native: 31.49% 6to4/Teredo: 0.01% Total IPv6: 31.50% | Jul 3, 2020



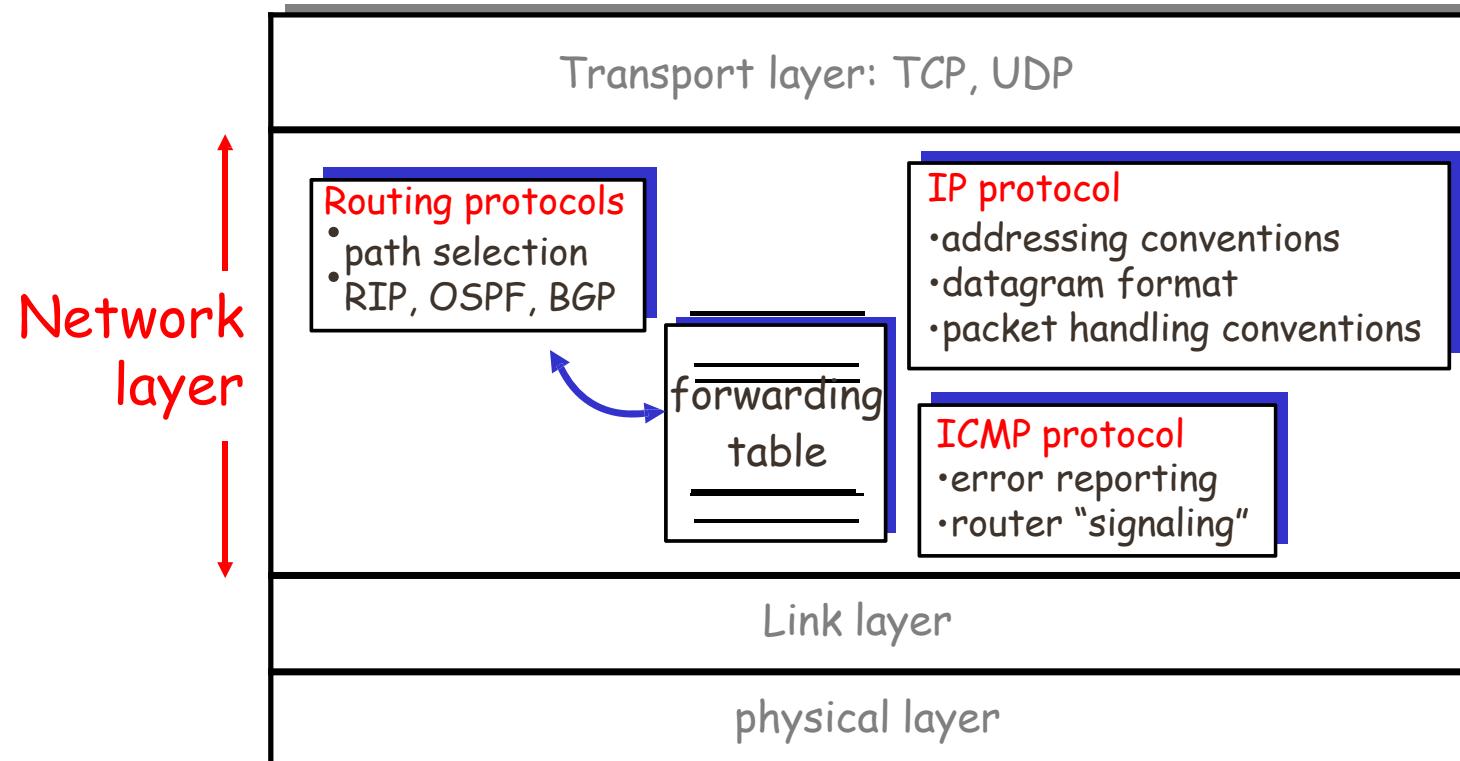
**IPv6
Last
Words**



Network Layer Protocols

- ICMP
- Traceroute Utility
- ARP

Network Layer Protocols



ICMP: Internet Control Message Protocol

- Used by hosts & routers to communicate network-level information

- **error reporting:** unreachable host, network, port, protocol
- echo request/reply (used by **ping**)

▪ ICMP message

- type, code plus first 8 bytes of IP datagram causing error

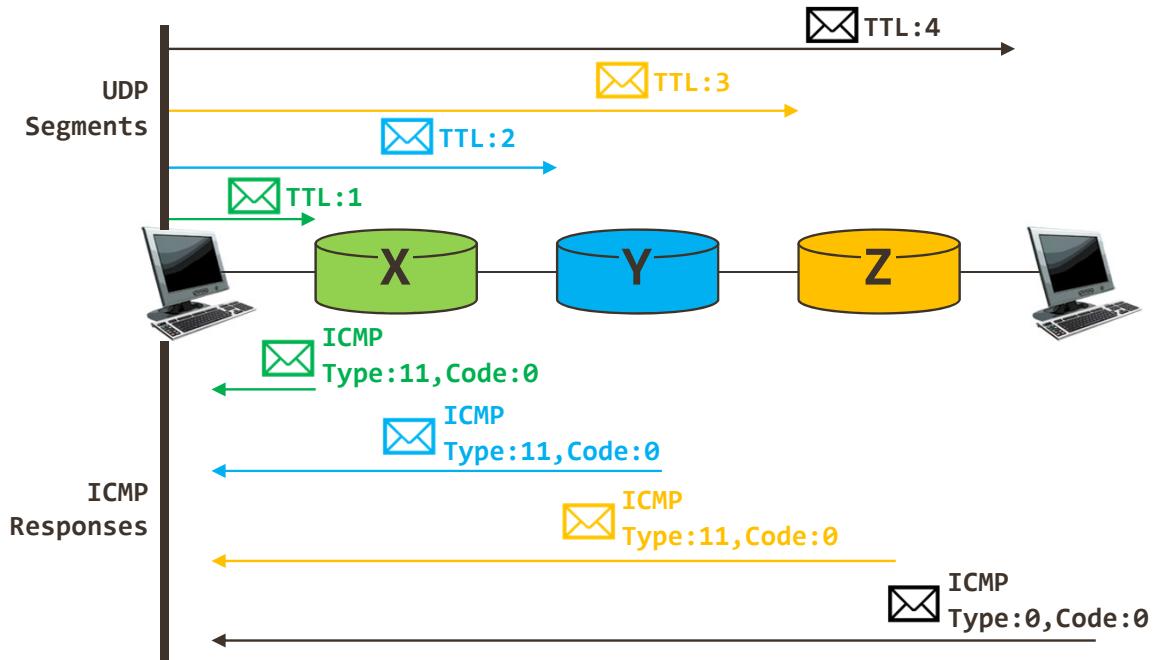
▪ ICMPv6: (IPv6)

- Additional message types, e.g. "Packet Too Big"
- Multicast group management functions

Type	Code	<u>description</u>
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

ICMP & traceroute utility

- Source sends series of **UDP segments** to destination
- When **TTL** expires on a router
 - Router discards datagram
 - Sends to source an ICMP message (type 11, code 0)
 - Message includes name of router & IP address
- When **ICMP** message arrives, **source** calculates **RTT**
- **Stopping criterion**
 - UDP segment arrives at destination host
 - Destination returns ICMP "host unreachable" (type 3, code 3)



ARP: Address Resolution Protocol

- Maps an **IP Address** to a physical **MAC Address on a LAN**
- A computer uses **ARP program** to find another computer's MAC address based on its IP address

LAYER 2
PROTOCOL

- Why MAC address?
 - **Communication within LAN:** MAC address is used
 - **Communication between LANs:** IP address is used

- **ARP cache:**
 - A table of IP address with their corresponding MAC addresses



ARP Cache

C:\>arp -a		
Internet Address	Physical Address	Type
10.172.112.1	cc-4e-24-1a-d7-00	dynamic
10.172.117.6	0c-84-dc-8e-5a-6b	dynamic
10.172.117.189	4c-0b-be-2c-f7-02	dynamic
10.172.119.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Static ARP Entries

- Manually entered

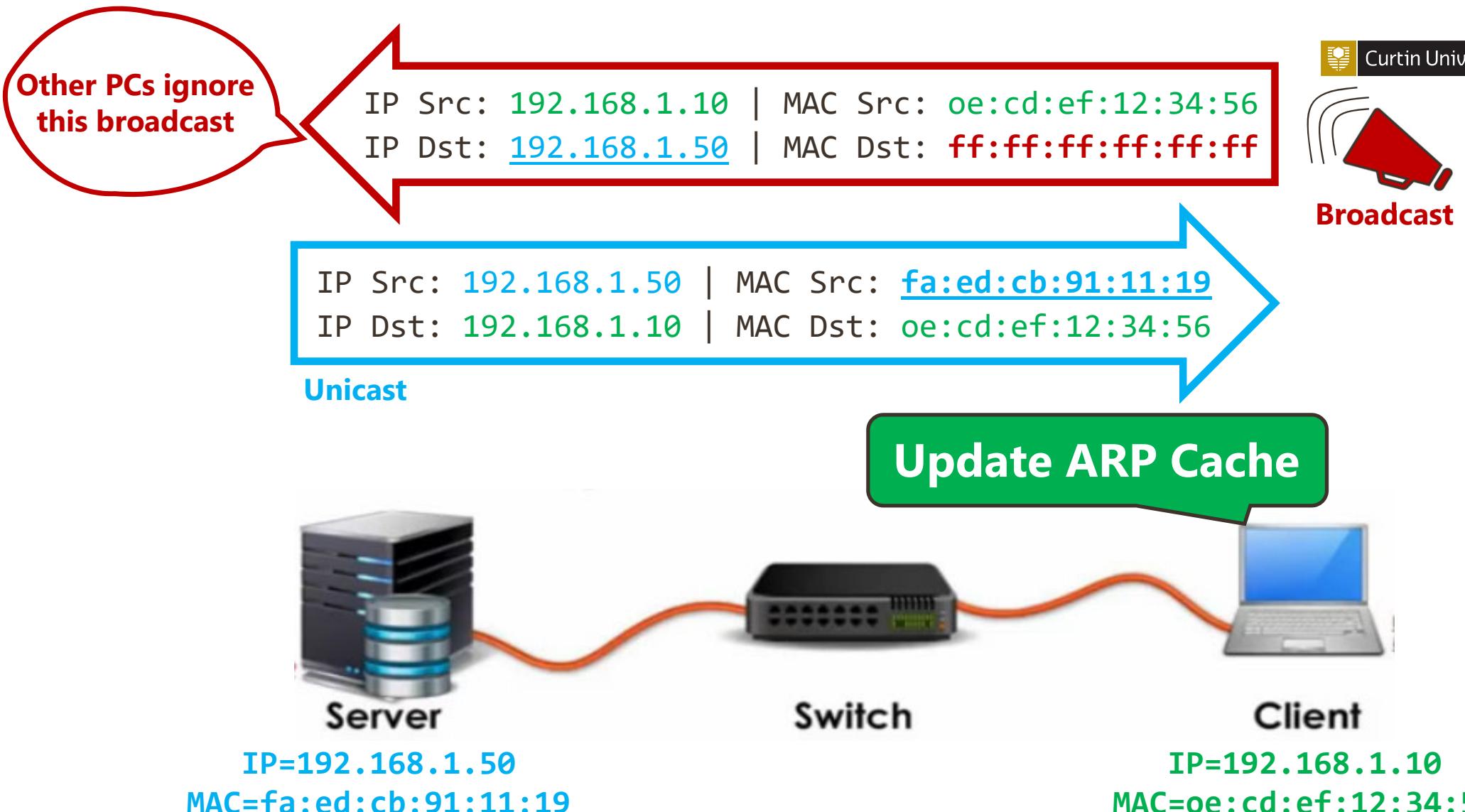
Dynamic ARP Entries

- Via ARP program



ARP Cache Timeout

Timeout for ARP entries





■ Network Layer

- Key Functions
- Services
- Virtual Circuit Networks
- Datagram Networks
 - IP datagram
 - Fragmentation

■ Address Types

- Unicast
- Multicast
- Broadcast
- Anycast
- Geocast

■ IPv4 Addressing

- Classful / Classless Addressing
- VLSM
- Static vs Dynamic IP Addresses
- Obtaining a Global IP Address
 - ISP Address Allocation
 - Hierarchical Addressing
- Special IPv4 Addresses
 - 0.0.0.0, loopback, broadcast, target broadcast
 - Link-local IP address
 - Private IP address

■ Address Translation

- SNAT
- DNAT
- PAT
- Port Forwarding

■ IPv6

- Address Space
- IPv6 Datagram
- IPv6 Address Simplification
- Multicast addressing
 - Solicited node address
- Unicast addressing
 - Global Unicast Address
- IPv6-IPv4 Transition

■ Network Layer Protocols

- ICMP
- Traceroute Utility
- ARP





Curtin University

THANK YOU

Make tomorrow better.



Curtin University

Network Layer II

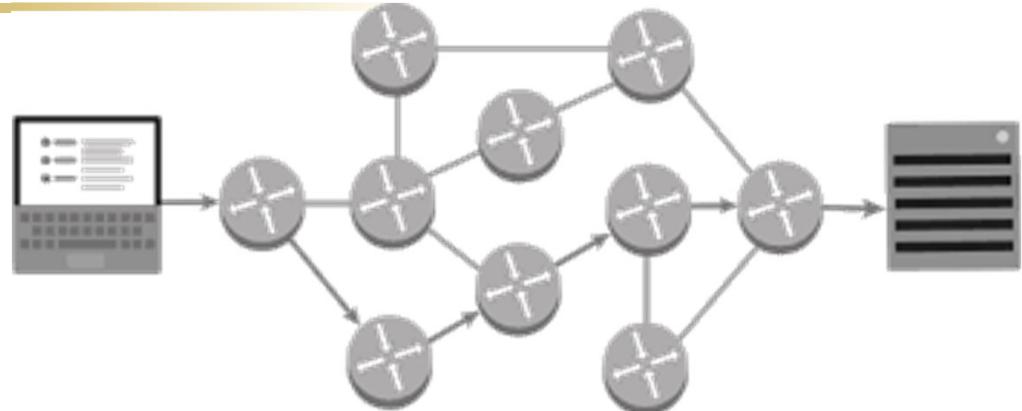
Prof. Ling Li | Dr. Nadith Pathirage | Lecture 06

Semester 1, 2021

A GLOBAL UNIVERSITY

WESTERN AUSTRALIA | DUBAI | MALAYSIA | MAURITIUS | SINGAPORE

Network Layer: Routing



Forwarding packet towards
destination network/host

Shortest path, load balancing, etc.

Routing Classification

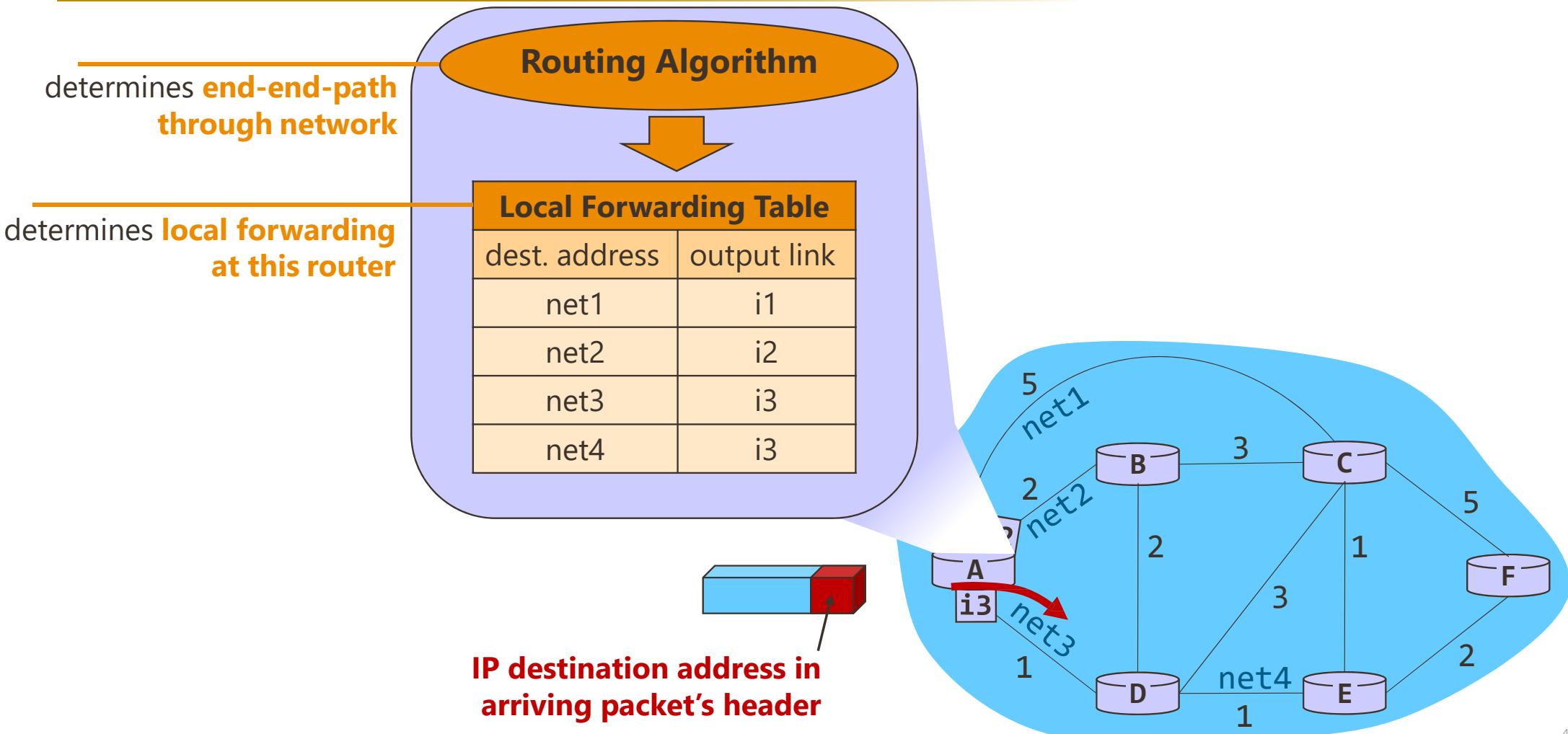
Adaptive (Dynamic) Routing

- Changes routes dynamically
 - Periodically
 - When load changes
 - When topology changes
- Gather information runtime
 - Locally
 - From adjacent routers
 - From all other routers
- Uses routing protocols
 - RIP, OSPF, IGRP

Non-Adaptive (Static) Routing

- Performed manually
- Choice of route is computed in advance, offline & downloaded to the routers when network is booted

Router: Algorithm & Forwarding



Routing Protocols

Combination of rules and procedures that let routers inform each other of changes

Intra-domain: works only within domains
Inter-domain: works within and between domains

Intra-Domain

Inter-Domain

Routing Algorithms

Distance Vector

Link State

Path Vector

Routing Protocols

RIPv1/v2

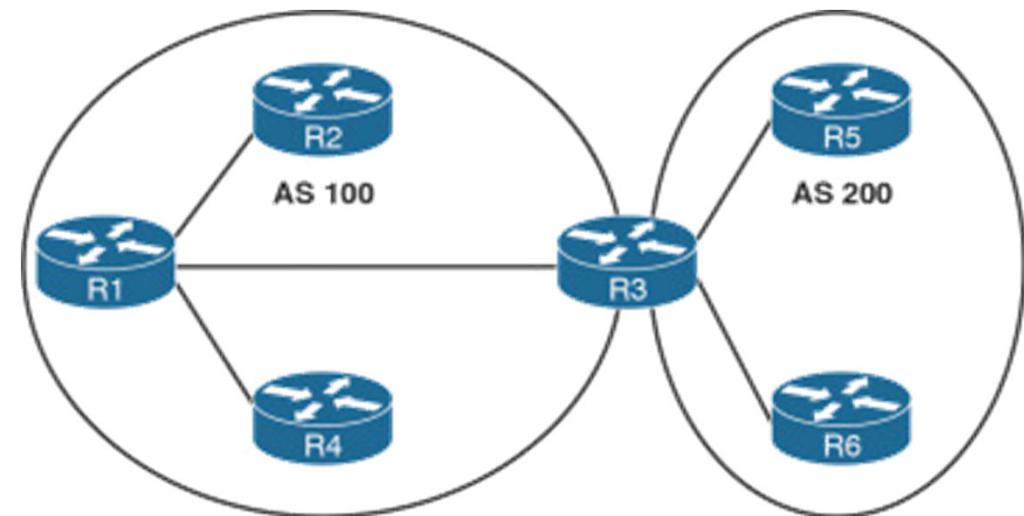
IGRP

OSPF

BGP

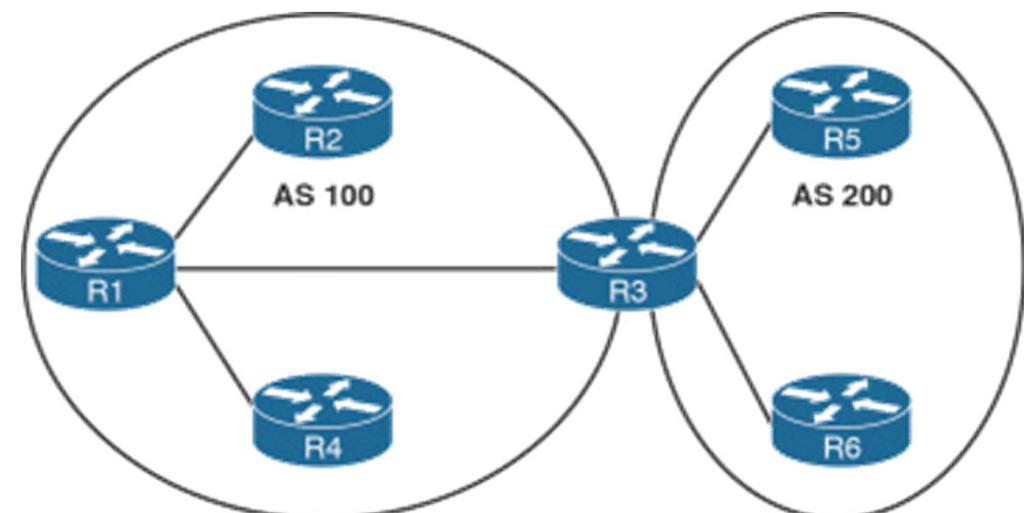
Intra-domain Routing

- **Routing** algorithm works **only within domains**
 - ✓ *Only aware of other routers within their domain*
- Protocols used: **Interior-gateway protocols (IGP)**
- **Routing within an autonomous network**
- **Ignores** the internet **outside** the **AS**(autonomous system)

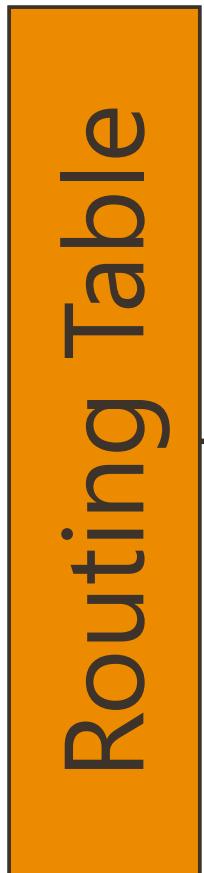


Inter-domain Routing

- **Routing** algorithm works **within and between domains**
 - ✓ Aware of other routers within and between their domain
- Protocols used: **Exterior-gateway protocols (EGP)**
- **Routing between the autonomous networks**
- **Assumes** the internet contains the **collection of interconnected AS**(autonomous systems)



Routing Table



Static Table

Manual entries with
ip route

Dynamic Table

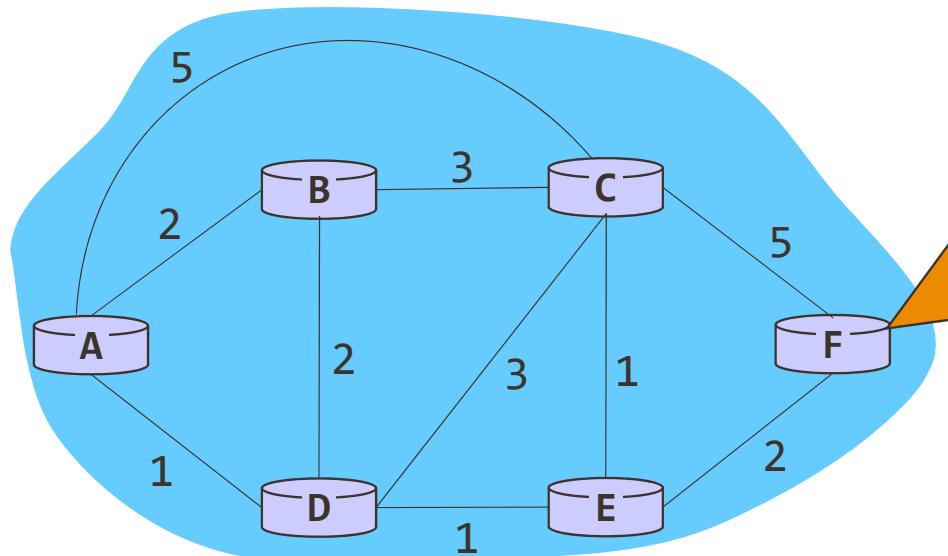
Automatic entries with routing protocols
when there is a change in the network.
i.e. **RIP, IGRP, OSPF, etc**



Routing Algorithms

- Link State Routing
 - Topology Dissemination
 - Computing Shortest Path (Dijkstra)
- Distance Vector Routing
 - Bellman Ford Algorithm
 - Distance Vector Updates

Graph Abstraction



cost could always be 1, or
 $\propto \frac{1}{\text{Bandwidth}}$
 $\propto \text{Congestion}$

key question: what is the least-cost path between S and R?

routing algorithm: algorithm that finds that least cost path

Routing Algorithms: Packet Switched Networks

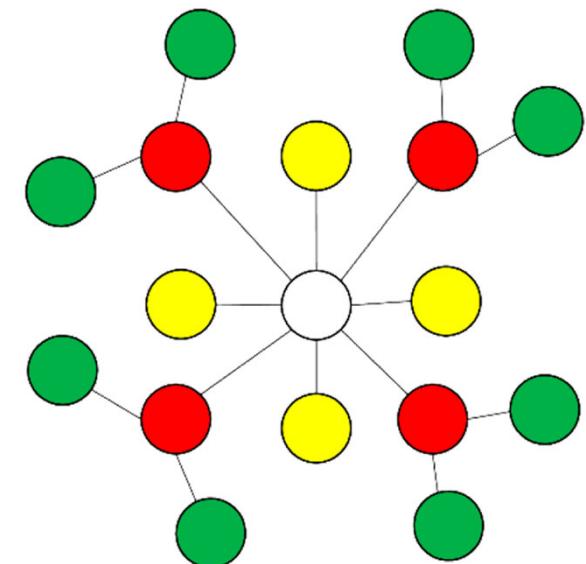
- **Link-State & Distance Vector** assume:

- a router knows
 - ✓ the address of each neighbor
 - ✓ cost of reaching each neighbor

- **Link State:** A node tells every other node in the network its distance to its neighbors

- **Distance Vector:** A node tells its neighbors its distance to every other node in the network

- Both are **distributed**



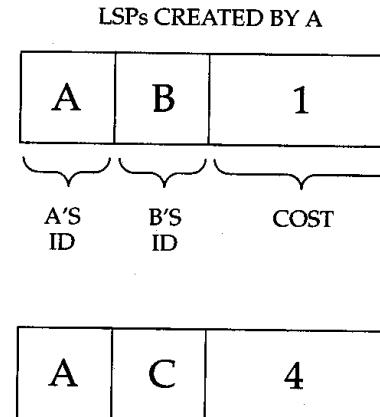
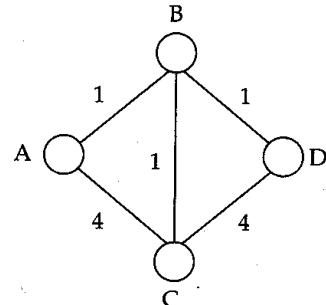
Link State Routing

- Router tells every other routers in the network its distance to its neighbors
- Router knows entire network topology (global information), and computes shortest path by itself
- Key elements
 1. Topology Dissemination
 2. Computing Shortest Routes (i.e. Dijkstra)

Independent computation of routes

1. Topology Dissemination

- A router describes its neighbors with a link state packet (LSP)



- Use controlled flooding to distribute this info to everywhere
 - store LSPs in an LSP database
 - if new, forward to every interface other than incoming one

2. Computing Shortest Routes: Dijkstra's Algorithm

- Assume network topology, link costs known to all nodes
 - accomplished via “link state broadcast”
 - all nodes have same info
- **Computes least cost paths** from one node ('source') to all other nodes
- **LSA exchange -> LSDB -> Dijkstra -> Routing Table** for the node

Link State
Advertisement



After **k iterations**, know least cost path to **k destinations**

Dijkstra's Algorithm

Notation:

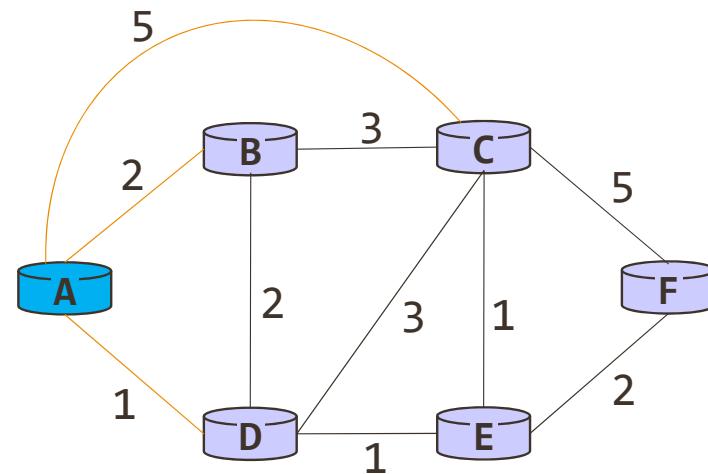
- **c(i,j)**: Link cost from node **i** to **j**; cost infinite if not direct neighbors
- **L(v)**: Current value of cost of path from source to destination **v**
- **p(v)**: Predecessor node along path from source to **v**, that is next **v**
- **N**: Set of nodes whose least cost path definitively known

```

1 Initialization:
2 N = {A}
3 for all nodes v
4   if v adjacent to A
5     L(v) = c(A,v)
6   else
7     L(v) = infinity
8
9 Loop
10 Find w not in N such that L(w) is a minimum
11 Add w to N
12 Update L(v) for all v adjacent to w and not in N
13   L(v) = min(L(v), L(w) + c(w,v))
14 // new cost to v is either old cost to v or
// known shortest path cost to w plus cost from w
// to v
15 Until all nodes in N

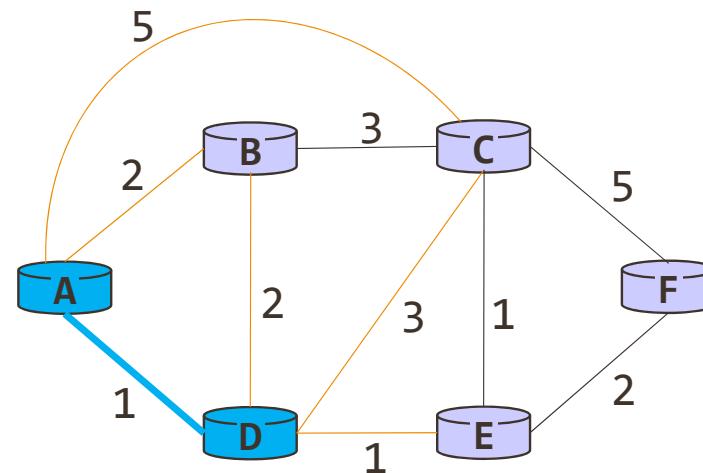
```

Dijkstra's Algorithm



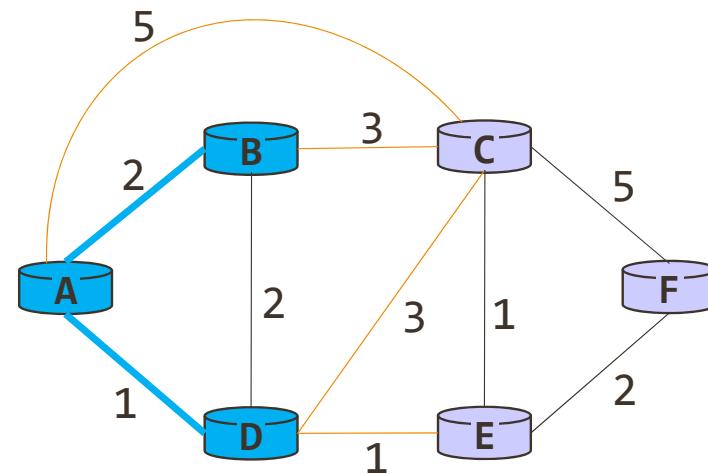
#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2											
3											
4											
5											
6											

Dijkstra's Algorithm



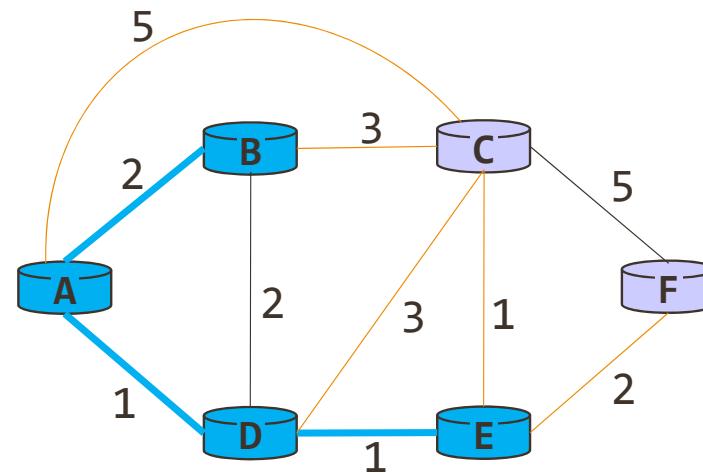
#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3											
4											
5											
6											

Dijkstra's Algorithm



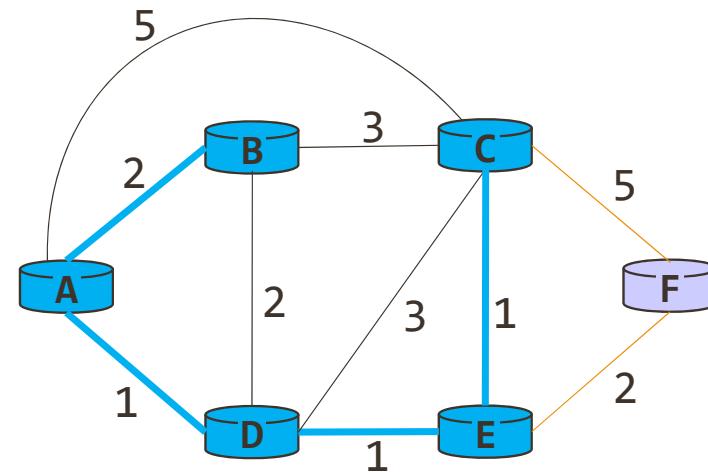
#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3	{A,D,B}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
4											
5											
6											

Dijkstra's Algorithm



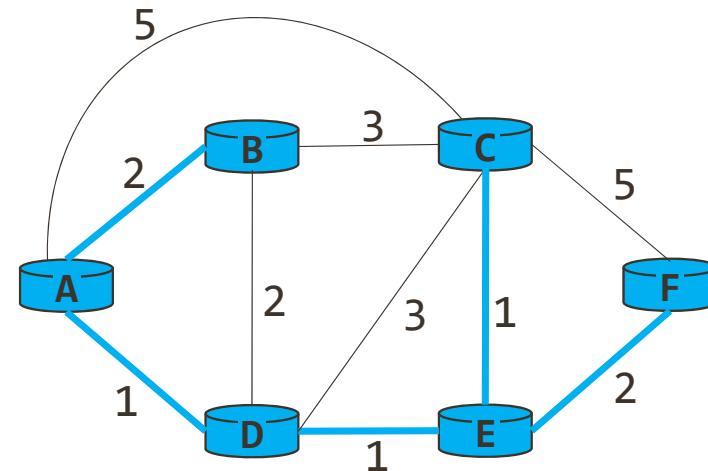
#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3	{A,D,B}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
4	{A,D,B,E}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
5											
6											

Dijkstra's Algorithm



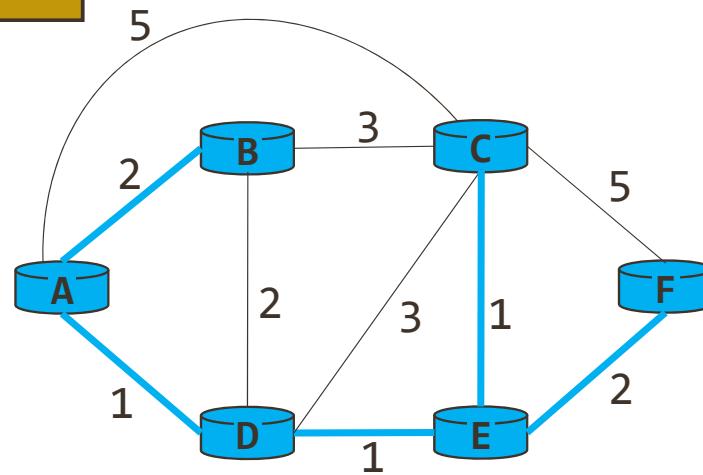
#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3	{A,D,B}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
4	{A,D,B,E}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
5	{A,D,B,E,C}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
6											

Dijkstra's Algorithm

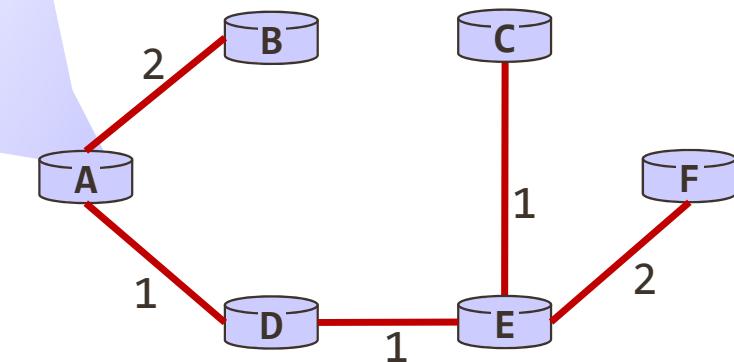


#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3	{A,D,B}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
4	{A,D,B,E}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
5	{A,D,B,E,C}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
6	{A,D,B,E,C,F}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F

Dijkstra's Algorithm



Dest.	Cost	Hop
B	2	B
C	3	D
D	1	D
E	2	D
F	4	D



#	N	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path	L(F)	Path
1	{A}	2	A-B	5	A-C	1	A-D	inf	-	inf	-
2	{A,D}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
3	{A,D,B}	2	A-B	4	A-D-C	1	A-D	2	A-D-E	inf	-
4	{A,D,B,E}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
5	{A,D,B,E,C}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F
6	{A,D,B,E,C,F}	2	A-B	3	A-D-E-C	1	A-D	2	A-D-E	4	A-D-E-F

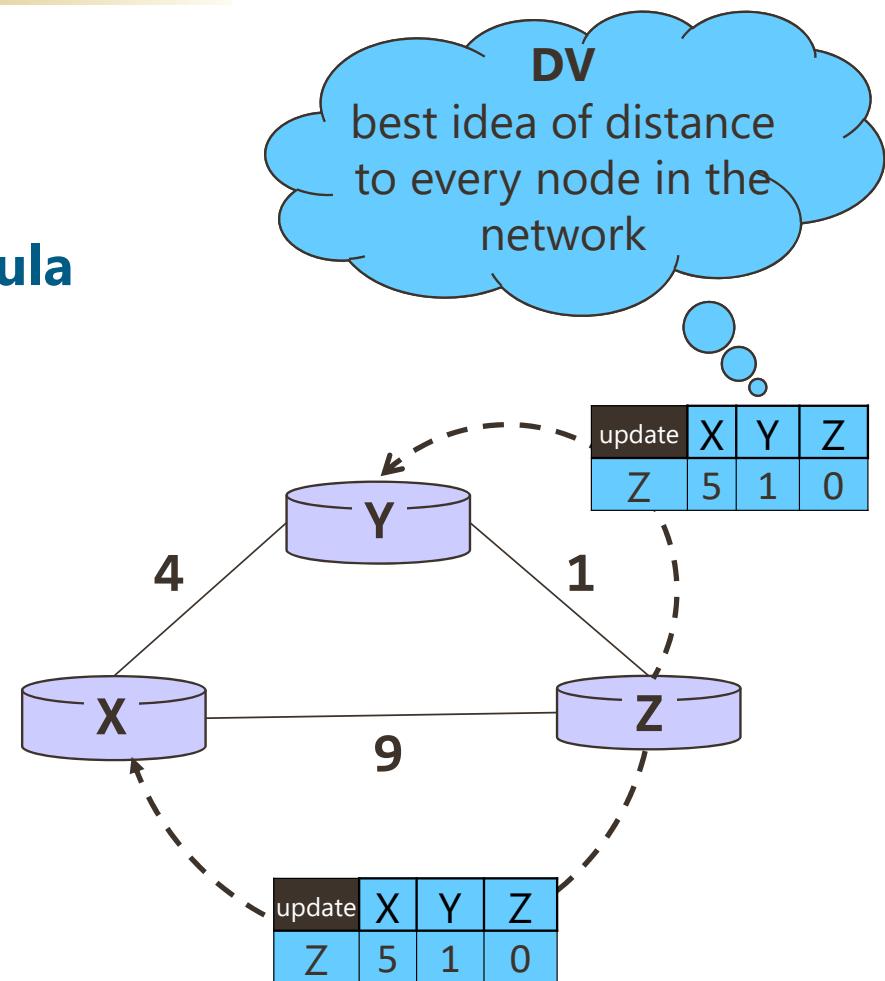
Distant Vector Routing

▪ Basic Idea

- 1) Send DV (Distant Vector) to neighbors
- 2) Update DV using **B-F (Bellman-Ford) Formula**

$$D_x(y) = \min_v \{C(x, v) + D_v(y)\}, \forall y \in \text{Nodes}$$

Least cost ($x \rightarrow y$) cost ($x \rightarrow v$)



Bellman Ford Algorithm

This implementation takes in a graph, represented as lists of vertices and edges, and fills two arrays (distance and predecessor) about the shortest path from the source to each vertex

```

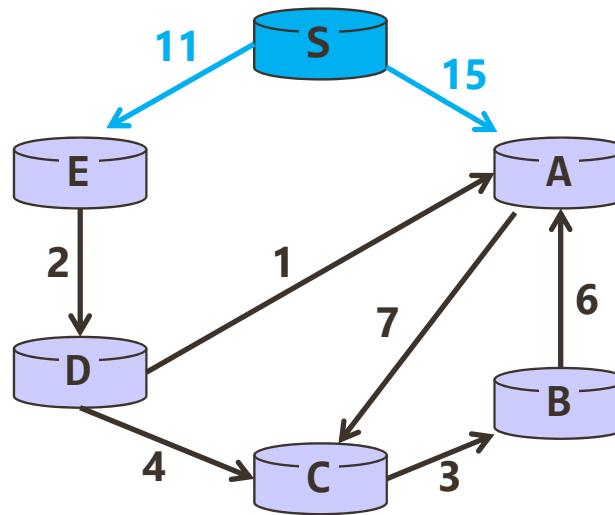
1 Initialization:
2 Input: Directed graph  $G=(V, E)$ ;  

   Edge lengths  $\{l_e : e \in E\}$ 
3 Output: reachable from  $s$ ,  $\text{dist}(u)$  is set to the  

   distance from  $s$  to  $u$ 
4
5 for all  $u \in V$ :
6    $\text{dist}(u) = \infty$ 
7    $\text{prev}(u) = \text{nil}$ 
8
9    $\text{dist}(s) = 0$ 
10 repeat  $|V| - 1$  times:
11   for all  $u \in V$ :
12     update( $e$ )
13
14 function update( $(u, v) \in E$ )
15    $\text{dist}(v) = \min\{\text{dist}(v), \text{dist}(u) + l(u, v)\}$ 
16    $\text{prev}(v) = u$  // conditionally, keep track of the predecessor

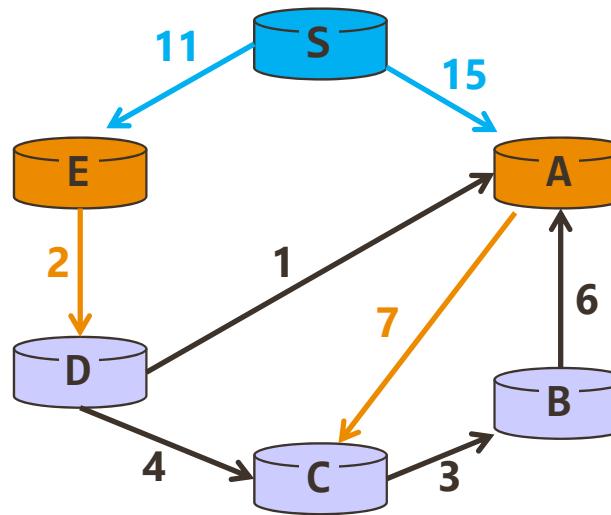
```

BF – 1st Hop



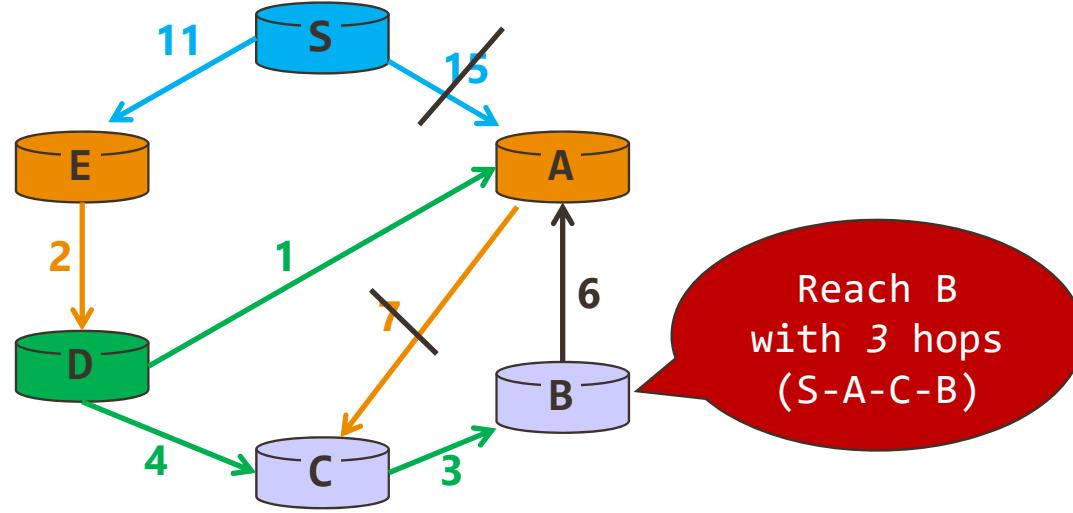
#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E

BF – 2nd Hop



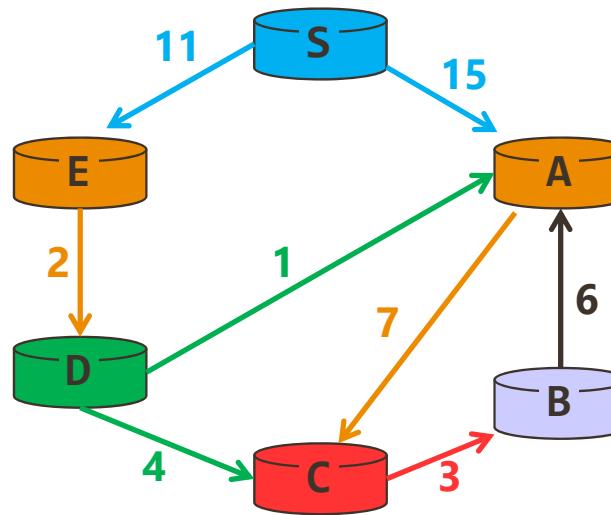
#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E
2	0	15	S-A	∞	-	22	S-A-C	13	S-E-D	11	S-E

BF – 3rd Hop



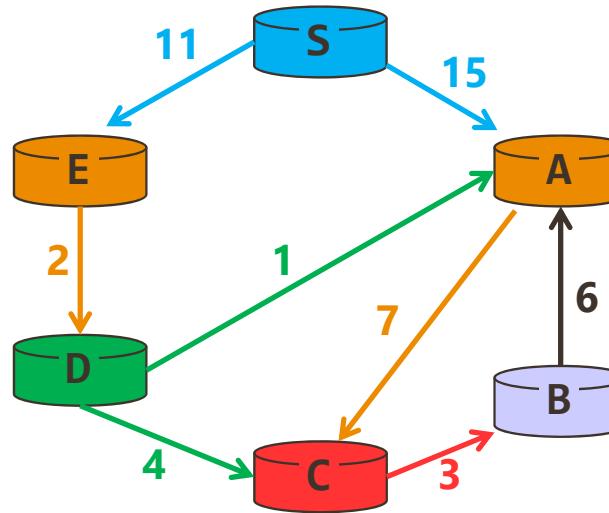
#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E
2	0	15	S-A	∞	-	22	S-A-C	13	S-E-D	11	S-E
3	0	15 14	S-A S-E-D-A	25	S-A-C-B	22 17	S-A-C S-E-D-C	13	S-E-D	11	S-E

BF – 4th Hop



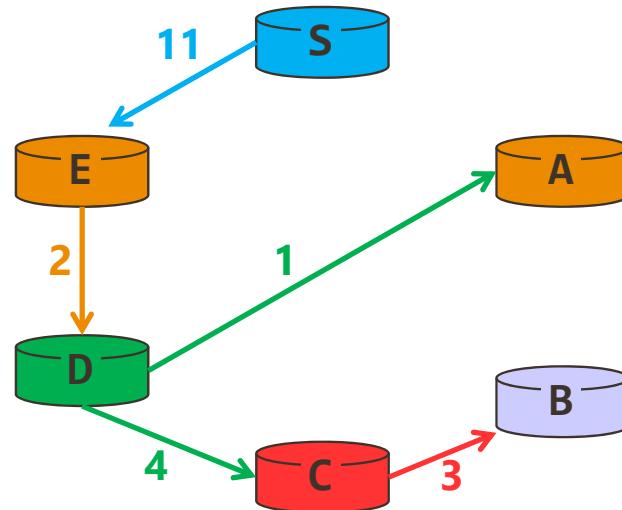
#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E
2	0	15	S-A	∞	-	22	S-A-C	13	S-E-D	11	S-E
3	0	14	S-E-D-A	25	S-A-C-B	17	S-E-D-C	13	S-E-D	11	S-E
4	0	14	S-E-D-A	25	S-A-C-B	17	S-E-D-C	13	S-E-D	11	S-E
				20	S-E-D-C-B						

BF – 5th Hop

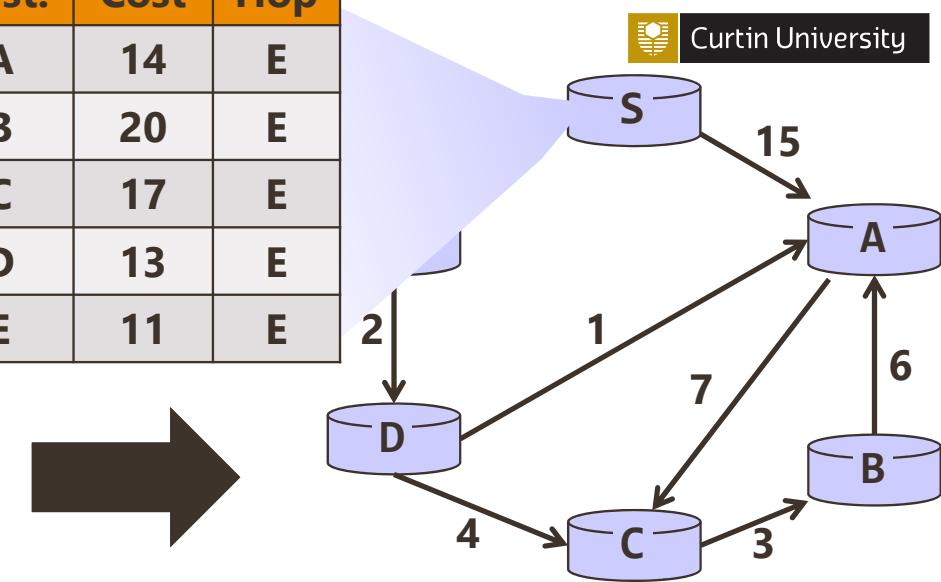


#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E
2	0	15	S-A	∞	-	22	S-A-C	13	S-E-D	11	S-E
3	0	14	S-E-D-A	25	S-A-C-B	17	S-E-D-C	13	S-E-D	11	S-E
4	0	14	S-E-D-A	20	S-E-D-C-B	17	S-E-D-C	13	S-E-D	11	S-E
5	0	14	S-E-D-A	20	S-E-D-C-B	17	S-E-D-C	13	S-E-D	11	S-E

Bellman Ford Algorithm



Dest.	Cost	Hop
A	14	E
B	20	E
C	17	E
D	13	E
E	11	E



#	S	L(A)	Path	L(B)	Path	L(C)	Path	L(D)	Path	L(E)	Path
1	0	15	S-A	∞	-	∞	-	∞	-	11	S-E
2	0	15	S-A	∞	-	22	S-A-C	13	S-E-D	11	S-E
3	0	14	S-E-D-A	25	S-A-C-B	17	S-E-D-C	13	S-E-D	11	S-E
4	0	14	S-E-D-A	20	S-E-D-C-B	17	S-E-D-C	13	S-E-D	11	S-E
5	0	14	S-E-D-A	20	S-E-D-C-B	17	S-E-D-C	13	S-E-D	11	S-E

Distance Vector

Node X	X	Y	Z
X	0 - 4 - 9 -		
Y	∞	∞	∞
Z	∞	∞	∞

Node X	X	Y	Z
X	0 - 4 - 5 Y		
Y	4	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 - 4 - 5 Y		
Y	4	0	1
Z	5	1	0

Node Y	X	Y	Z
X	∞	∞	∞
Y	4 - 0 - 1 -		
Z	∞	∞	∞

Node Y	X	Y	Z
X	0	4	9
Y	4 - 0 - 1 -		
Z	9	1	0

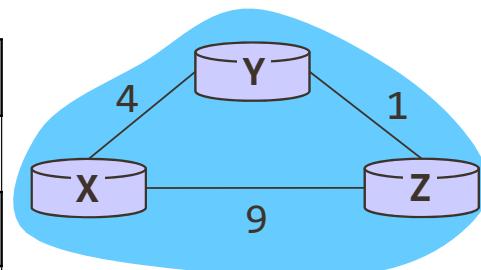
Node Y	X	Y	Z
X	0	4	5
Y	4 - 0 - 1 -		
Z	5	1	0

Node Z	X	Y	Z
X	∞	∞	∞
Y	∞	∞	∞
Z	9 - 1 - 0 -		

Node Z	X	Y	Z
X	0	4	9
Y	4	0	1
Z	5 Y	1 - 0 -	

Node Z	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5 Y	1 - 0 -	

$$\text{dist}(v) = \min\{\text{dist}(v), \text{dist}(u) + l(u, v)\}$$



Distance Vector: Points to Note

- If no topology change, **convergence in a few rounds**
 - ✓ After one message exchange, node knows about nodes two hops away
 - ✓ After two message exchange, node knows about nodes three hops away
- **No node has global knowledge**
- **Fully distributed**, yet maintains correct view

Distance Vector: Updates

▪ Triggered Updates

Send whenever the DV changes

Link/Node failure
or cost changes

▪ Periodic Updates

Sent even when no change in routing table

- ✓ To tell others that “**I am still alive**”
- ✓ To update others’ DV in case some **route** becomes **invalid**



Good news travels fast ! ≡



Bad news travels slow !

Distance Vector

Node X	X	Y	Z
X	0 - 1 - 5 Y		
Y			
Z			

Node X	X	Y	Z
X	0 - 1 - 2 Y		
Y	1 0 1		
Z			

Node X	X	Y	Z
X	0 - 1 - 2 Y		
Y	1 0 1		
Z	2 1 0		

Node Y	X	Y	Z
X			
Y	1 - 0 - 1 -		
Z			

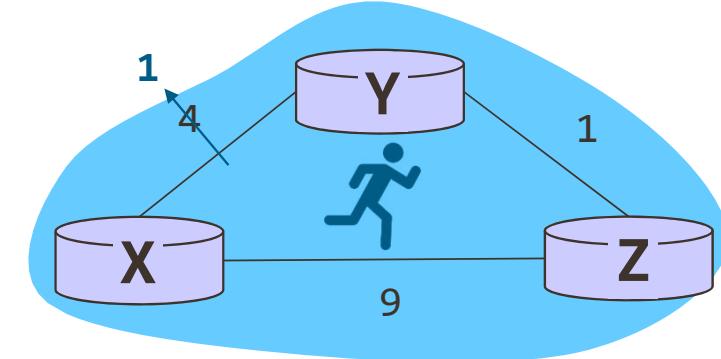
Node Y	X	Y	Z
X	0 1 5		
Y	1 - 0 - 1 -		
Z			

Node Y	X	Y	Z
X	0 1 2		
Y	1 - 0 - 1 -		
Z	2 1 0		

Node Z	X	Y	Z
X			
Y			
Z	5 Y 1 - 0 -		

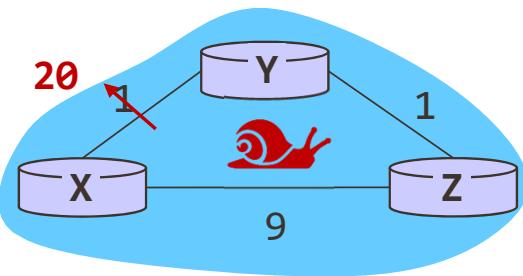
Node Z	X	Y	Z
X	0 1 5		
Y	1 0 1		
Z	2 Y 1 - 0 -		

Node Z	X	Y	Z
X	0 1 2		
Y	1 0 1		
Z	2 Y 1 - 0 -		



Good news travels fast !

Distance Vector



Bad news travels slow !

Full Story

Node X	X	Y	Z
X	0 -	20 -	2 Y
Y			
Z			

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	20	0	1
Z	2	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	3	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	10	0	1
Z	4	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	5	0	1
Z	9	1	0

Node Y	X	Y	Z
X			
Y	20 -	0 -	1 -
Z			

Node Y	X	Y	Z
X	0	20	2
Y	3 Z	0 -	1 -
Z	2	1	0

Node Y	X	Y	Z
X	0	10	9
Y	10 Z	0 -	1 -
Z	9	1	0

Node Y	X	Y	Z
X	0	10	9
Y	5 Z	0 -	1 -
Z	4	1	0

Node Y	X	Y	Z
X	0	10	9
Y	10 Z	0 -	1 -
Z	9	1	0

Node Z	X	Y	Z
X			
Y			
Z	2 Y	1 -	0 -

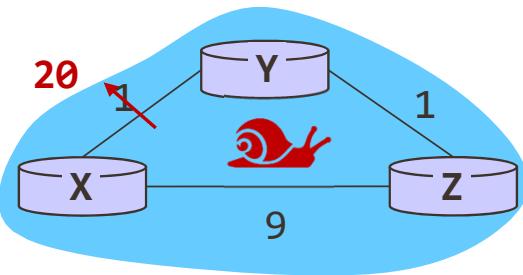
Node Z	X	Y	Z
X	0	20	2
Y	20	0	1
Z	9 -	1 -	0 -

Node Z	X	Y	Z
--------	---	---	---

Node Z	X	Y	Z
--------	---	---	---

Node Z	X	Y	Z
--------	---	---	---

Distance Vector



Full Story – cont

Bad news travels slow !

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	10	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	10	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	10	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 -	10 Z	9 -
Y	10	0	1
Z	9	1	0

Node X	X	Y	Z
X	0 -	10 Z	9
Y	10	0	1
Z	9	1	0

Node Y	X	Y	Z
X	0	10	9
Y	7 Z	0 -	1 -
Z	6	1	0

Node Y	X	Y	Z
X	0	10	9
Y	10 Z	0 -	1 -
Z	9	1	0

Node Y	X	Y	Z
X	0	10	9
Y	9 Z	0 -	1 -
Z	8	1	0

Node Y	X	Y	Z
X	0	10	9
Y	10 Z	0 -	1 -
Z	9	1	0

Node Y	X	Y	Z
X	0	10	9
Y	10 Z	0 -	1
Z	9	1	0

Node Z	X	Y	Z
X	0	10	9
Y	10	0	1
Z	9 -	1 -	0 -

Node Z	X	Y	Z
X	0	10	9
Y	7	0	1
Z	8 Y	1 -	0 -

Node Z	X	Y	Z
--------	---	---	---

Node Z	X	Y	Z
--------	---	---	---

Node Z	X	Y	Z
--------	---	---	---

Basis	Link State	Distant Vector
Summary	"Tell the world about neighbors"	"Tell the neighbors about the world"
Updates	Triggered/Periodic Updates	Periodic Updates



Intra-Domain Routing Protocols

- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)

Combination of rules and procedures that let routers inform each other of changes

Routing Protocols

Intra-domain: works only within domains
Inter-domain: works within and between domains

Intradomain

Interdomain

Routing Algorithms

Distant Vector

Link State

Path Vector

Routing Protocols

RIPv1/v2

IGRP

OSPF

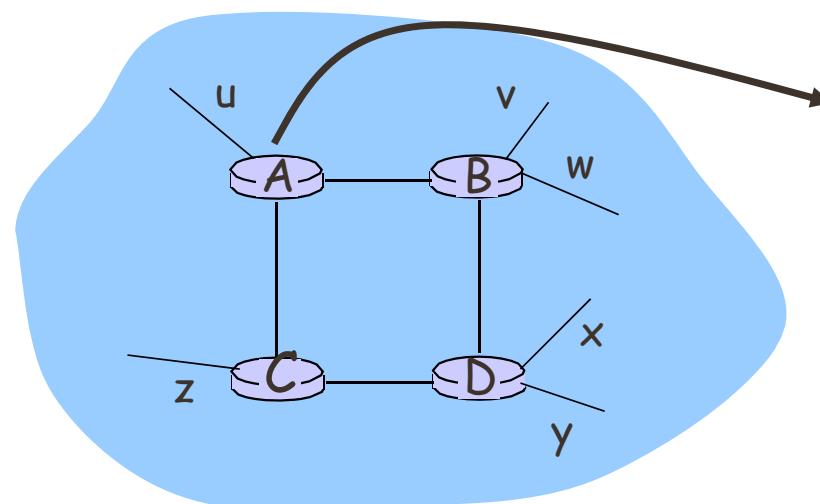
BGP

Intradomain Routing

- a.k.a **Interior Gateway Protocols (IGP)**
- Common Protocols:
 - **RIP:** Routing Information Protocol
 - **IGRP:** Interior Gateway Routing Protocol (Cisco proprietary)
 - **EIGRP:** Extended IGRP (Cisco proprietary)
 - **OSPF:** Open Shortest Path First

RIP: Routing Information Protocol

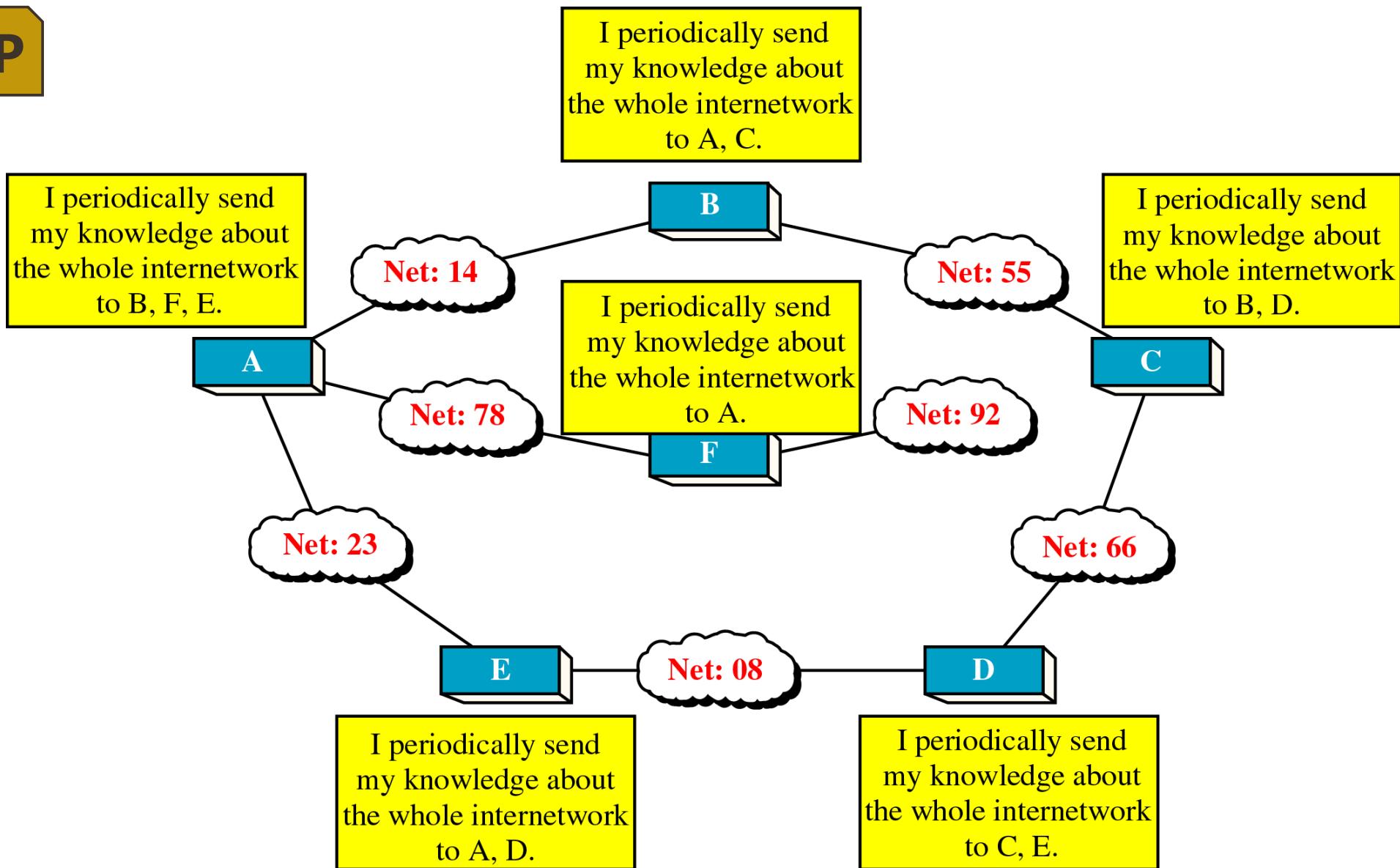
- Uses **Distance Vector** Algorithm
- **Distance Metric:** # of hops (max = 15 hops)
- **Distance Vectors** exchanged among neighbors every **30 sec**
 - via Response Message (also called **advertisement**)

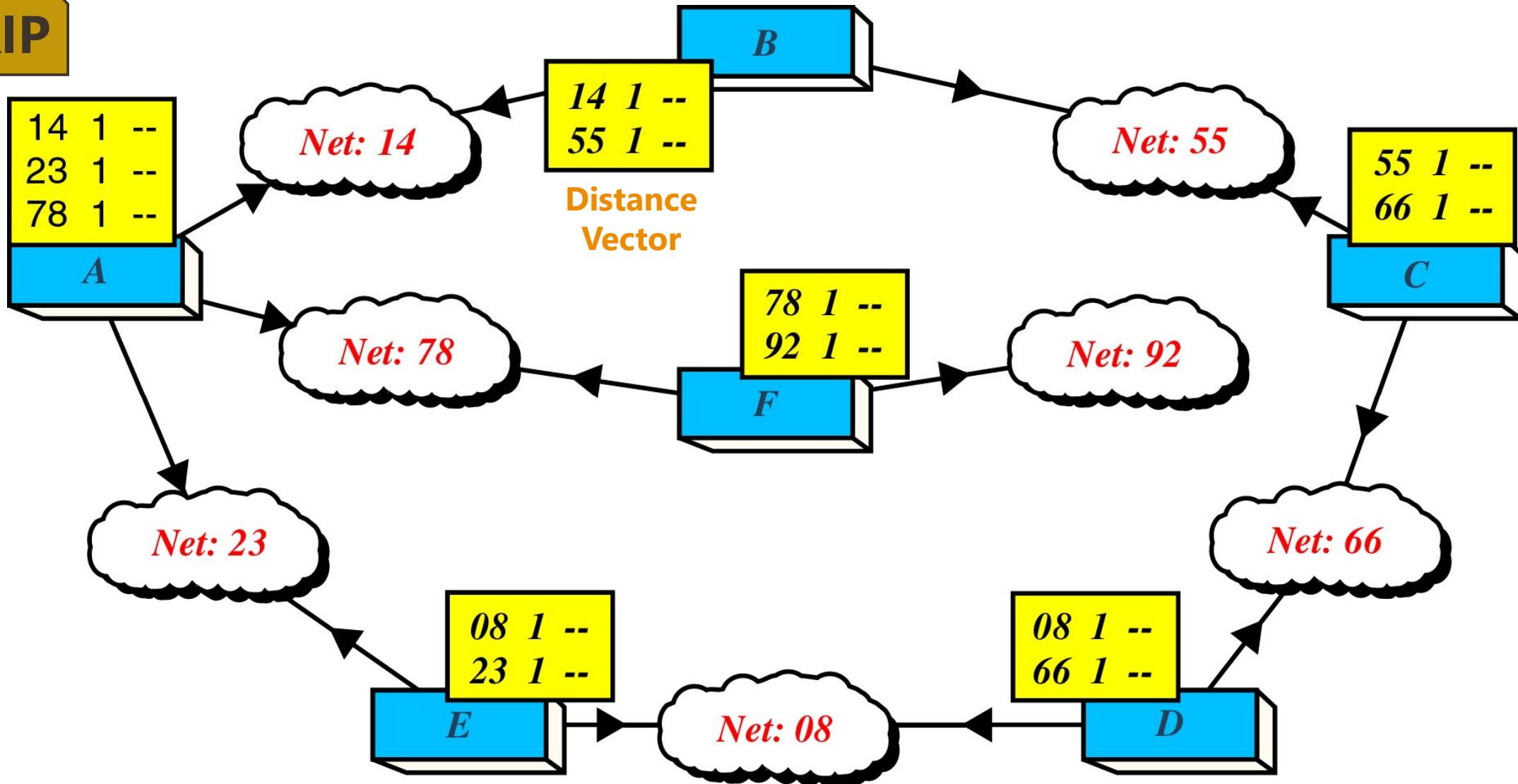


Destination	Cost (hops)	Next hop
U	1	-
V	2	B
W	2	B
X	3	B
Y	3	B
Z	2	C

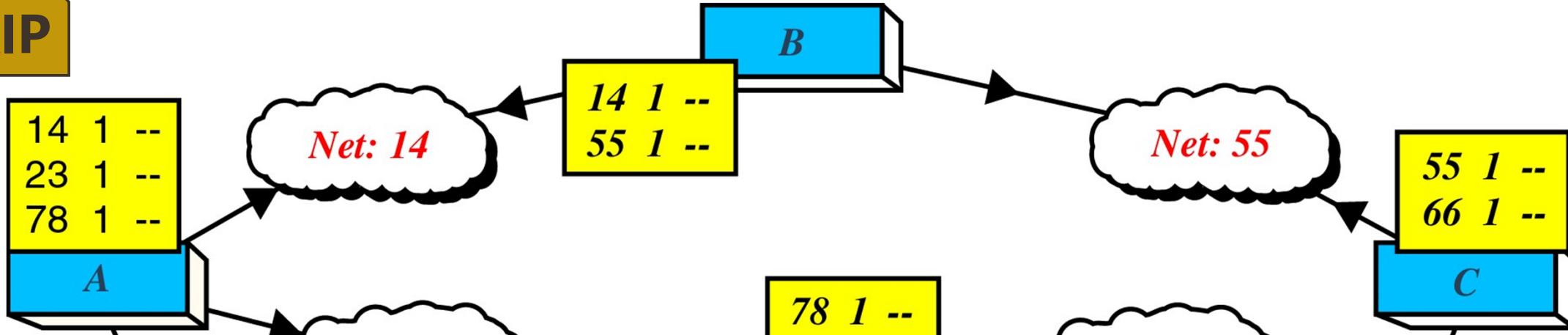
RIP

in University



RIP

RIP



A's old table

14	1	—
23	1	—
78	1	—

14	1
55	1

+

one hop

=

14	2	B
55	2	B

After adjustment

14	1	—
14	2	B
23	1	—
55	2	B
78	1	—

Combined

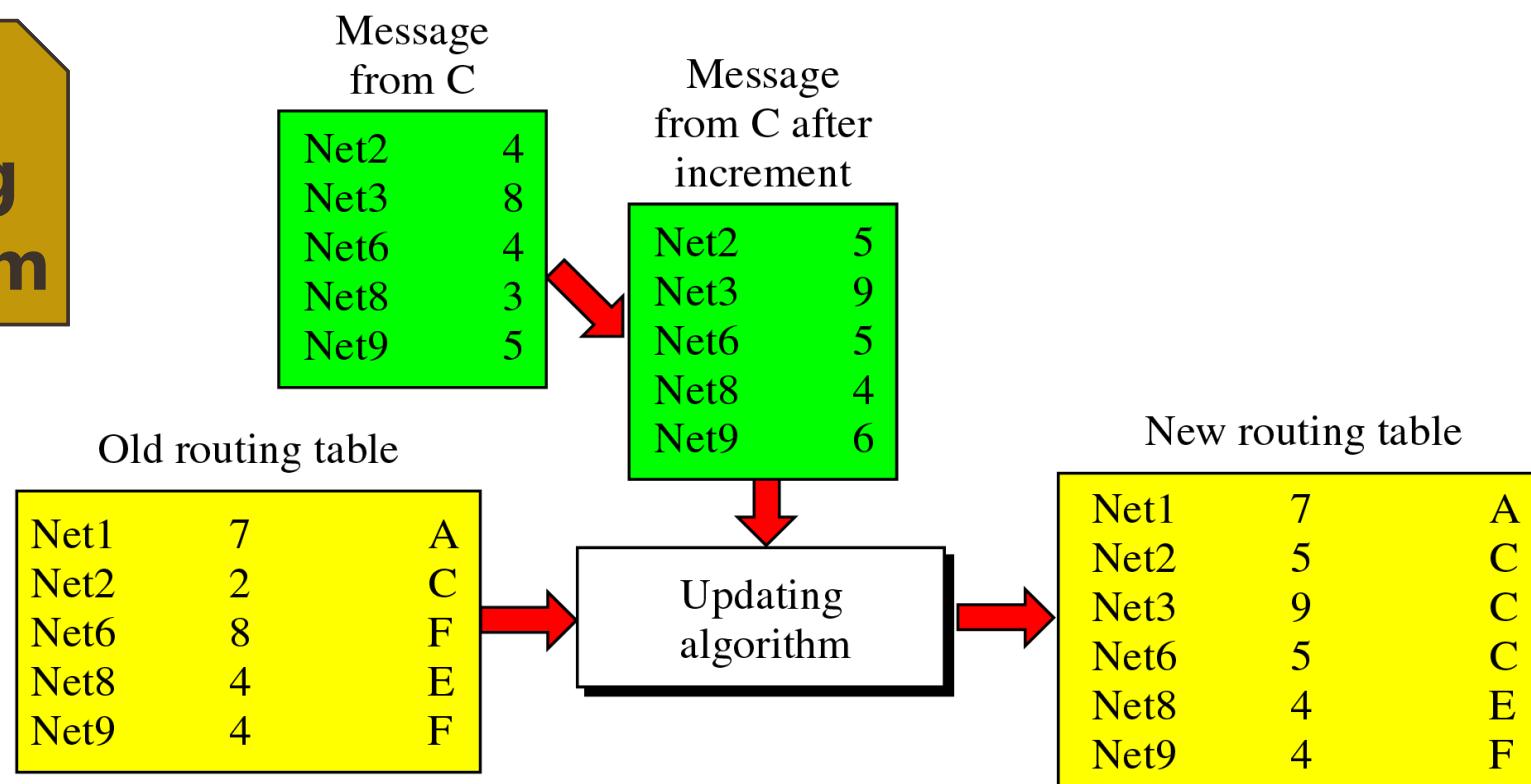
14	1	—
23	1	—
55	2	B
78	1	—

A's new table

RIP - Updating Algorithm

1. Router adds 1 hop to the hop count field for each advertised route
2. Add advertised destination if not in table
3. If destination is in the table
 - a. **Advertising Node == Next hop** field, replace the entry
 - b. **Advertising Node != Next hop** field
 - I. smaller hop count, replace the entry
 - II. Larger or equal hop count, do nothing

RIP - Updating Algorithm



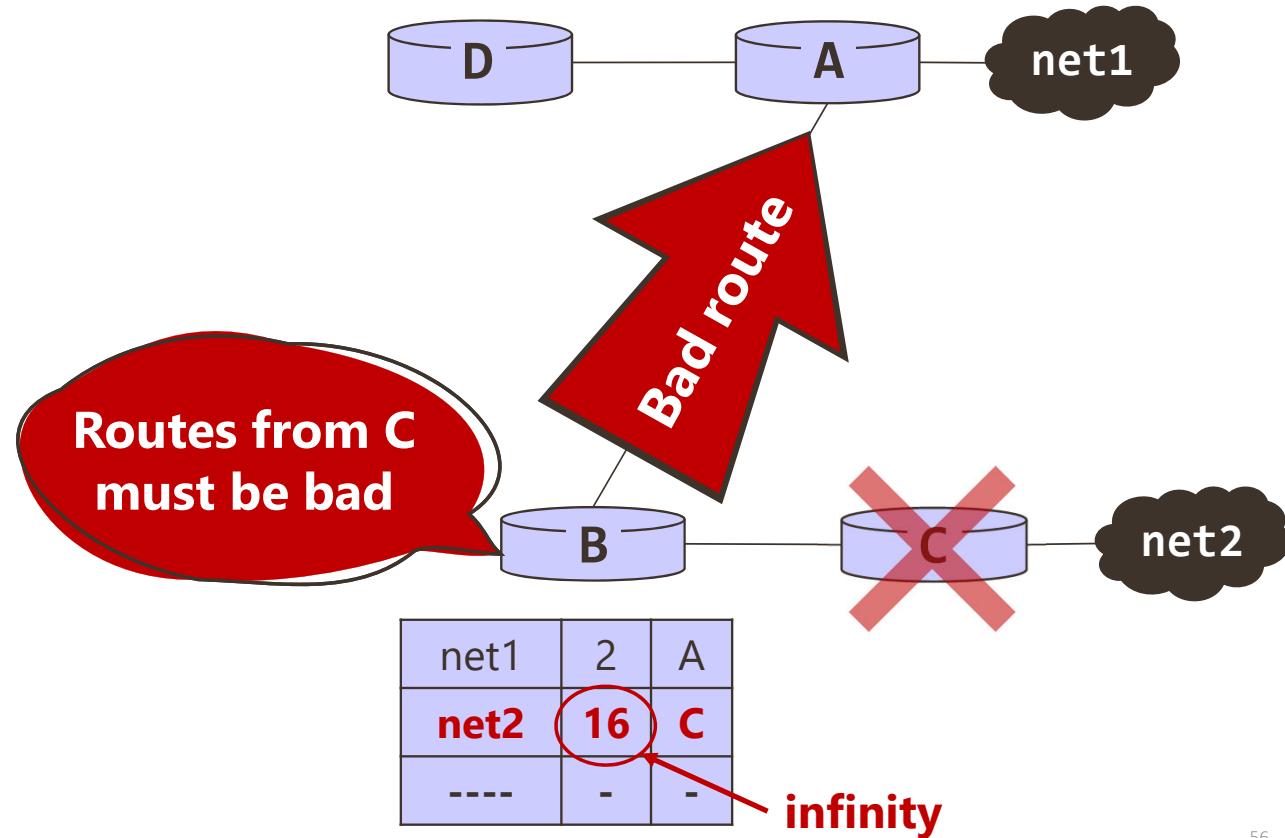
Rules

- Net2: Replace (**Rule 3.a**)
- Net3: Replace (**Rule 1**)
- Net6: Replace (**Rule 3.b.i**)
- Net8: Replace (**Rule 3.b.ii**)
- Net9: Replace (**Rule 3.b.ii**)

Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

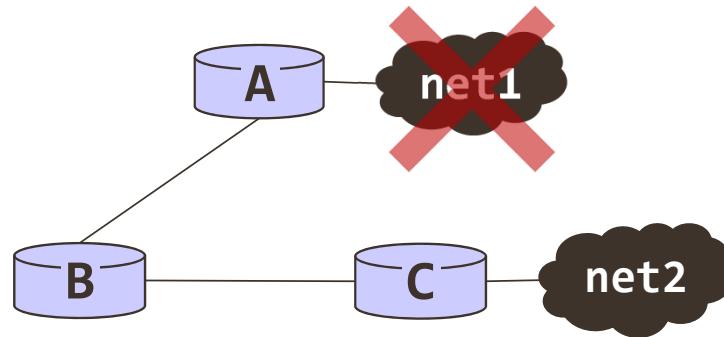
RIP - Route Poisoning

- **Infinity** -> 16 hops (unreachable)



RIP - Count to Infinity

Unstable routing
during this whole time



Node A	cost	hop
net1	16	-
net2	3	B
-----	-	-

+ 1	cost
net1	16
net2	3

Node A	cost	hop
net1	3	B
net2	3	B
-----	-	-

Node A	cost	hop
net1	16	B
net2	3	B
-----	-	-

Node A	cost	hop
net1	5	B
net2	3	B
-----	-	-

Node B	cost	hop
net1	2	A
net2	2	C
-----	-	-

+ 1	cost
net1	2
net2	2

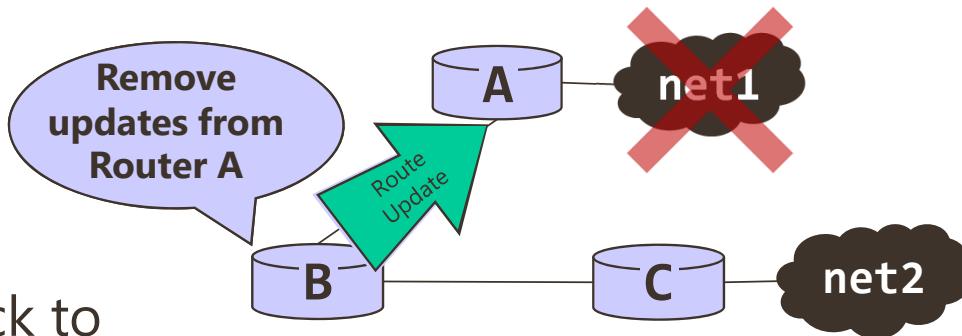
Node B	cost	hop
net1	16	A
net2	2	C
-----	-	-

Node B	cost	hop
net1	4	A
net2	2	C
-----	-	-

Node B	cost	hop
net1	16	A
net2	2	C
-----	-	-

RIP - Simple Split Horizon

Don't advertise routes back to the router you learn them from



Node A	cost	hop
net1	16	-
net2	3	B
-----	-	-

Node A	cost	hop
net1	16	-
net2	3	B
-----	-	-

Node A	cost	hop
net1	16	-
net2	3	B
-----	-	-

Node A	cost	hop
net1	16	-
net2	3	B
-----	-	-

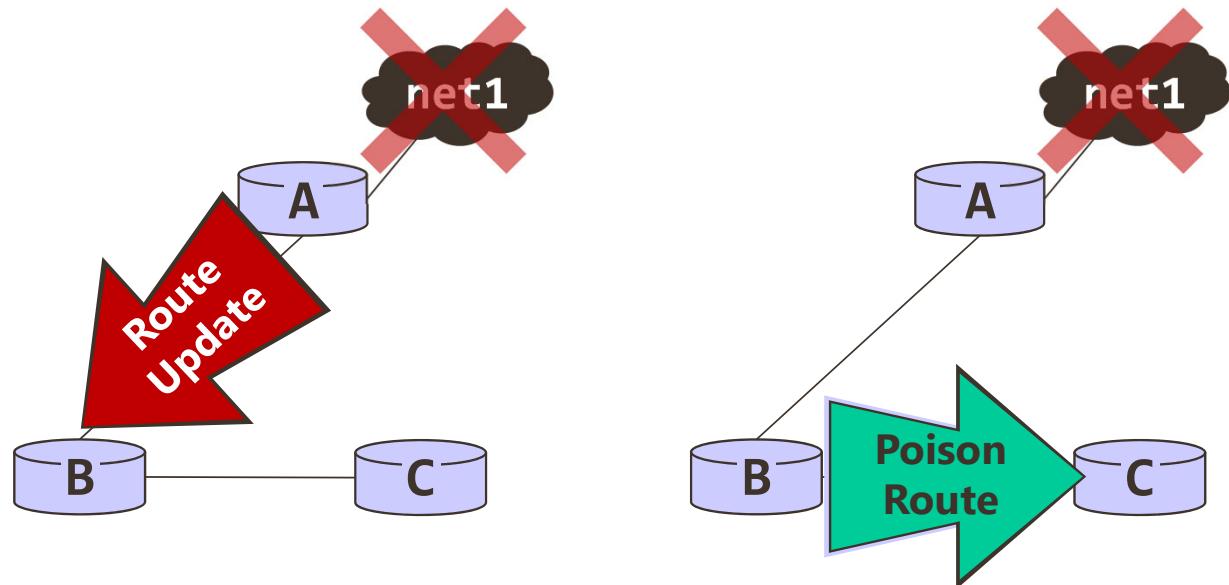
Node B	cost	hop
net1	2	A
net2	2	C
-----	-	-

Node B	cost	hop
net1	16	A
net2	2	C
-----	-	-

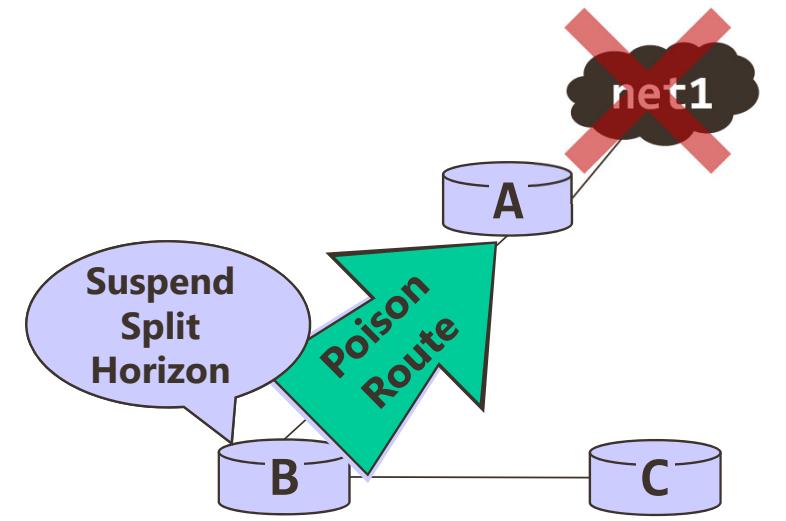
Node B	cost	hop
net1	16	A
net2	2	C
-----	-	-

Node B	cost	hop
net1	16	A
net2	2	C
-----	-	-

RIP - Split Horizon with Poison Reverse



Triggered Partial Update
Send only the info about bad route.



Reverse Route Poisoning
Send confirmation from B
(just the poison route)
immediately.



Redundancy In Networks?



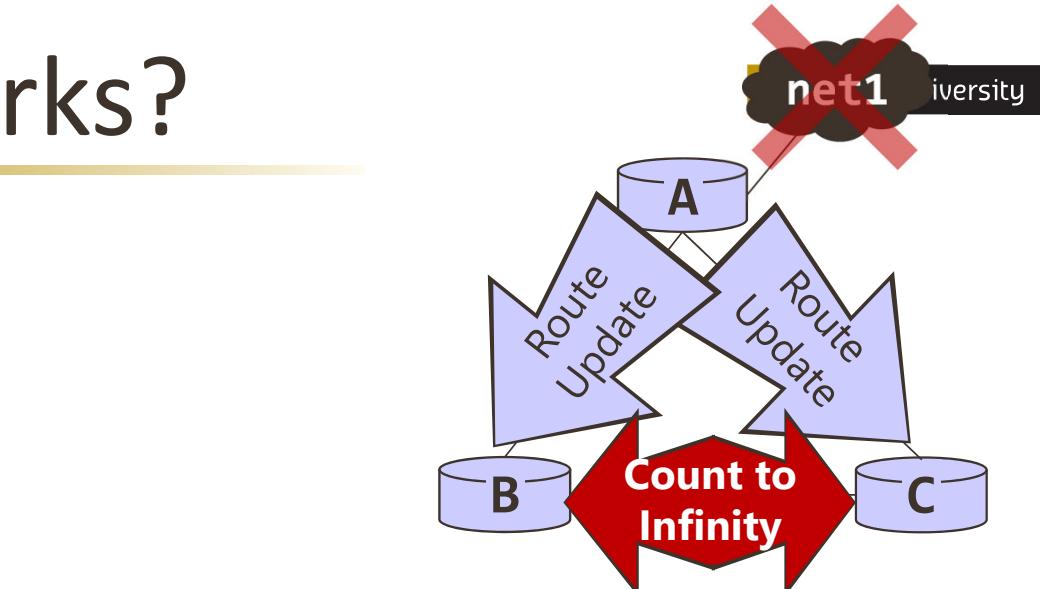
- **Split horizon does not work**



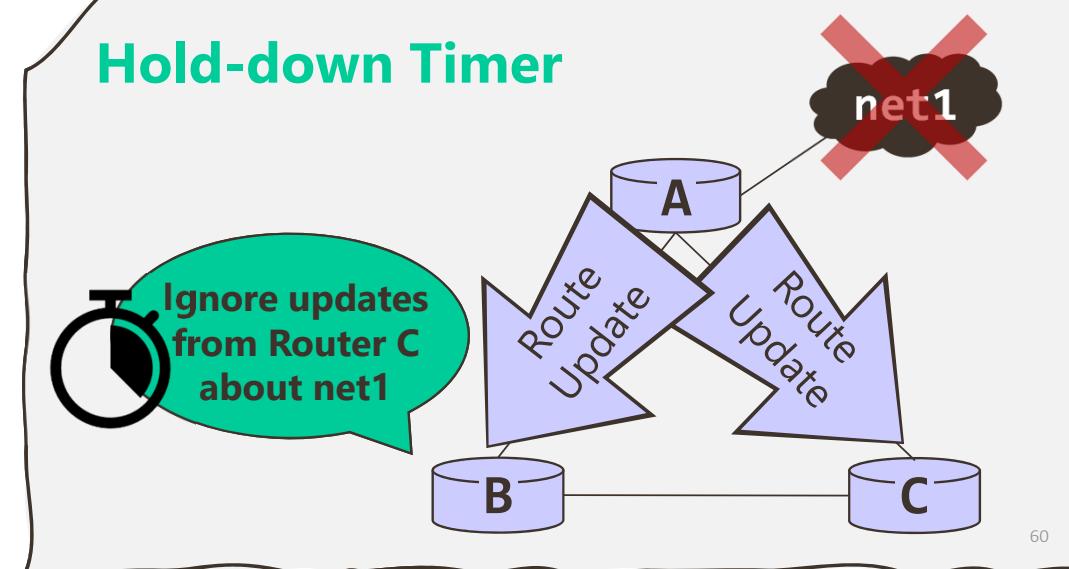
- **Hold-down Timer**



- ✓ Suppressing any routing update after a poison route is received
- ✓ Gives time to converge
- ✓ But, B and C will listen to new updates of net1 from A

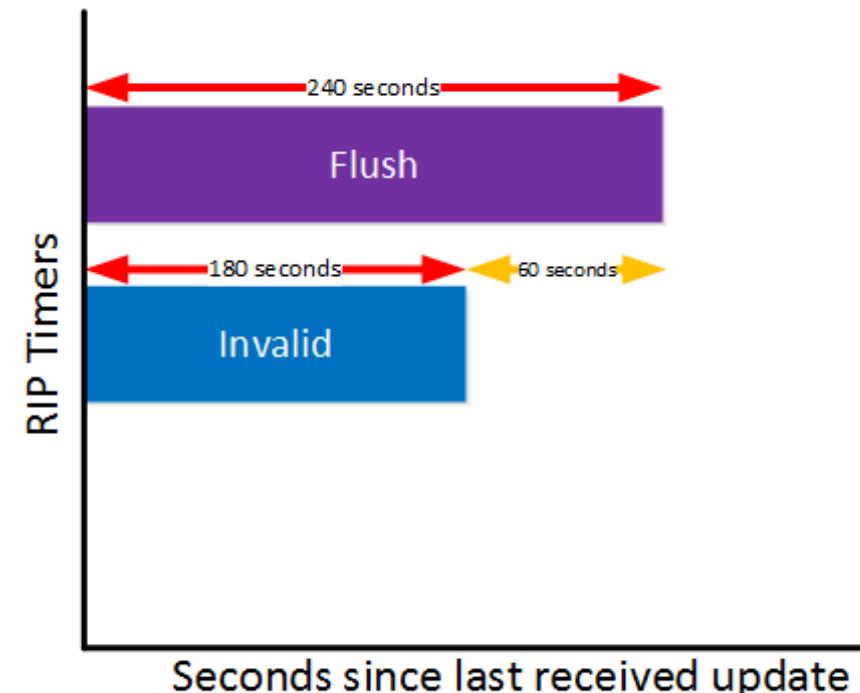


Hold-down Timer



RIP Timers

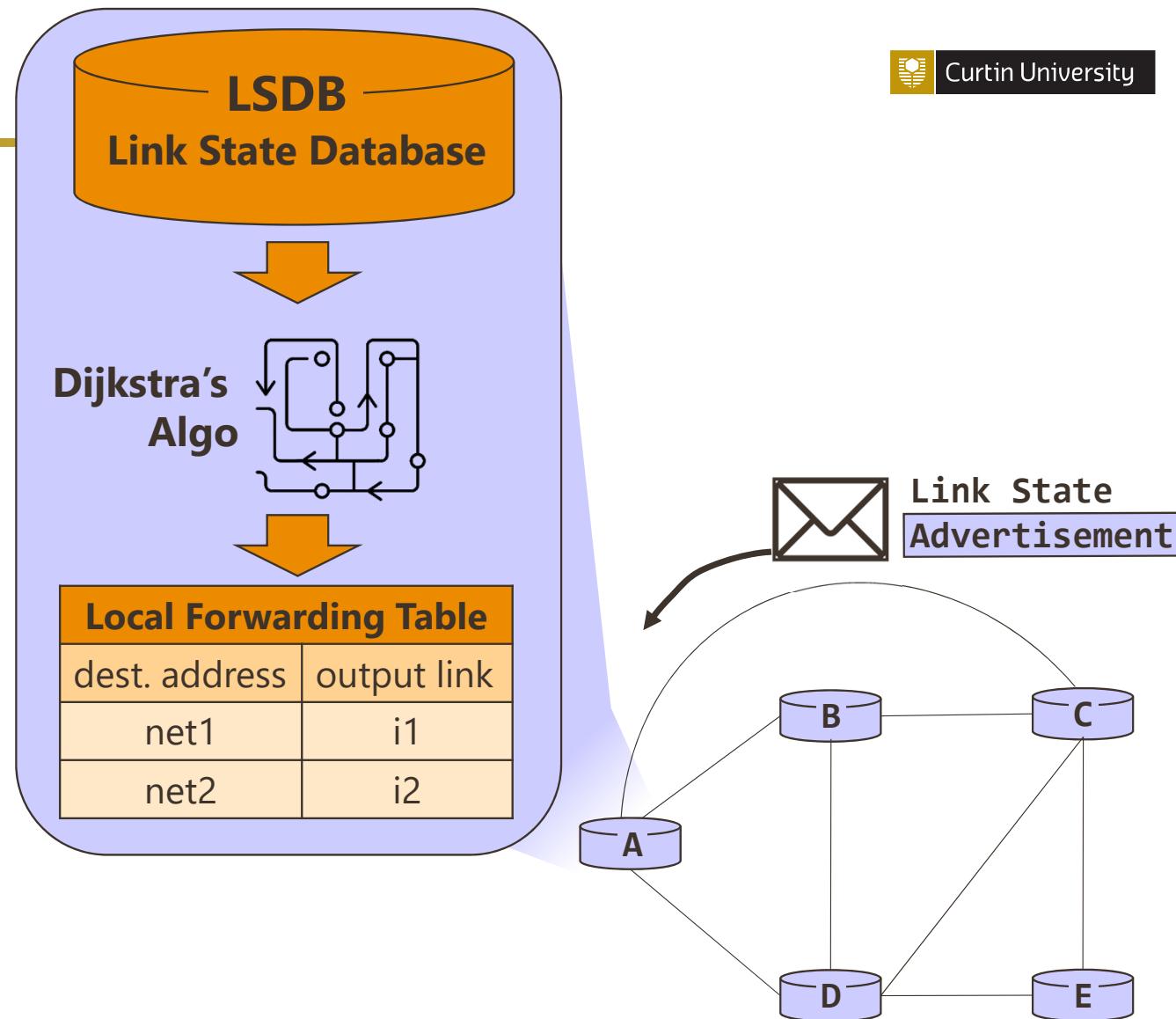
- **Update:** interval to send routing updates (distance vectors), *default: 30s*
- **Invalid:** time to wait since received the last valid update, *default: 180s*
- **Flush:** time to wait since received the last valid update before throwing a route away, *default: 240s*



OSPF

Open Shortest Path First

- ✓ Uses **linked state routing**
- ✓ Faster convergence than distance vectors
- ✓ Low bandwidth requirements
- ✓ Supports **CIDR, VLSM & authentication**
- ✓ Hierarchical design using “**areas**” for larger networks



OSPF 3 Step Process

OSPF States

1. Initializing Bi-directional Communication

- Two routers running OSPF on the same link initialize a bi-direction communication via exchanging HELLO messages



DOWN
INIT

2. Become Neighbors

- Two routers running OSPF on the same link **may not** form a neighbor relationship, **remain in TWO-WAY state**
- **Or,** Become fully adjacent and **EXCHANGE LSAs**



EXCHANGE
FULL

3. Choose the Best Routes

- Each router chooses the best routes running link-state algorithm on their local database

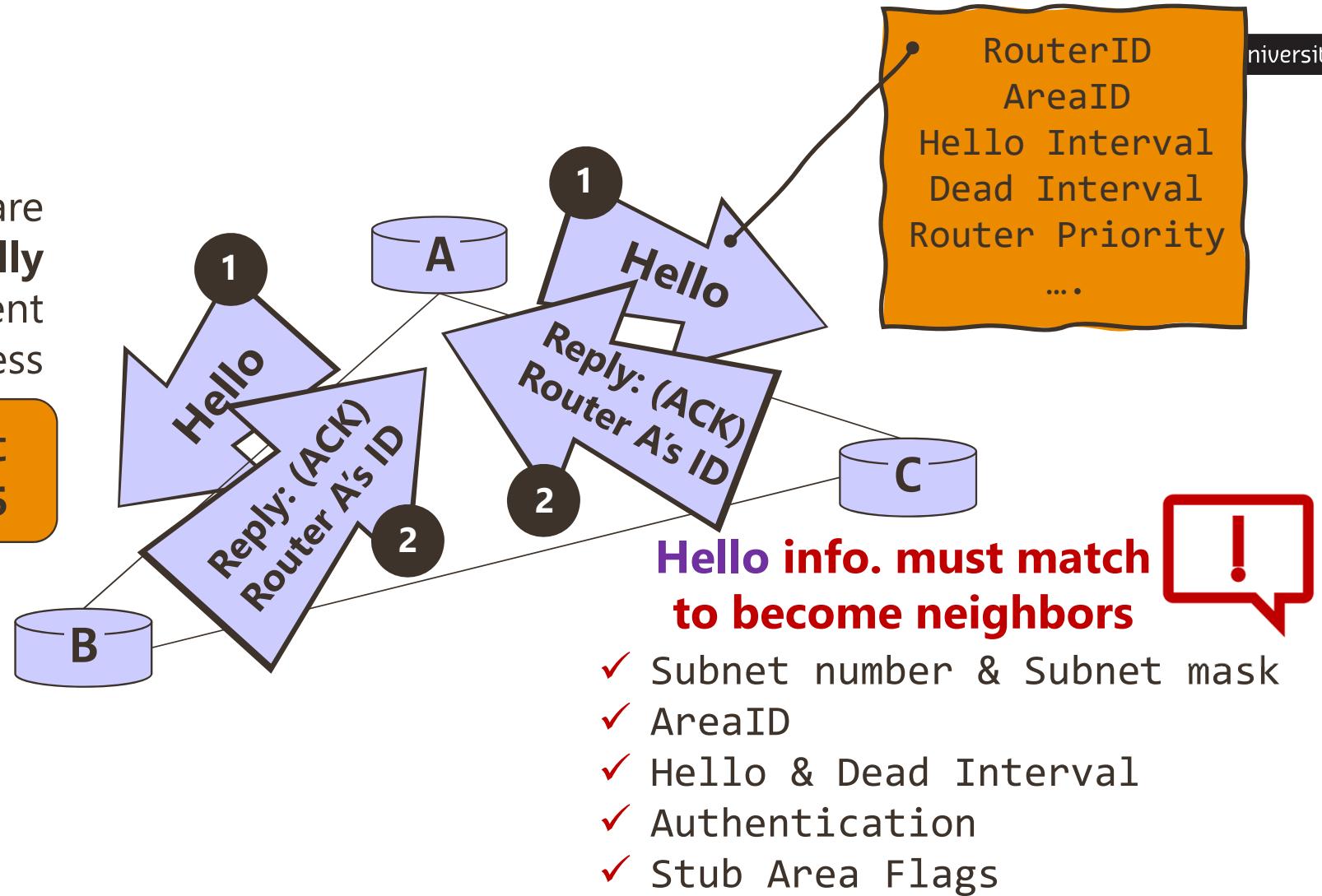


FULL

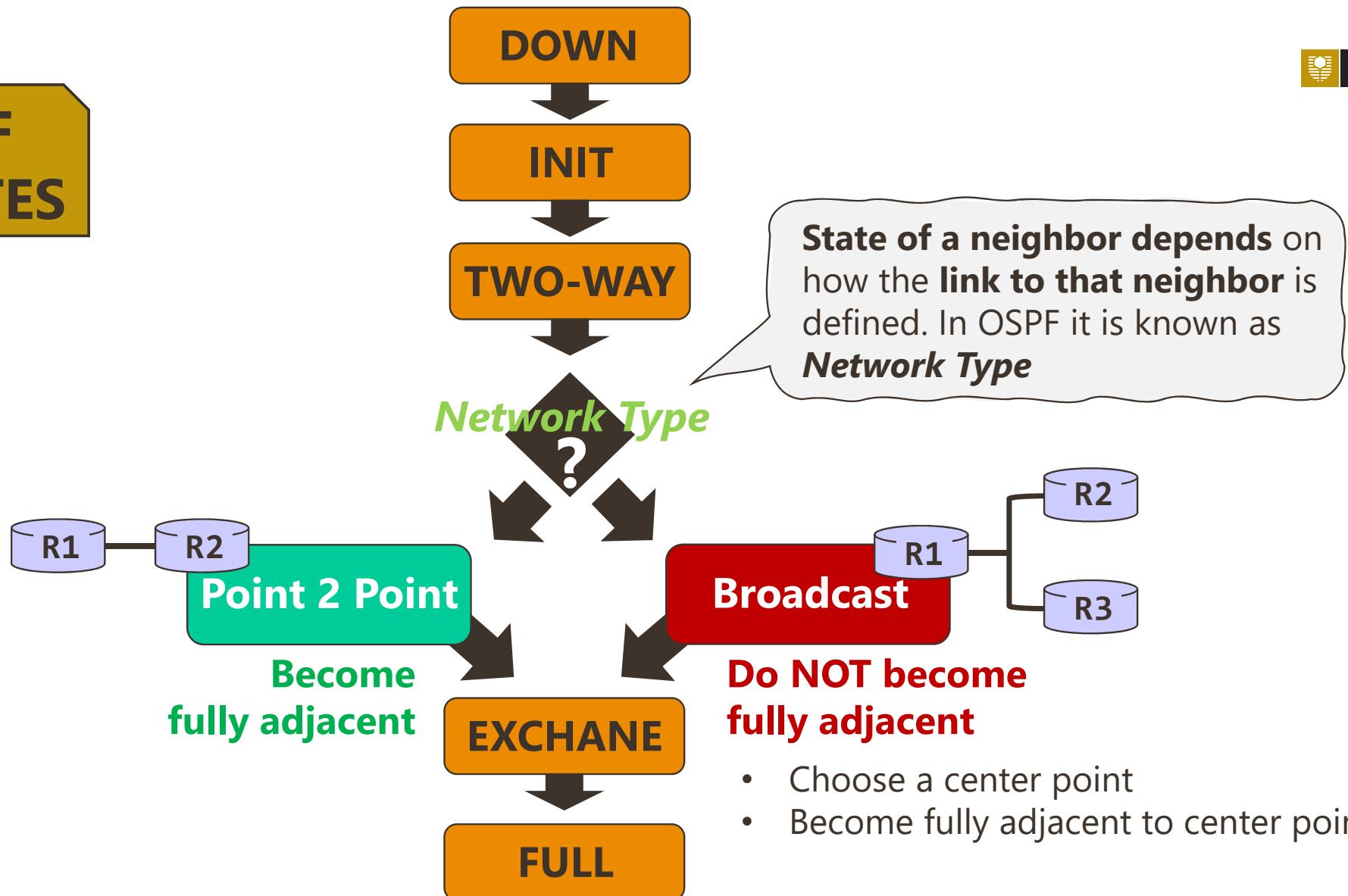
OSPF INIT

Neighbors are discovered dynamically using **hello packets** sent to multicast address

Multicast
224.0.0.5

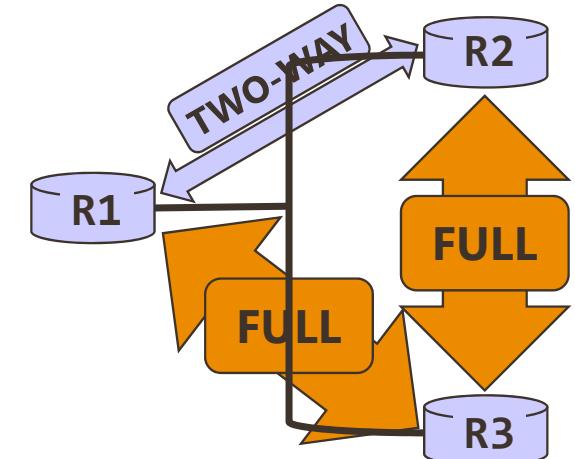


OSPF STATES



Network Type: **Broadcast**

- **Do NOT become fully adjacent to each other**
- **Choose** a center point (**Designated Router - DR**)
 - ✓ based on HELLO message's **Router Priority** and/or **RouterID** fields
- Each router in the broadcast network segment will **become fully adjacent only to Designated Router (DR)**
- A **Backup Designated Router (BDR)** is also chosen in case primary DR fails



R3 will relay Information between R1 and R2 since R1, R2 don't communicate directly

Network Type:

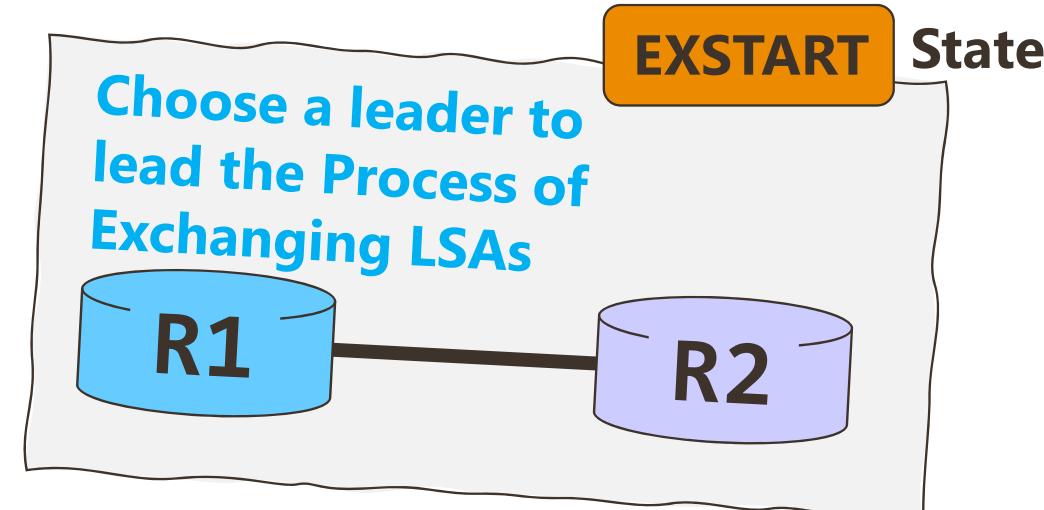
Point 2 Point

OR

Broadcast

In EXCHANGE State

- Exchange database description (summary of LSDB, not entire LSDB)
- Request part of the LSAs
- Once both have the same LSDB



FULL State

- Use Dijkstra's Algorithm on LSDB -> Best Routes
- Best Routes -> Routing Table

IN

FULL

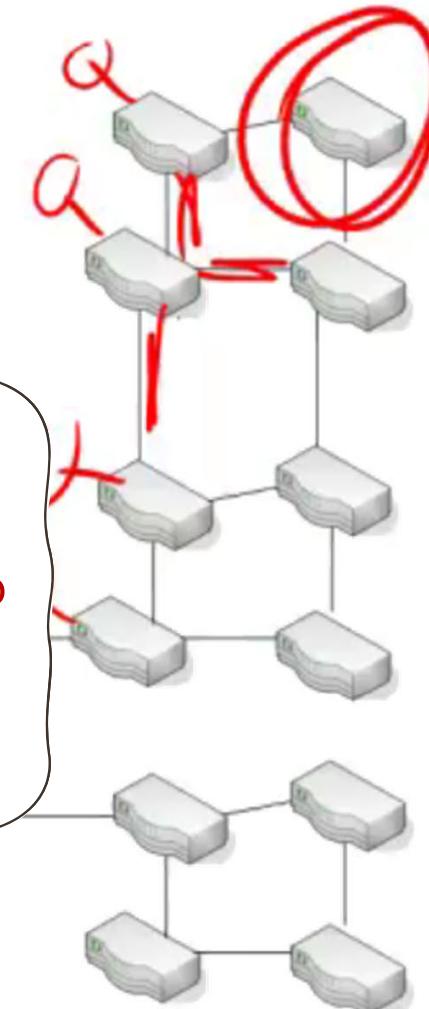
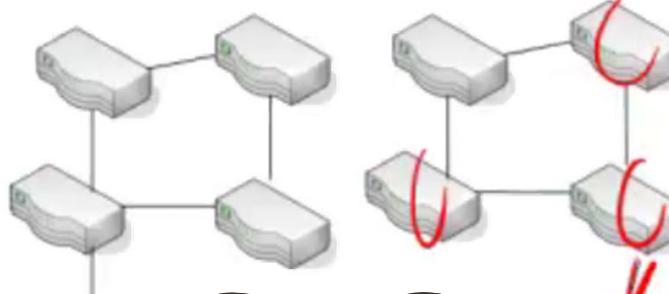
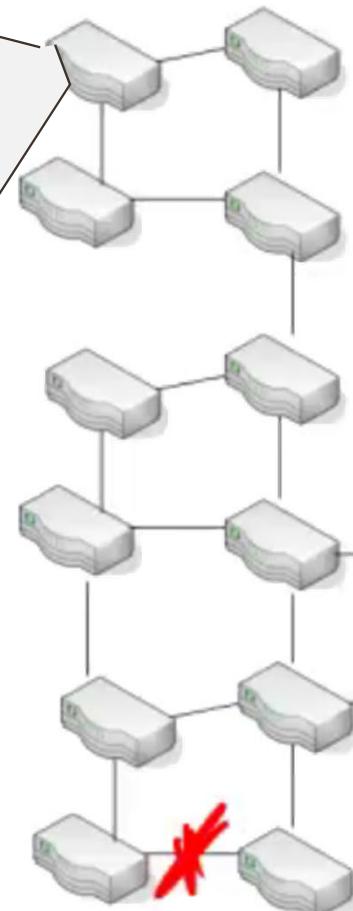
STATE

- Continue to exchange hello packets (heartbeat) - **HELLO Interval**
- Declare dead if not heard in **Dead Interval**
- Continue to exchange **LSAs Every 30 min**
- **If network failure**, send updates *immediately*

OSPF Design

LSDB includes

Links
Routers
Subnets



Large Network ?

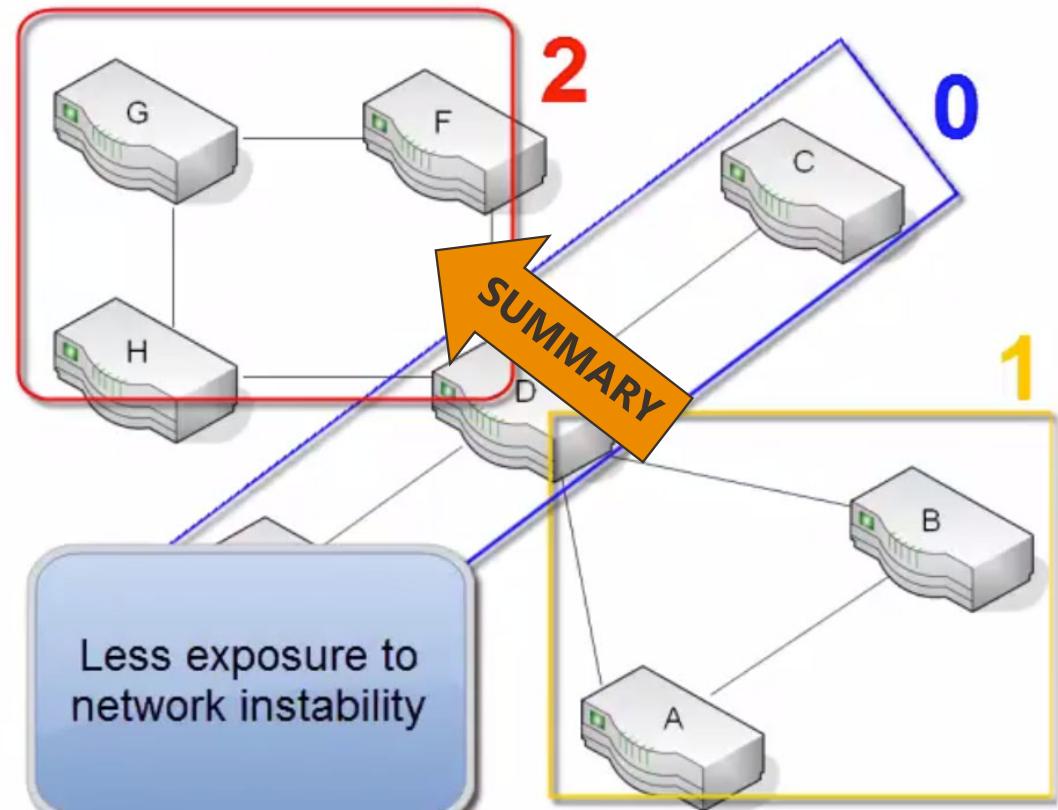
- Large LSDB
- All routers are exposed to all network instability
- Takes time to run Dijkstra in case one router fails

▪ Use the concept of Area

- Modular approach to split the network
- Area = Group of routers with same LSDB
- **Area 0 = Backbone Area**
- All other areas must connect via backbone

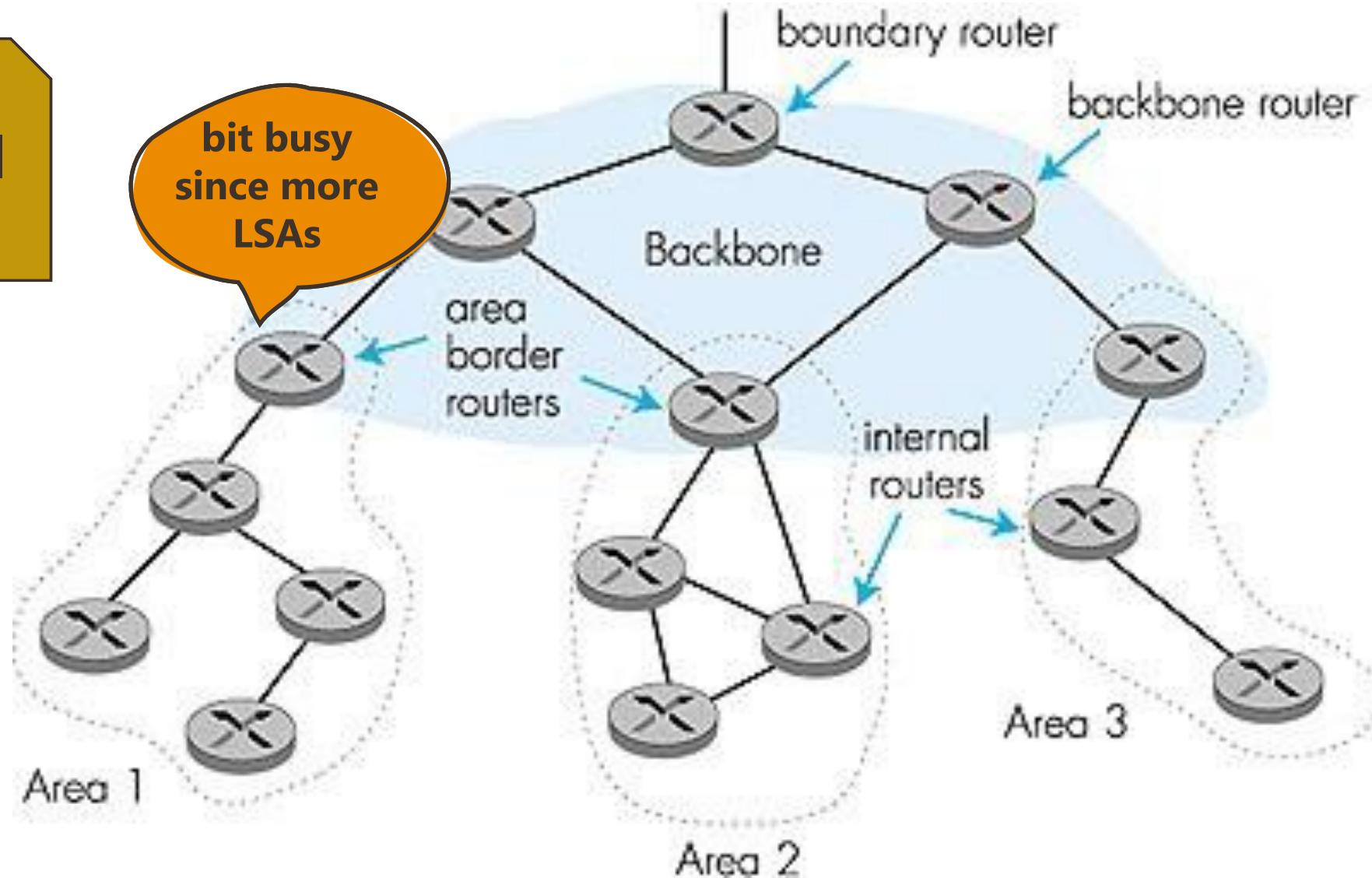
If Router A fails,

- Router B, D only will get LSAs
- Reapply Dijkstra Algorithm



- Router B is **only** fully aware of Router A, D
 - But it can still route to Router G
 - Router B has summary information of Area 2
 - Smaller LSDB

OSPF Hierarchical Design



Summary



RIP	OSPF
Distance Vector (DV) Routing	Link State Routing
Metric: Hop Count	Bandwidth, Delay
Best Path Calculation: Bellman Ford Formula	SPF (e.g. Dijkstra) Algorithm
Routing: Networks are not divided into areas or multiple tables	Routing: done with Autonomous Systems, Areas, Stub Areas and Backbone Areas
Maximum Hop Count: 15	No hop count



Inter-Domain Routing

- BGP (Border Gateway Protocol)
- IDRP (Inter-Domain Routing Protocol)

Routing Protocols

Combination of rules and procedures that let routers inform each other of changes

Intra-domain: works only within domains
Inter-domain: works within and between domains

Routing Algorithms

Distance Vector

Routing Protocols

RIPv1/v2

Intra-Domain

Inter-Domain

Link State

Path Vector

IGRP

OSPF

BGP

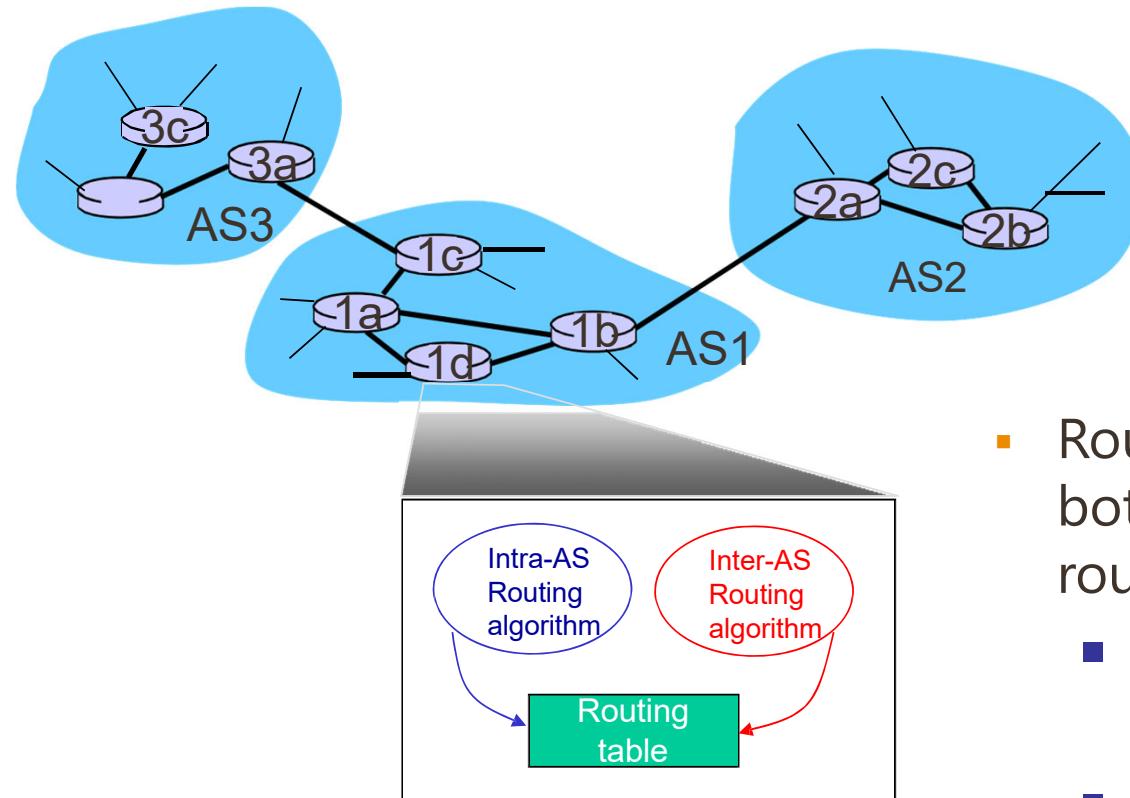
Hierarchical routing

- Aggregate routers into regions, “autonomous systems” (AS)
- Routers in the same AS run the same routing protocol
 - “intra-AS” routing protocol
 - routers in different AS can run different intra-AS routing protocol

Gateway router:

- at “edge” of its own AS
- has link to router in another AS

Interconnected AS's



- Routing table configured by both intra-AS and inter-AS routing algorithm
 - intra-AS sets entries for internal dests
 - inter-AS & intra-AS sets entries for external dests

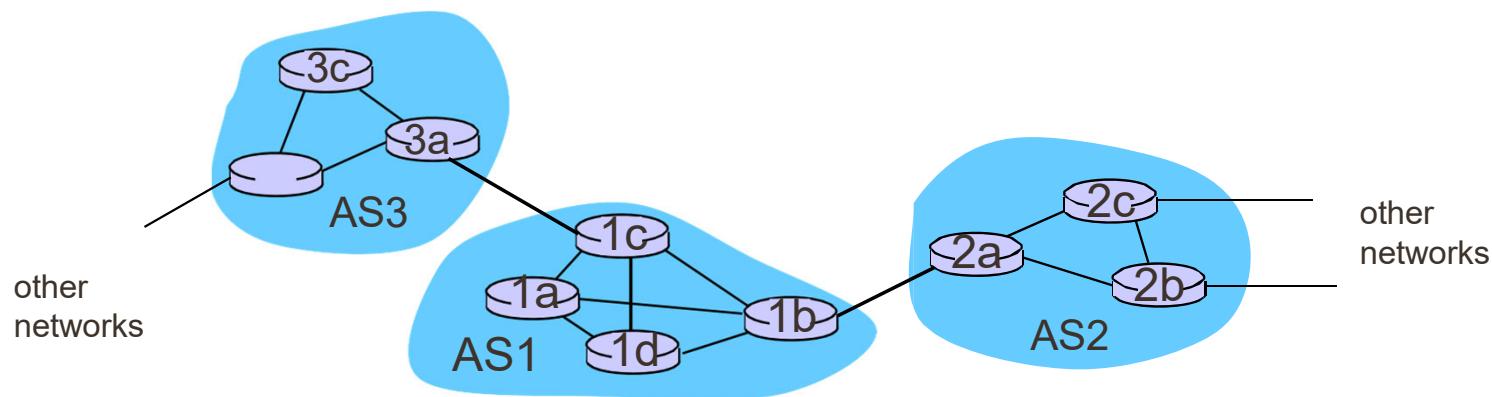
Inter-AS tasks

- Suppose router in AS1 receives datagram destined outside of AS1:
 - Router should forward packet to gateway router, but which one?

AS1 must:

1. learn which destinations are reachable through AS2, which through AS3
 2. propagate this reachability info to all routers in AS1

job of inter-AS routing!



Distant Vector and Link State

- distance-vector protocol is ineffective
 - Not all routers uses the same distance metric.
 - different AS's have different priorities and may have restrictions on other AS. No information is given about the AS that will be visited along a route.
- link-state is also unsuitable
 - flooding to all router across multiple AS's is unmanageable
 - metrics used in different AS's can be different

Path Vector Routing

- An alternative: path-vector routing
 - dispense with distance metrics.
 - Simply provide information about which network can be reached by a given router and the AS's that must be crossed to get there.
 - since a complete list of AS's traversed by a route is provided in the path vector, it allows policy routing.



Inter vs Intra Domain Routing

Policy:

- inter-AS: admin wants control over how its traffic is routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance

BGP(Border Gateway Protocol)

- The de facto inter-domain routing protocol
- BGP provides each AS a means to:
 - **eBGP:** obtain subnet reachability information from neighboring AS's.
 - **iBGP:** propagate reachability information to all AS-internal routers.
 - determine "good" routes to other networks based on reachability information and policy.
- allows subnet to advertise its existence to the rest of the Internet: "I am here"

IDRP (Inter-Domain Routing Protocol)

- Designated for use with IPv6. It is a super set of BGP's functions.
- An ISO standard within the OSI family, but not dependent on OSI networking.
- Can deal with multiple internet protocols and address schemes.
- Allows confederations: aggregates of autonomous systems which can be viewed externally as a single entity. This can be done hierarchically, allowing scalable routing.

BGP vs IDRPs

- IDRPs are not tied to TCP, as BGP is.
- BGP uses 16-bit AS numbers. IDRPs use various-length identifiers.
- IDRPs can deal with multiple internet protocols and address schemes.
- BGP specifies the complete list of ASes that a path visits. IDRPs allow confederations. (Most important difference).



▪ Network Layer

- Routing Fundamentals
- Classification of Protocols
- Routing Table

▪ Routing Algorithms

- Fundamentals
- Link State Routing
 - Topology Dissemination
 - Computing Shortest Path (Dijkstra)
- Distance Vector Routing
 - Bellman Ford Algorithm
 - Distance Vector Updates

• Intra-domain Routing

- RIP
 - rip updates
 - route poisoning
 - count-to-infinity problem
 - redundancy-in-networks problem
 - rip timers
 - RIPv1, RIPv2, RIPng
- OSPF
 - 3-step-process
 - init-state
 - OSPF-states
 - two-way-state-to-full-state
 - full-state
 - OSPF hierarchical design

▪ Inter Domain Routing

- Path Vector Routing
- BGP
- IDRP



Curtin University

THANK YOU

Make tomorrow better.



Curtin University

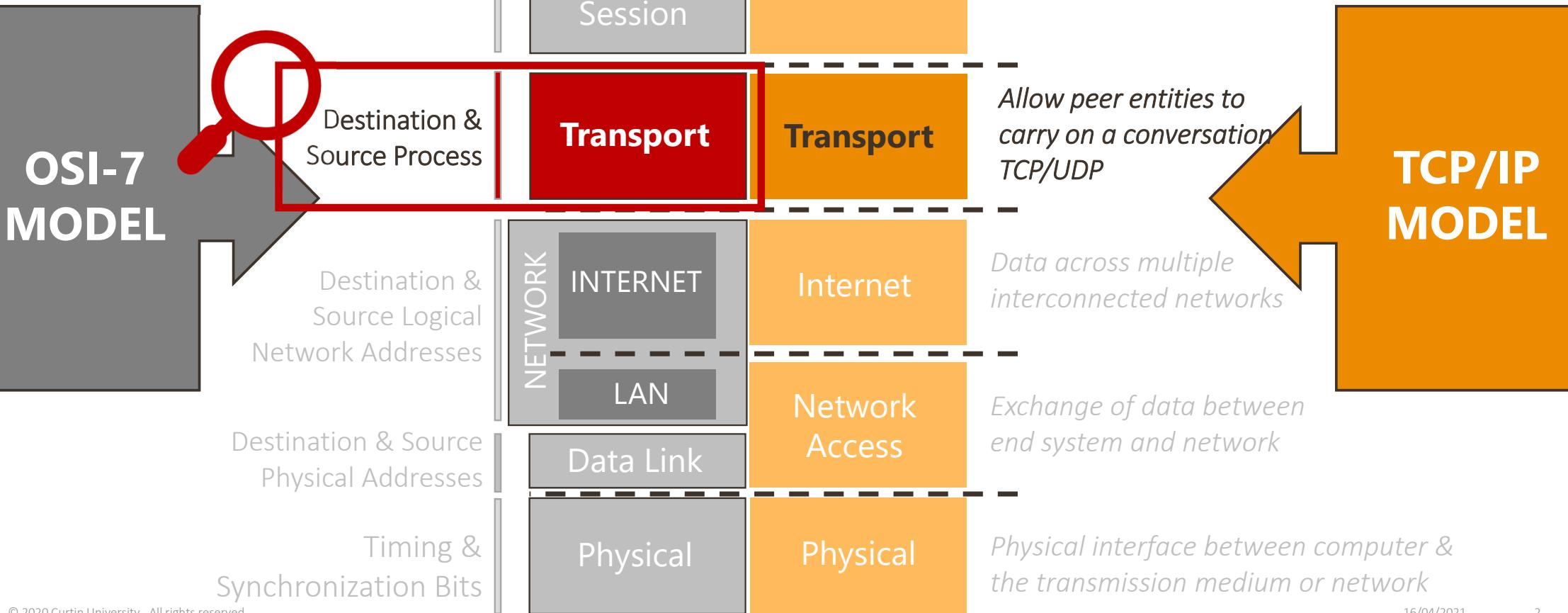
Transport Layer I

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 07

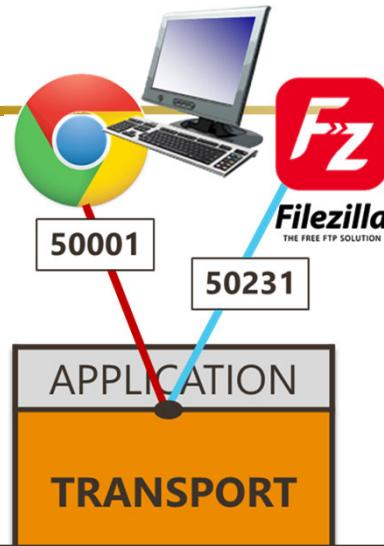
Semester 1, 2021

A GLOBAL UNIVERSITY

WESTERN AUSTRALIA | DUBAI | MALAYSIA | MAURITIUS | SINGAPORE



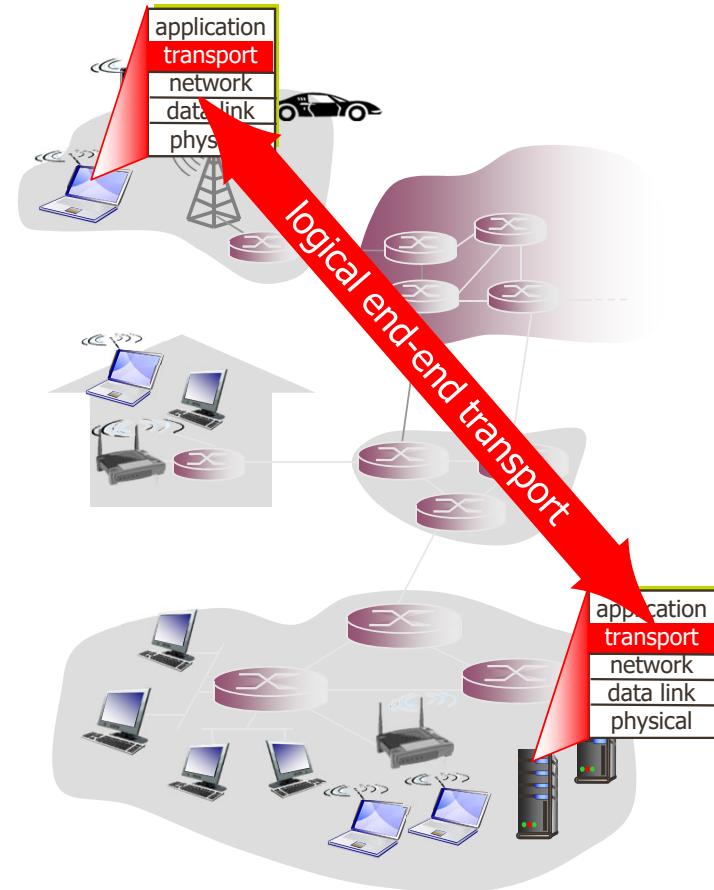
Transport Layer



Transport layer provides communication services for a **process** in a host **to** a **process** in another host

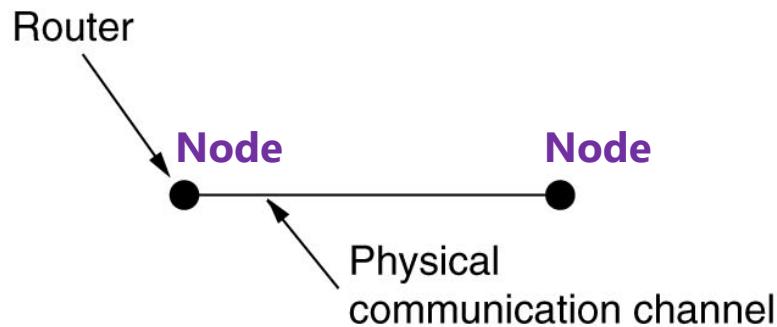
Transport Layer

- **Heart** of whole protocol hierarchy
- **Reliable** data transmission
- **Cost-effective** data transport
- **Independent** of physical network

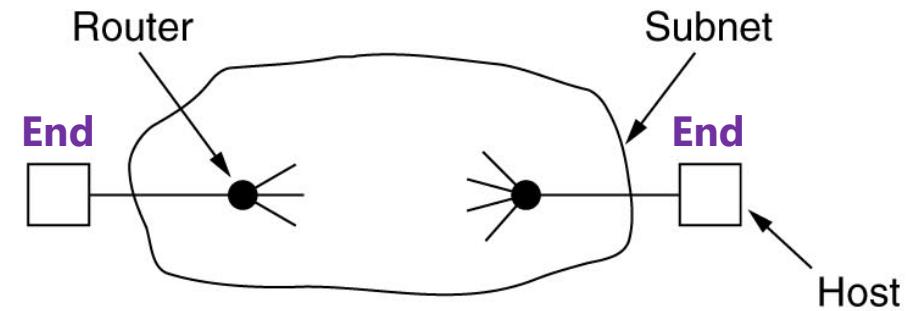


Transport Layer – cont.

- Provides **end-to-end communication** for individual applications



(a)

Data Link Layer

(b)

Network Layer

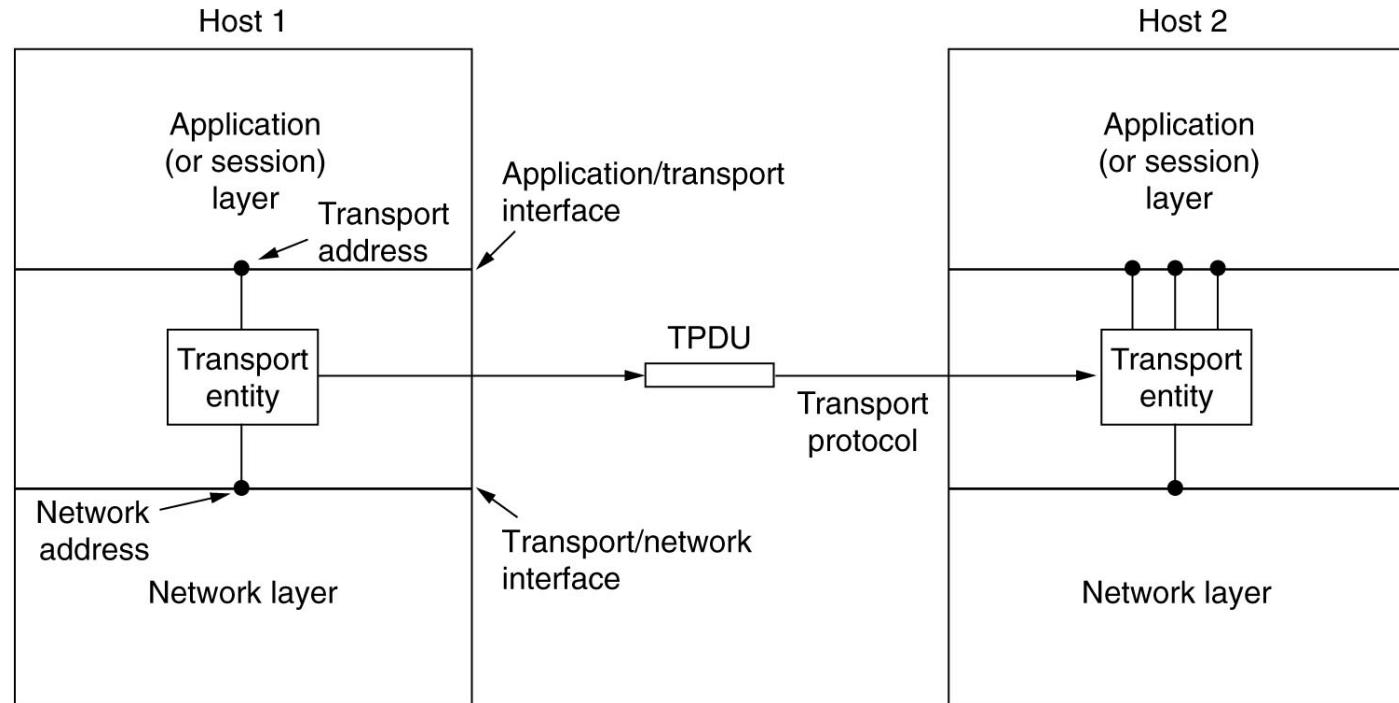
Transport Entity

Hardware and/or software within the transport layer



- Located in:

- ✓ OS **kernel** or
- ✓ Separated **user process** or
- ✓ **Lib** package bound into network application or
- ✓ Conceivably on the network interface card (**NIC**)



Why Transport layer?

- Transport layer entity on user's machines, while network layers mostly runs on routers
- To be more reliable than the underlying network service
- **Transport layer primitives** can be implemented as calls to library procedures in order to make them independent of network service primitives



Transport Layer Elements

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

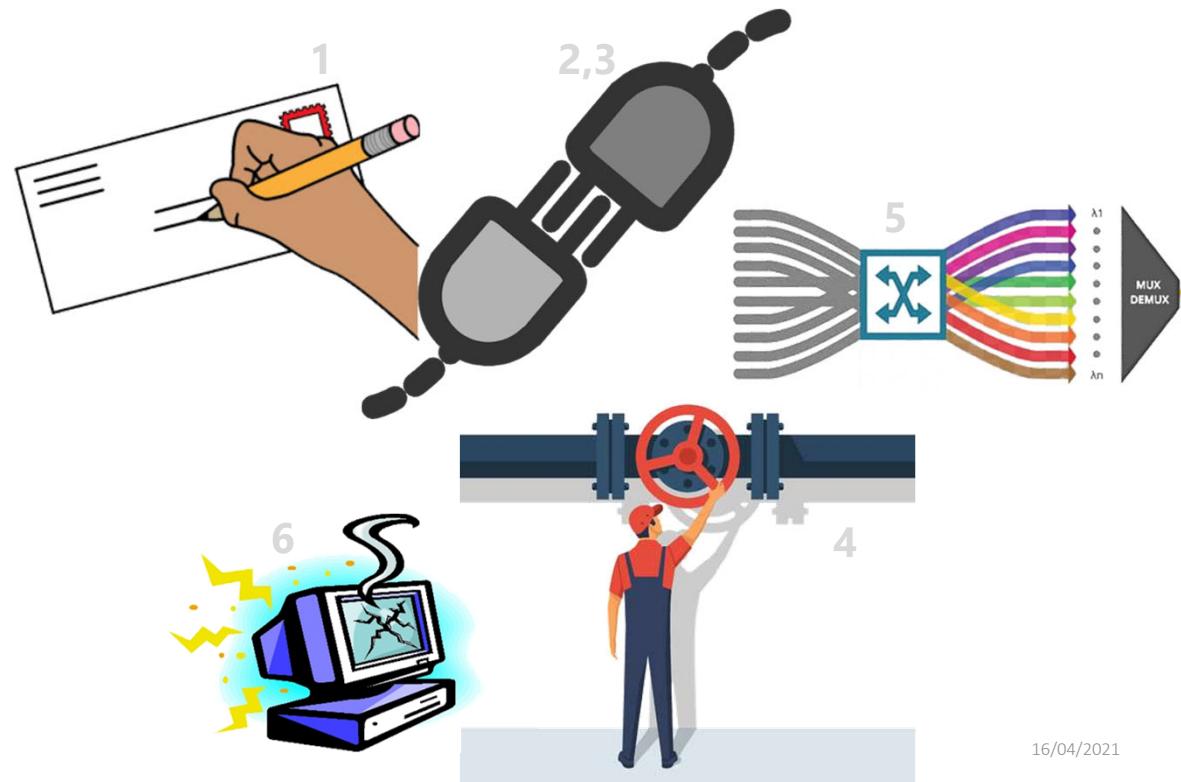
Transport Layer Elements

- **Transport Protocols ~= Data Link Protocols**

- ✓ *Error control*
 - ✓ *Sequencing*
 - ✓ *Flow control and other issues.*

- *differences in:*

1. **Addressing**
2. **Connection Establishment**
3. **Connection Release**
4. **Flow Control & Buffering**
5. **Multiplexing**
6. **Crash Recovery**



1. Addressing



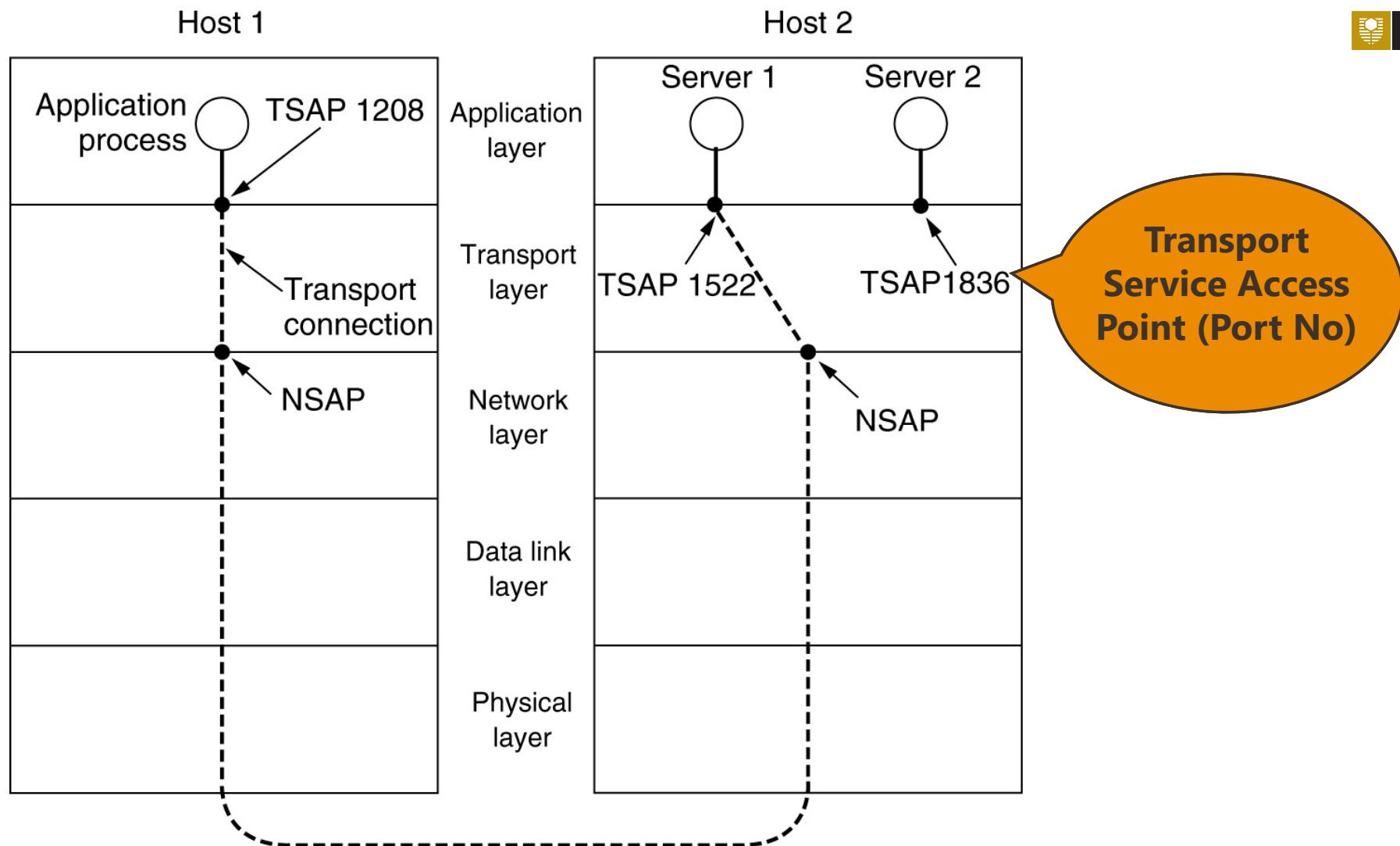
Which application (server) on a remote host **to connect**

- ✓ *Assign a port number to Application !*



How to determine the **port number** of a particular application



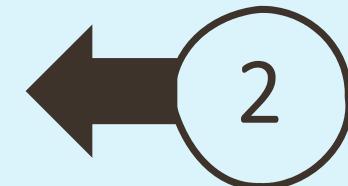
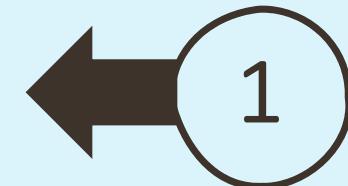


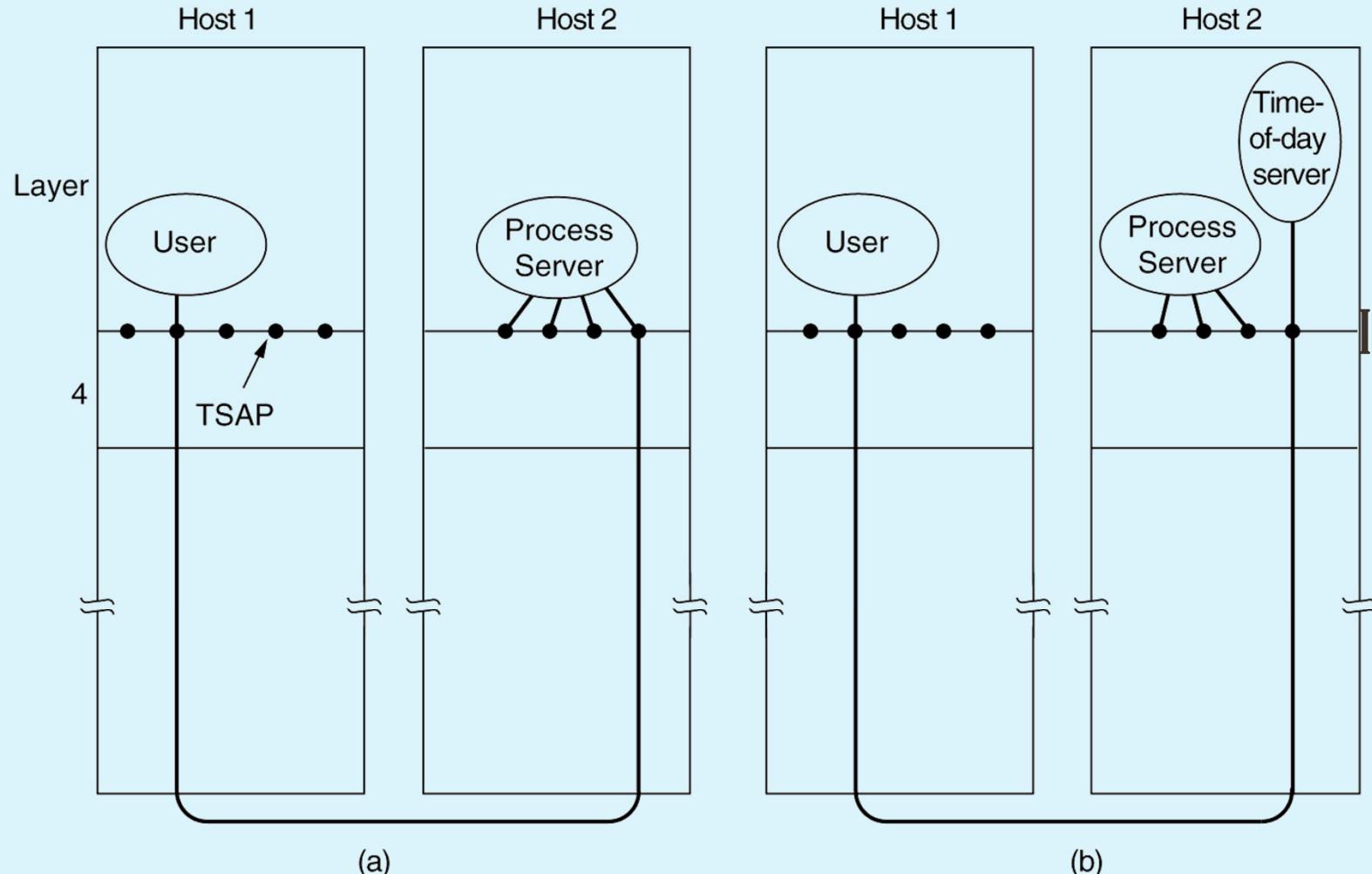
1. Addressing: TSAP

- Popular applications use the same TSAP address (permanently) on every host
 - **well-known ports**

IMPLEMENTATION OPTIONS

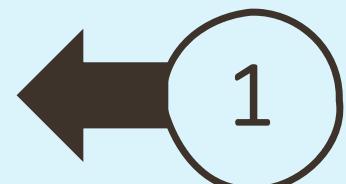
- Others use:
 - Initial Connection Protocol**
 - a special Process Server acts as a **proxy** which listen to a set of less heavily used servers at the same time
 - Directory Server**
 - a.k.a. Name Server
 - Remote user connect to a well known TSAP/port address to resolves symbolic service names into TSAPs
 - Client break connection with the directory server and make a new connection with the required service on the TSAP/port number supplied





Example initial connection protocol: a user process in host 1 establishing a connection with a time-of-day server in host 2 (Tanenbaum)

IMPLEMENTATION OPTIONS



**Initial
Connection
Protocol**

Complete TCP address (**Socket**)

<IP>: <Port> or <NSAP>:<TSAP>
i.e. **192.168.10.5:80**

- Port Numbers Ranges [RFC768]:

Well Known Ports: 0 - 1023

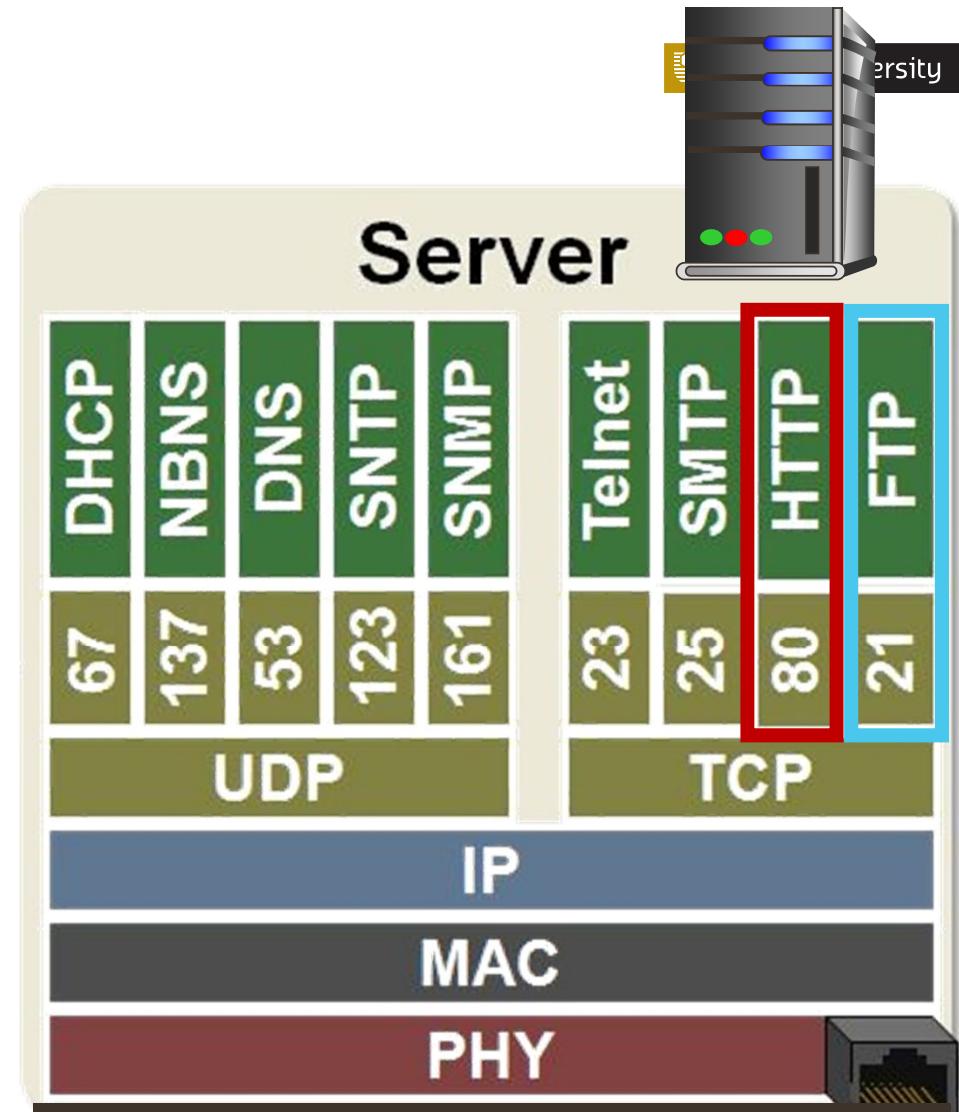
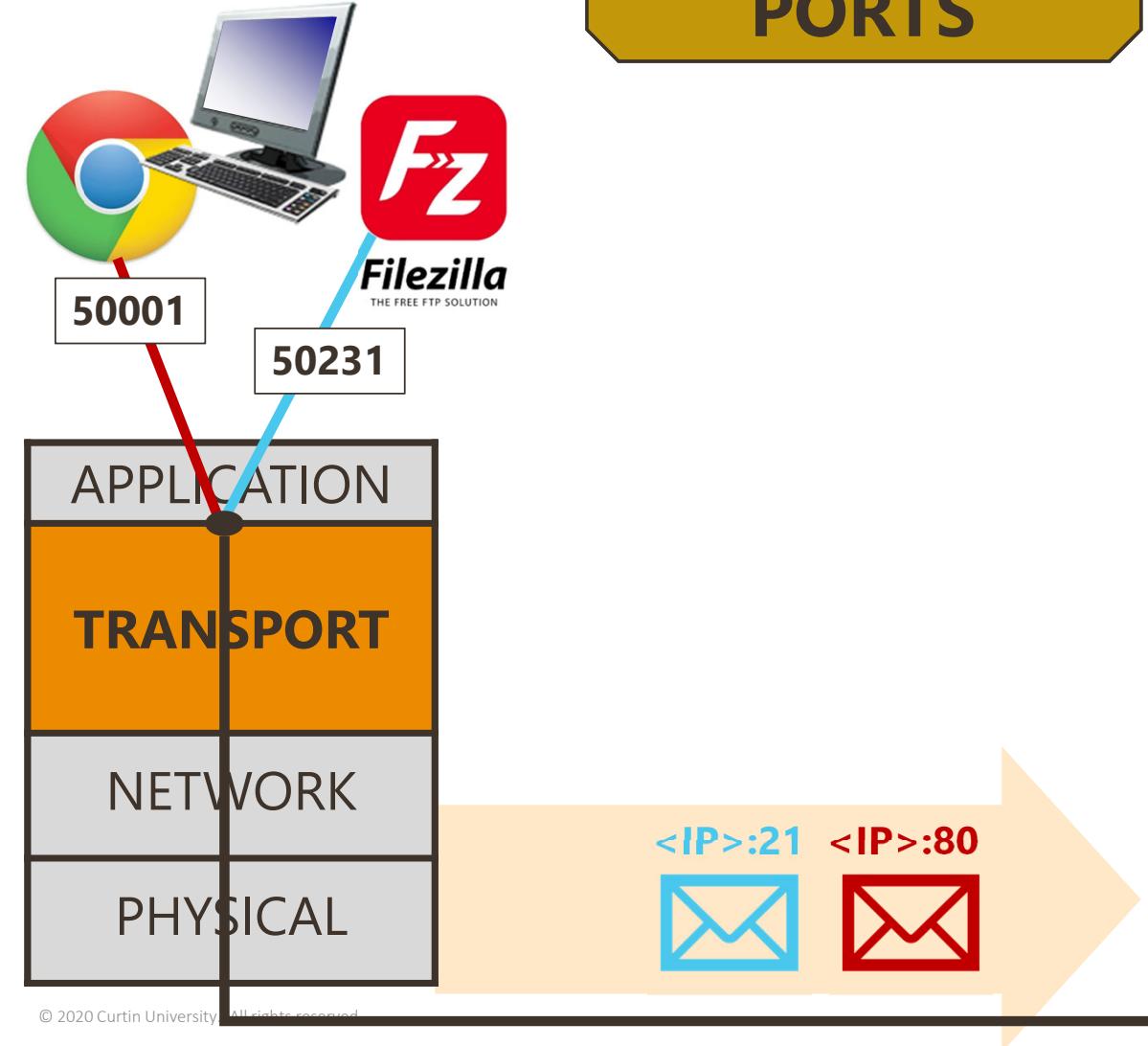
Assigned by the IANA
(Internet Assigned
Numbers Authority)

Registered Ports: 1024 - 49151

Dynamic / Private Ports: 49152 - 65535

IANA registered, for
the convenience of
Internet Community

WELL-KNOWN PORTS



2. Connection Establishment

- Send a Connection Request (**CR**)
- Wait for a Connection Accepted (**CA**)



network can lose, store, and duplicate packets.

Duplicates

- Ensure packets will not live in the network forever

- **Solutions:**

- ✓ Restrict Packet **lifetime**
- ✓ Restricted **subnet design**
- ✓ Putting a **hop counter** in each packet
- ✓ **Time-stamping** each packet

Duplicates – cont.

- Need to guarantee **not only** that **TPDU** is “dead”, **but** also all **acknowledgements**

- Wait a time T after a packet has been sent
 - ✓ *Ensure all traces of the packet + its acknowledgements are gone*

T : some small **multiple** of the true **maximum packet lifetime**

The multiple is **protocol dependent**; makes T longer

Tomlinson (1975)



Equip each host with a **clock** in the form of a **binary** counter that **continue running even when the host goes down**

- Some part of the counter is used as the **initial sequence no.** when a connection is set up
- Ensure that **two identically numbered TPDUs** are **never outstanding** at the same time
 - ✓ **Sequence space** should be large
- Once both transport entities agreed on the initial sequence number, **any sliding window protocol** can be used for data flow control.

Tomlinson (1975) – cont.

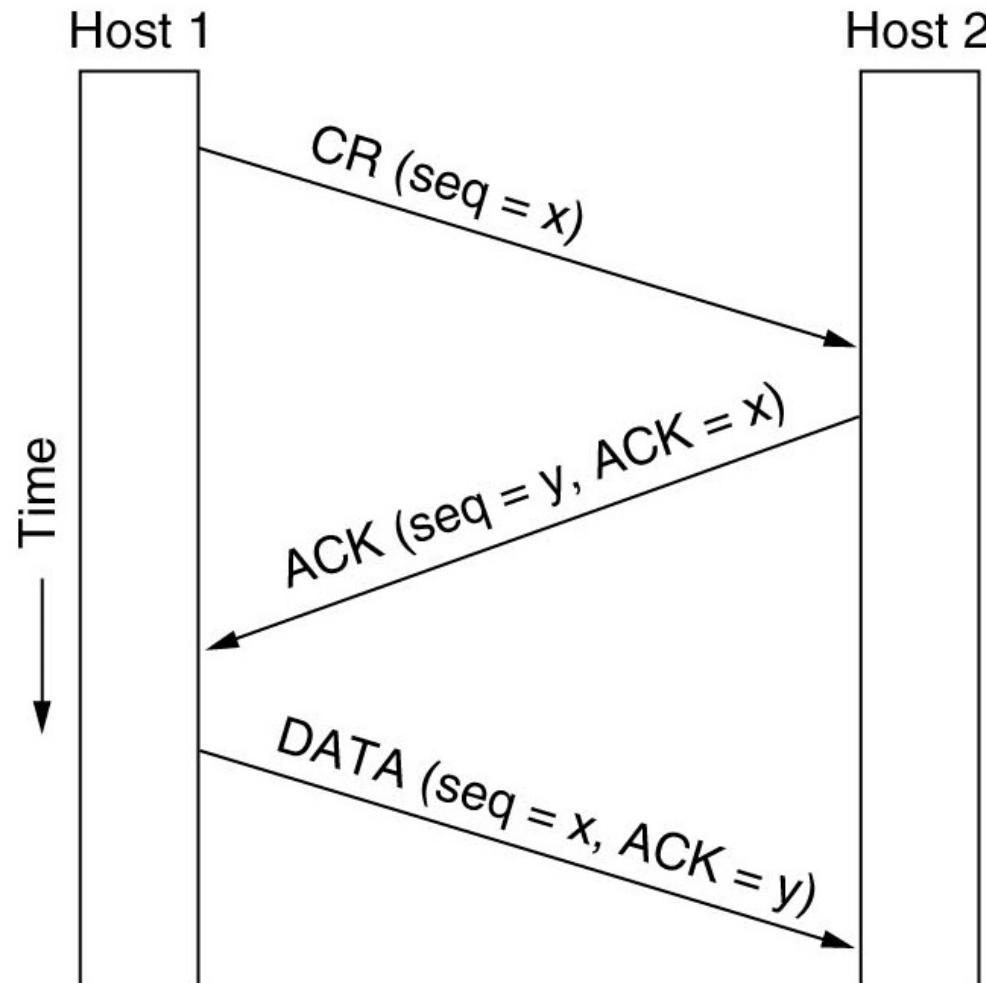
▪ what happen when a host crashed?

- transport entity's sequence number is lost
- one solution - require transport entities to be idle for T seconds
however, T may be large

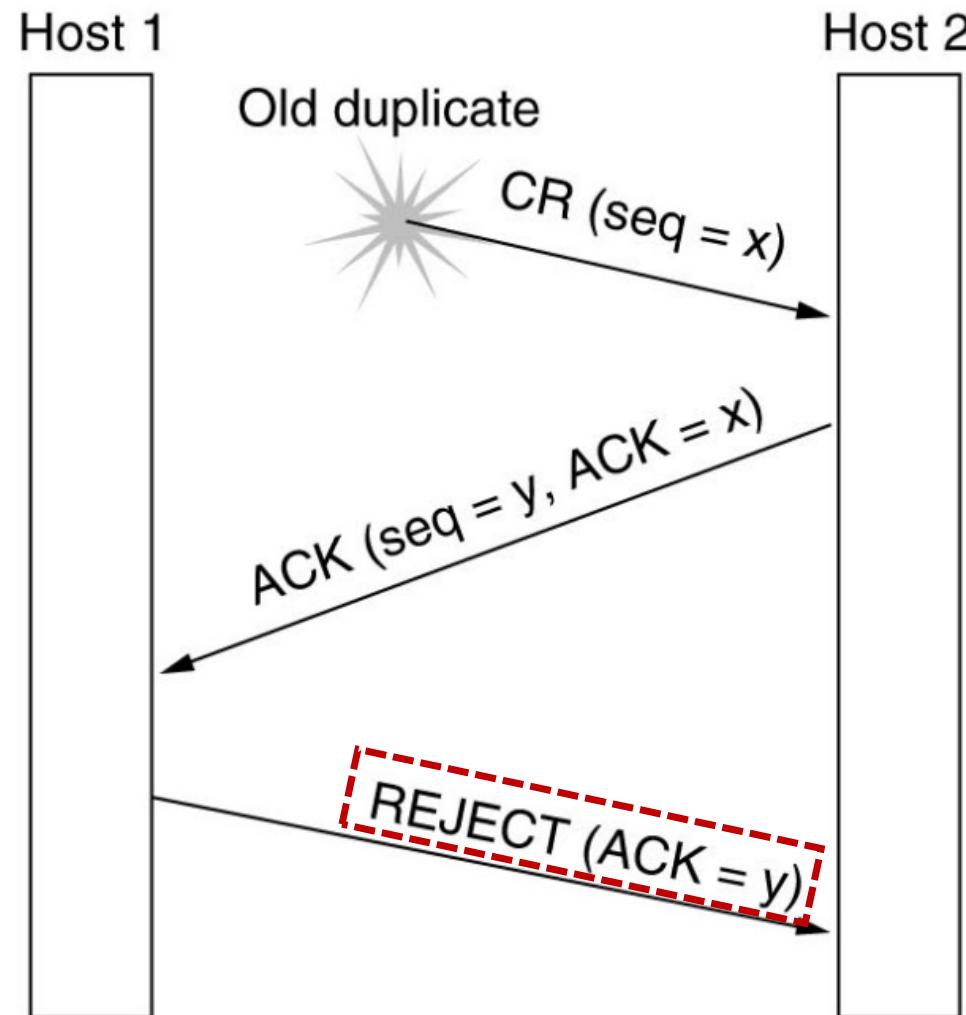
▪ Restriction on the use of sequence numbers

- Prevent sequence numbers from being used (assigned to new TPDUs)
for a time T before their potential use as initial sequence numbers

Tomlinson (1975): 3-Way Handshake

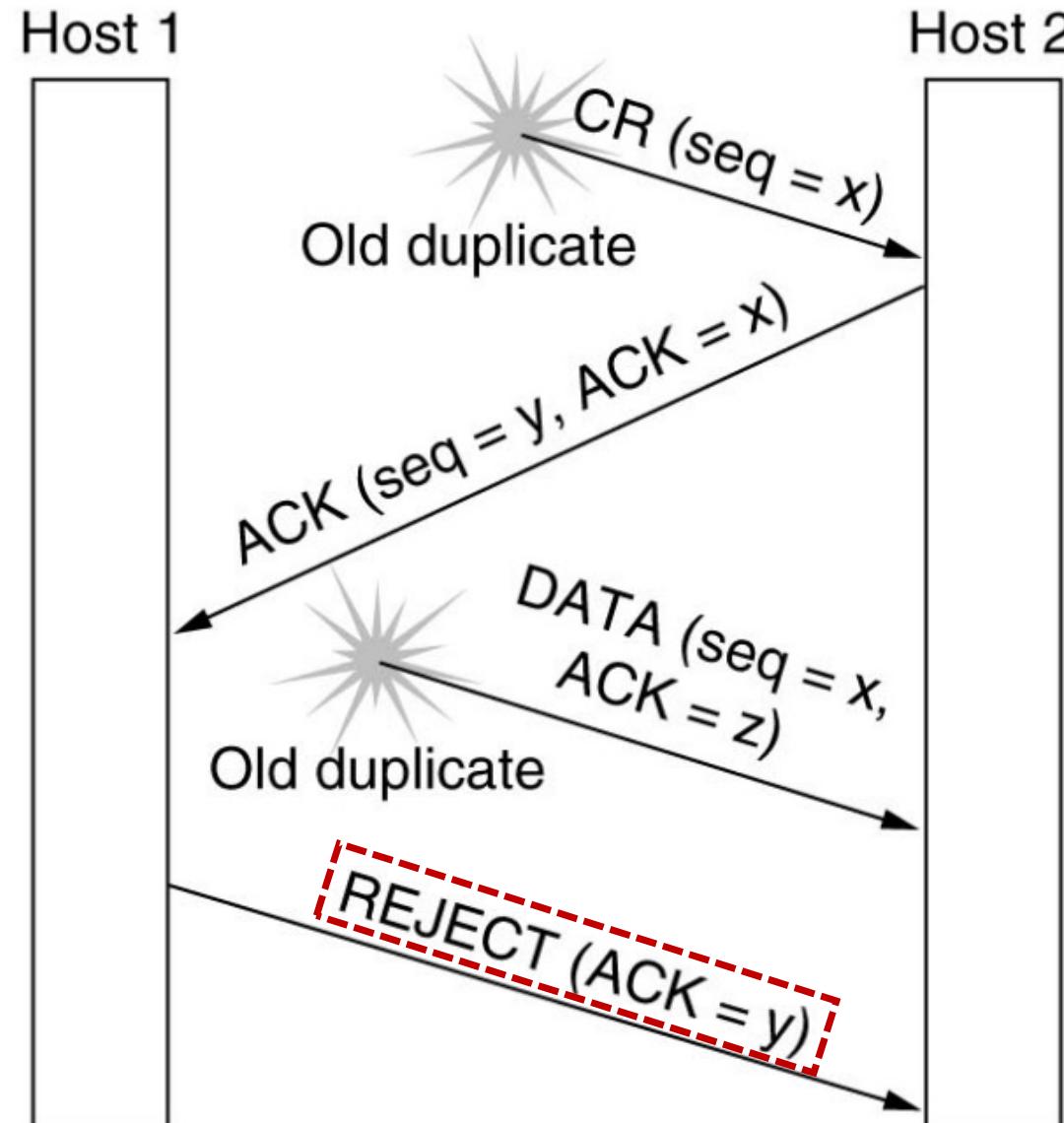


DUPLICATE CR



DUPLICATE CR & DUPLICATE ACK

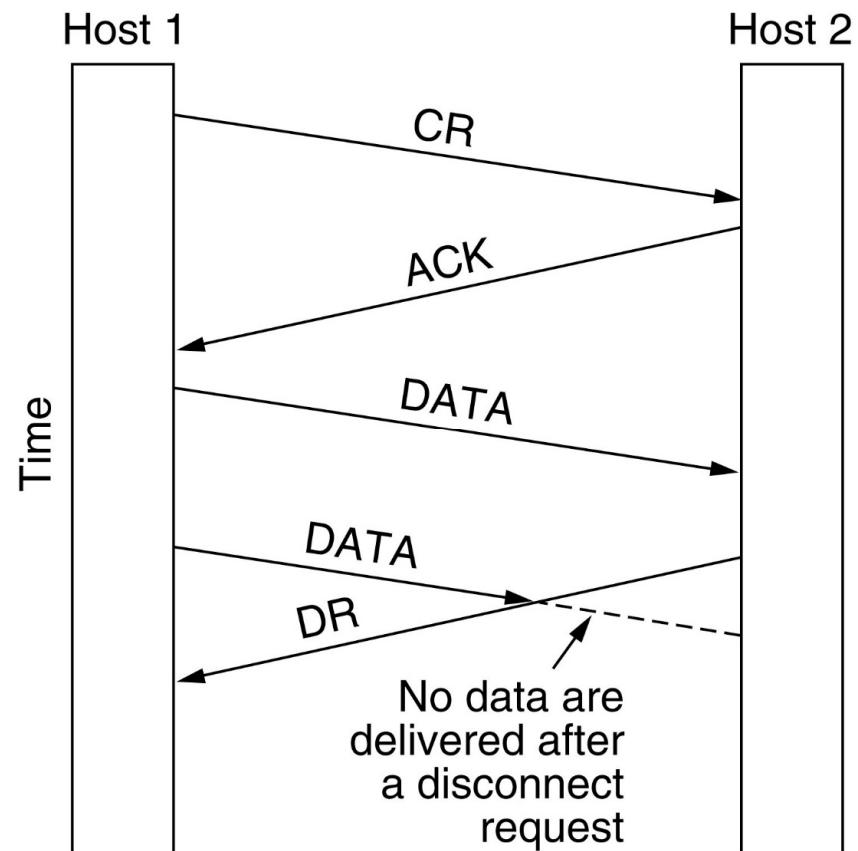
Initial sequence number previously acknowledged will be the **incorrect** one, and the connection will be **abandoned**



3. Connection Release

1. Asymmetric Release

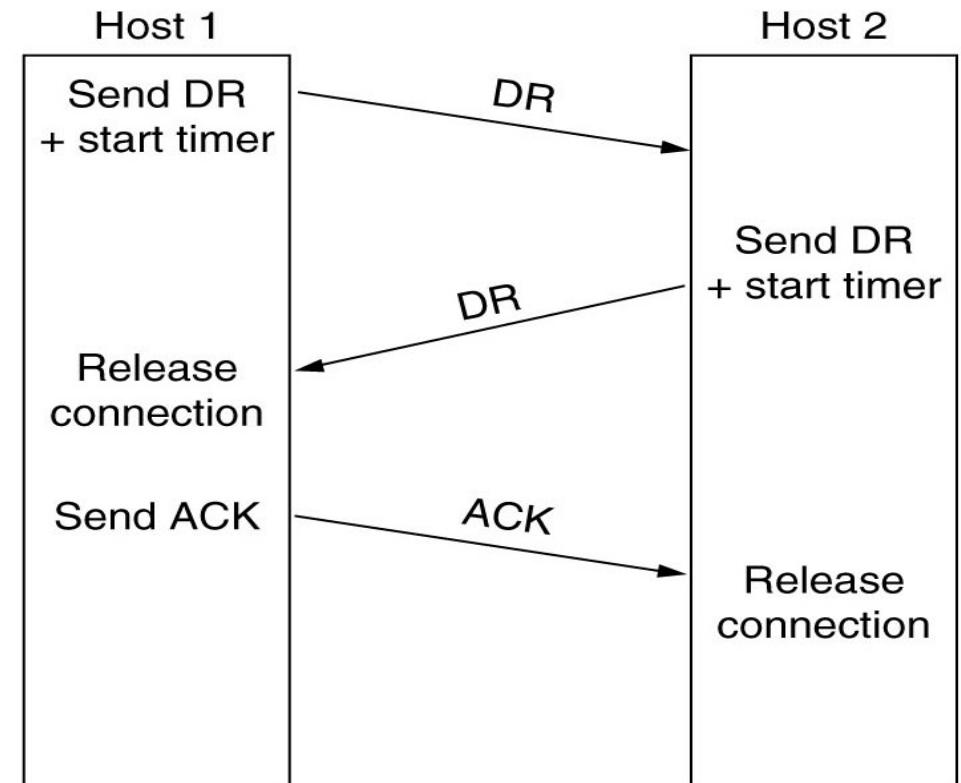
- ✓ Either user can issue a disconnect
- ✓ Rather abrupt, can result in **loss of data**



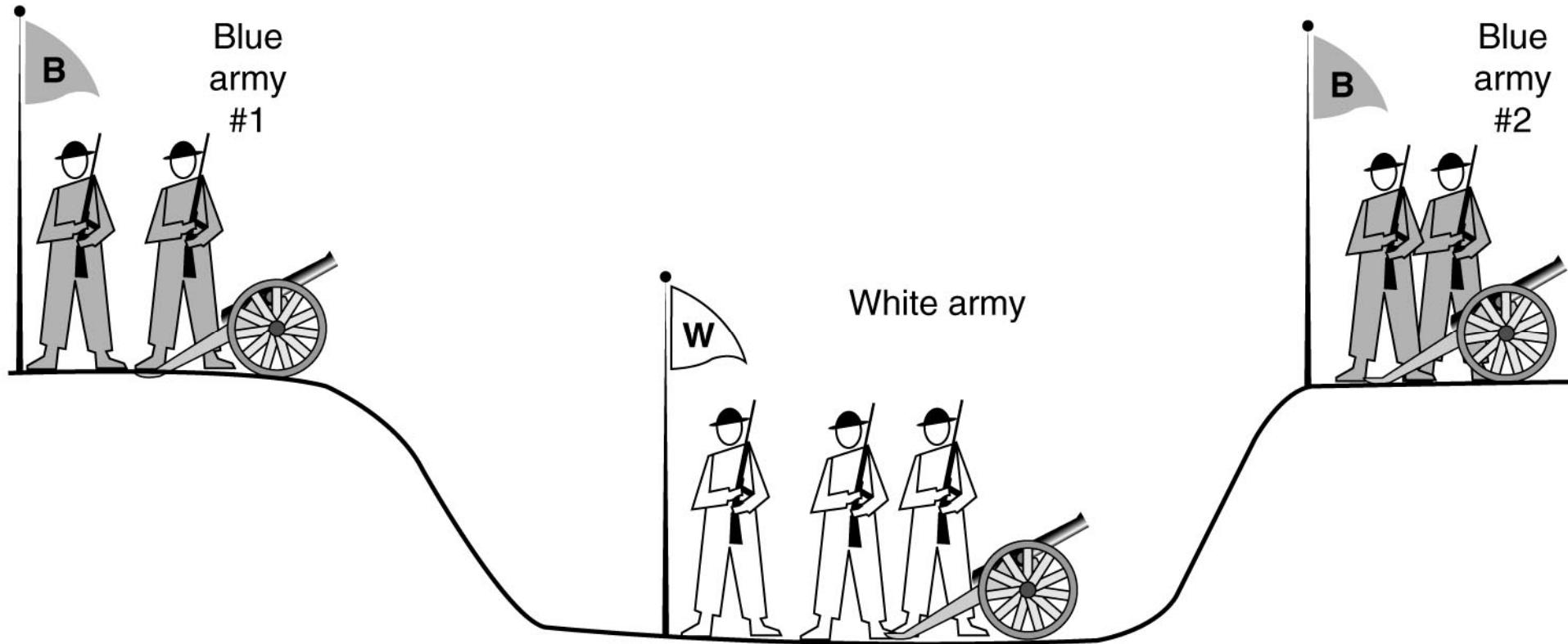
3. Connection Release – cont.

2. Symmetric Release

- ✓ each direction of the connection is closed separately.
- ✓ **Not foolproof !** (two-army problem)
- ✓ Can result in one **half** of a connection remaining **open**

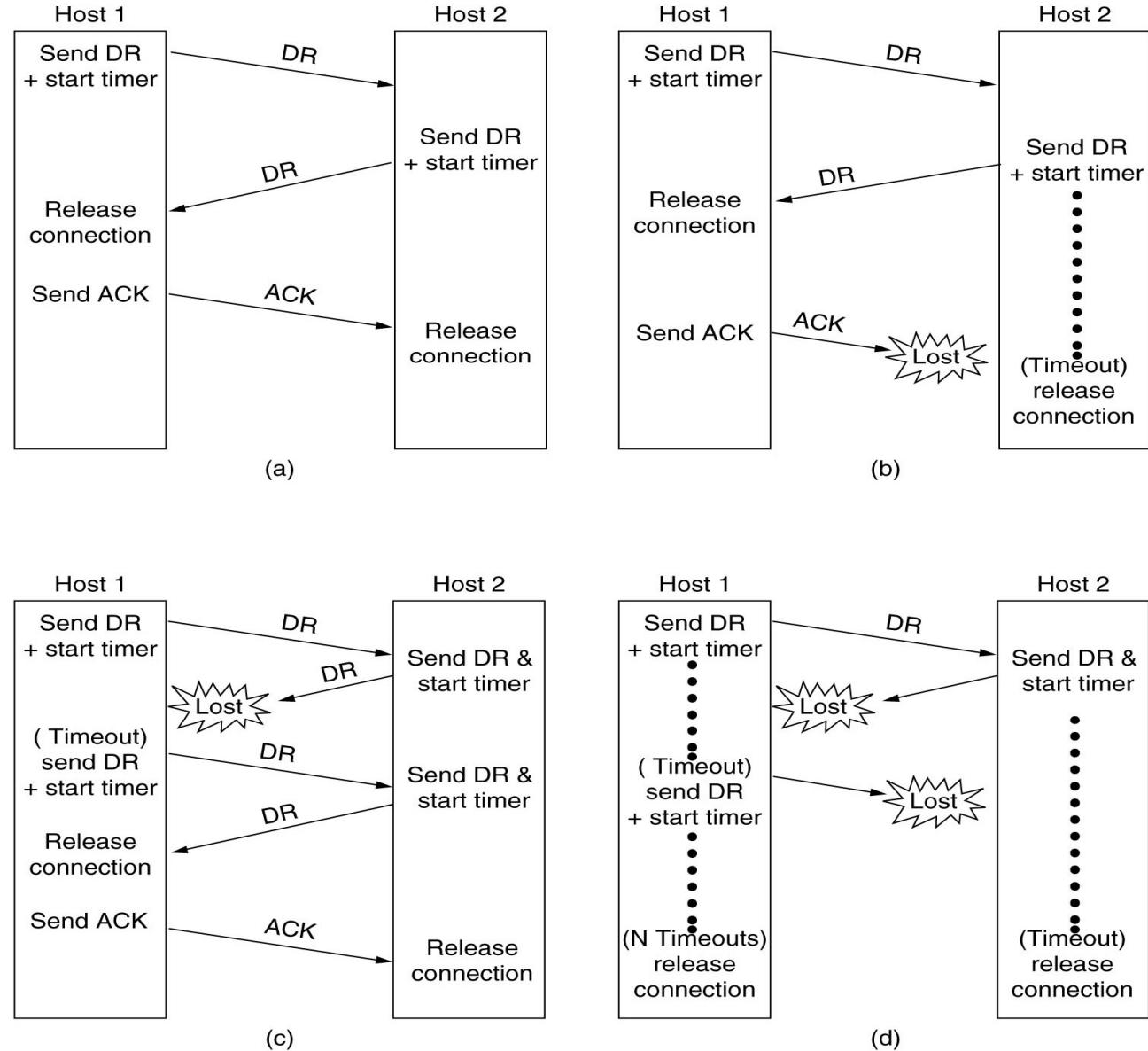


Two Army Problem



CONN. RELEASE

- a) Normal three-way handshake
- b) Final ACK lost
- c) Response lost
- d) Response lost and subsequent DRs lost



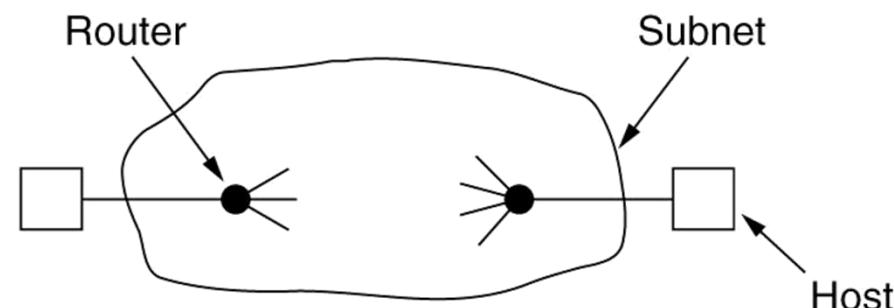
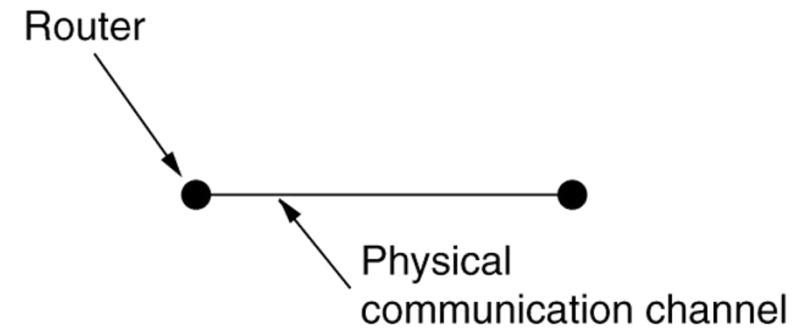
4. Flow Control & Buffering

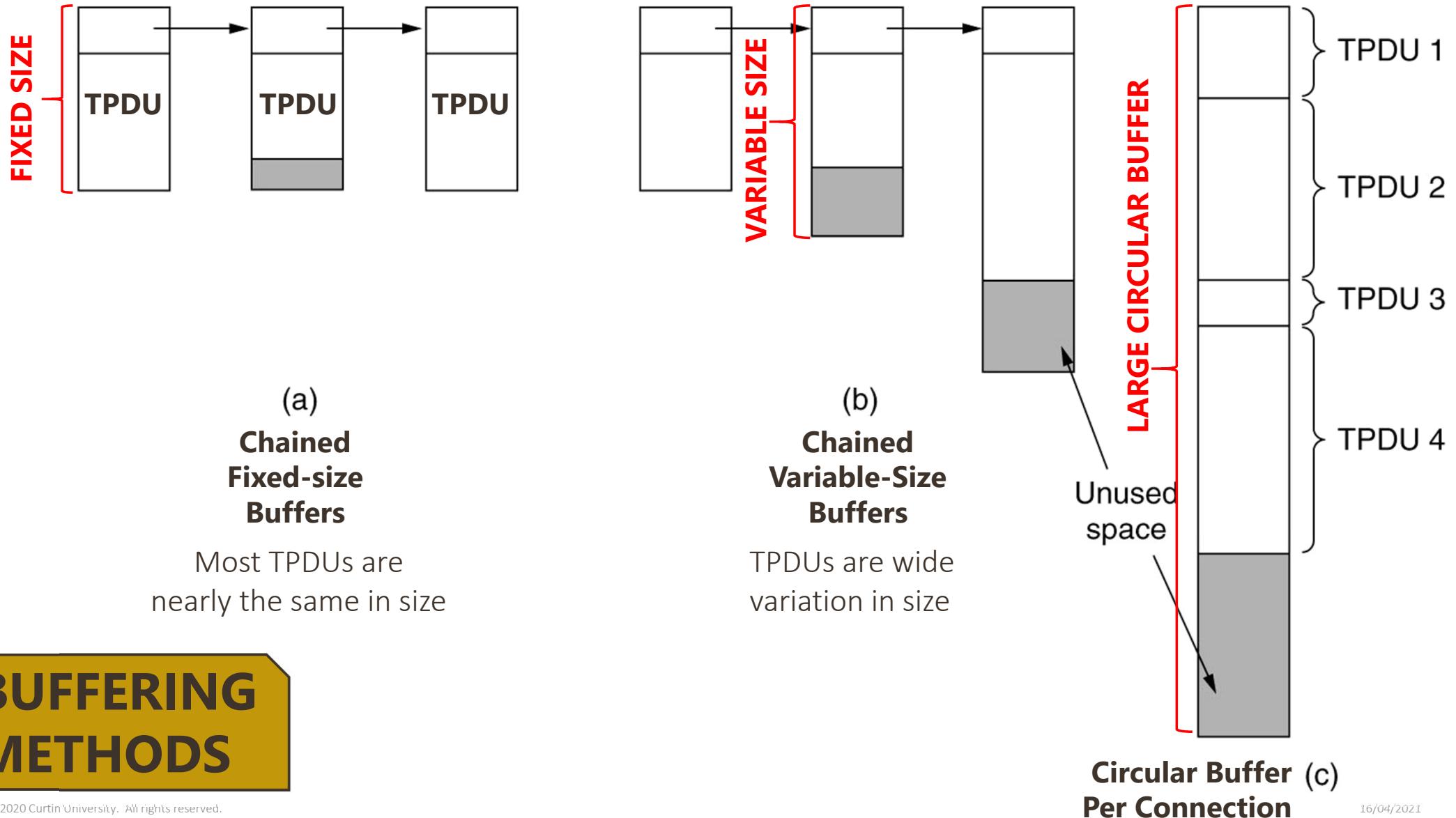
- Flow control in **TL** ~ = Flow control in **DL**
 - ✓ But not the same
 - ✓ Both use sliding window or other scheme

- A router usually has relatively few connection lines

- A host may have numerous connections
 - ✓ transport entity must buffer all TPDUs sent, same reason for data link layer
 - ✓ data link layer buffering strategy cannot be used

- Source Buffering & Destination Buffering





Source / Destination Buffering

- ✓ **low-bandwidth, bursty traffic:** *buffer at the sender*
- ✓ **high-bandwidth, smooth traffic:** *buffer at the receiver*

Dynamic Buffer Management

more on this later !

- ✓ decouple the buffering from the acknowledgement
- ✓ a **variable-sized window**

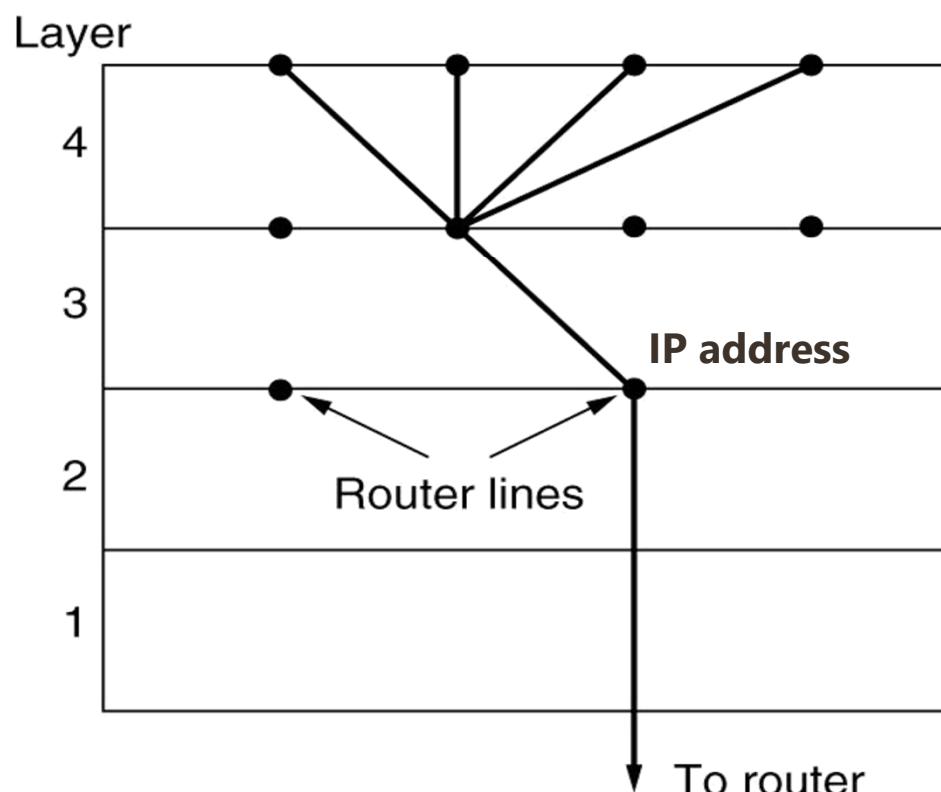
As memory price fall – equip hosts with more memory.

BUT, the **carrying capacity of the subnet is the bottleneck**

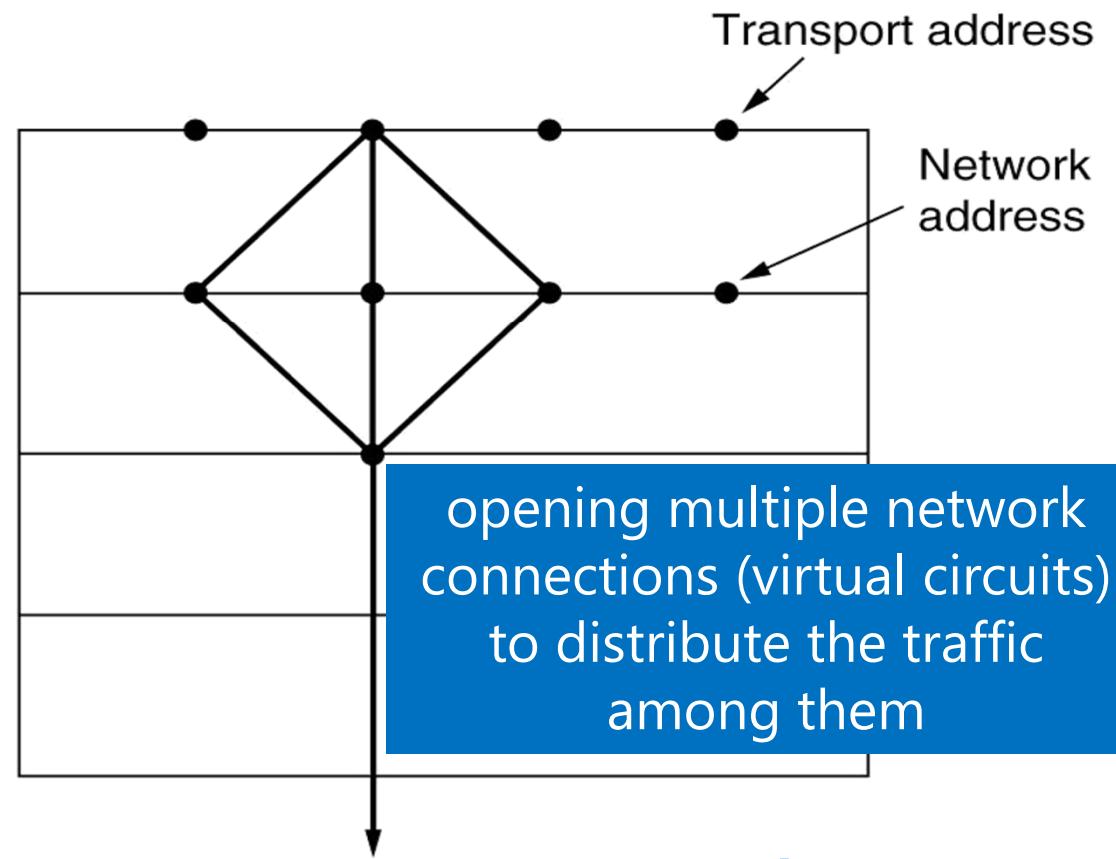


solution: mechanism based on subnet's carrying capacity rather than on the receiver's buffering capacity.

5. Multiplexing



**Upward
Multiplexing** (a)



**Downward
Multiplexing** (b)

opening multiple network connections (virtual circuits) to distribute the traffic among them

Crash Recovery

1. Router / Network Crash

- **For datagram service network:**

- ✓ Transport entities expect lost TPDUs all the time and know how to cope

- **For connection-oriented network:**

- ✓ Loss of virtual circuit is handled by establishing a new one and probing the remote transport entity for TPDUs received and not received



Crash Recovery

2. Host Crash

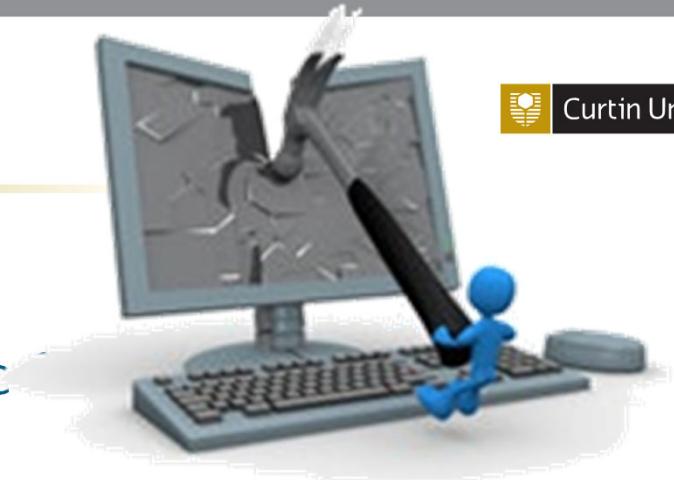
- After reboot tables/parameters are reinitialized
- To recover previous status:
 - ✓ Server (receiver) might broadcast TPDU to all other hosts, announcing that it had just rebooted and request that its clients (senders) inform it of the status of all open connections. Each client (sender) can be in one of the two states
 - a. **No TPDUs outstanding, S0**
 - b. **One TPDU outstanding, S1**
- Server (receiver) can be programmed in one of two ways:
 - ✓ **Send acknowledgement First (A)**
 - ✓ **Write to application process First (W)**



Crash Recovery

2. Host Crash – cont.

- Client (sender) can be programmed in one of the
 - ✓ **always** retransmit the last TPDU
 - ✓ **never** retransmit the last TPDU
 - ✓ retransmit only in **state S0** (No TPDU outstanding)
 - ✓ retransmit only in **state S1** (TPDU outstanding)



There are **always** situations where
the protocol **fails to recover** properly

!

FALIURES

Strategy used by sending host

	AC(W)	AWC	C(AW)
Always retransmit	OK	DUP	OK
Never retransmit	LOST	OK	LOST
Retransmit in S0	OK	DUP	LOST
Retransmit in S1	LOST	OK	OK

Strategy used by receiving host

	First ACK, then write		First write, then ACK		
	AC(W)	AWC	C(WA)	W AC	WC(A)
OK	OK	DUP	OK	DUP	DUP
LOST	LOST	OK	LOST	OK	OK
OK	OK	DUP	LOST	DUP	OK
LOST	LOST	OK	OK	OK	DUP

OK = Protocol functions correctly

DUP = Protocol generates a duplicate message

LOST = Protocol loses a message



Transport Control Protocol **(TCP)**

- Fundamentals
- TCP Header
 - Flags (SYN, FIN, ACK, RST)
 - Flag (URG, PSH) – in depth
 - TCP Options
 - Window Size (Dynamic Buffer Management)
- TCP Flow Control
 - Dynamic Buffer Management

TCP

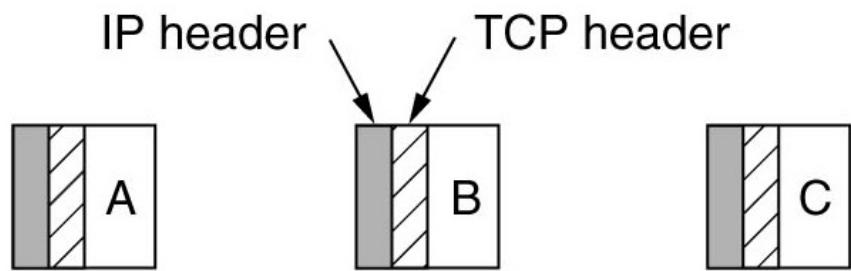
- The (other) **main** transport **protocol** used in the **Internet**
 - ✓ **Connection-oriented** protocol
 - ✓ RFC 793 (formal), RFC 1122 & 1323 (bug fixes)
 - ✓ Provide a reliable end-to-end communication over an unreliable internetwork
- **Connections** are:
 - ✓ **Full duplex** and point-to-point
 - ✓ A **byte stream** not a message stream



No support for Multicasting or Broadcasting



TCP: Header



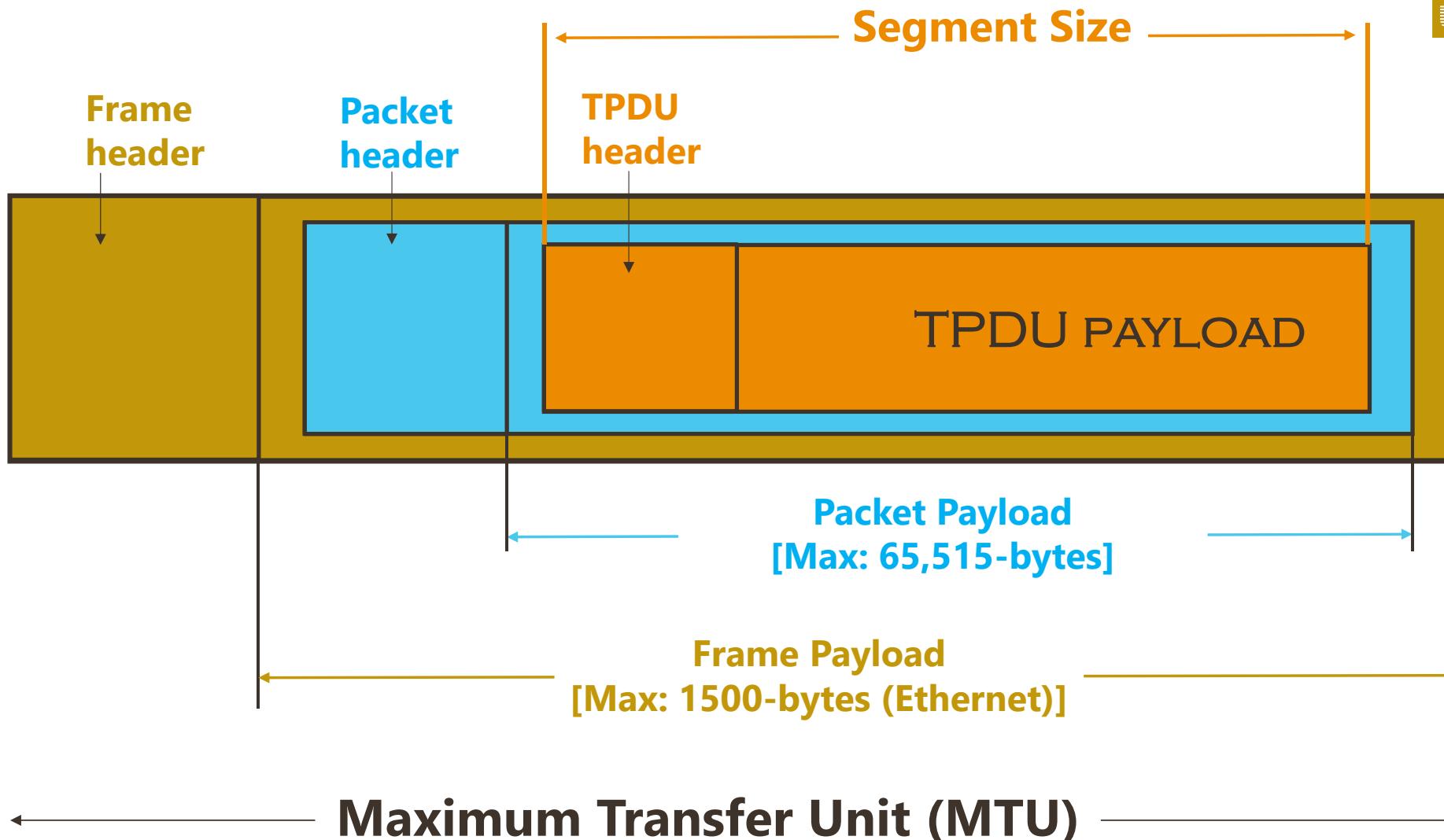
(a)

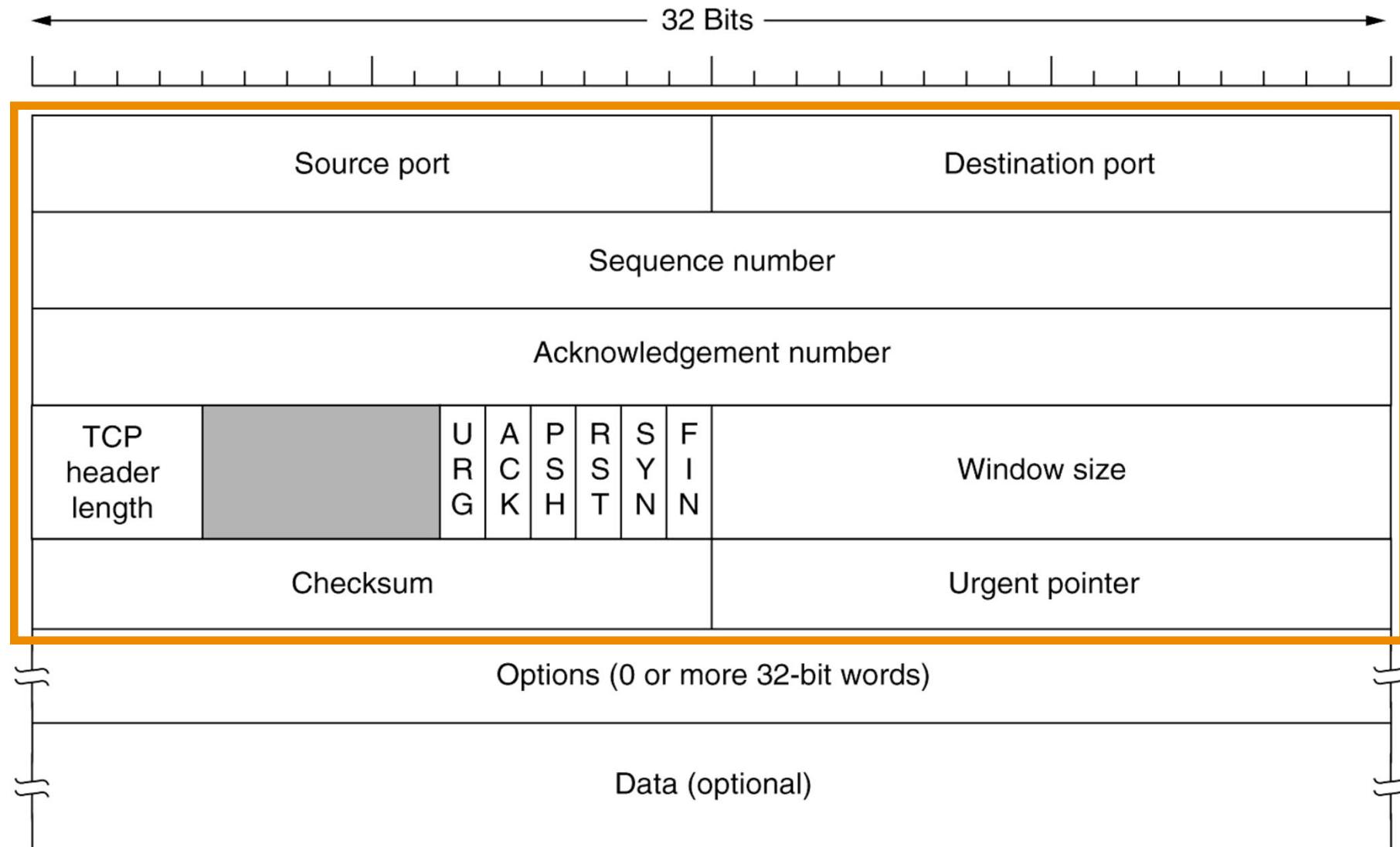
512-byte segments
sent as a separated IP
datagrams



(b)

2048-bytes of data
received to the application
in a single **READ** call





TCP HEADER

Fixed
20-bytes
header

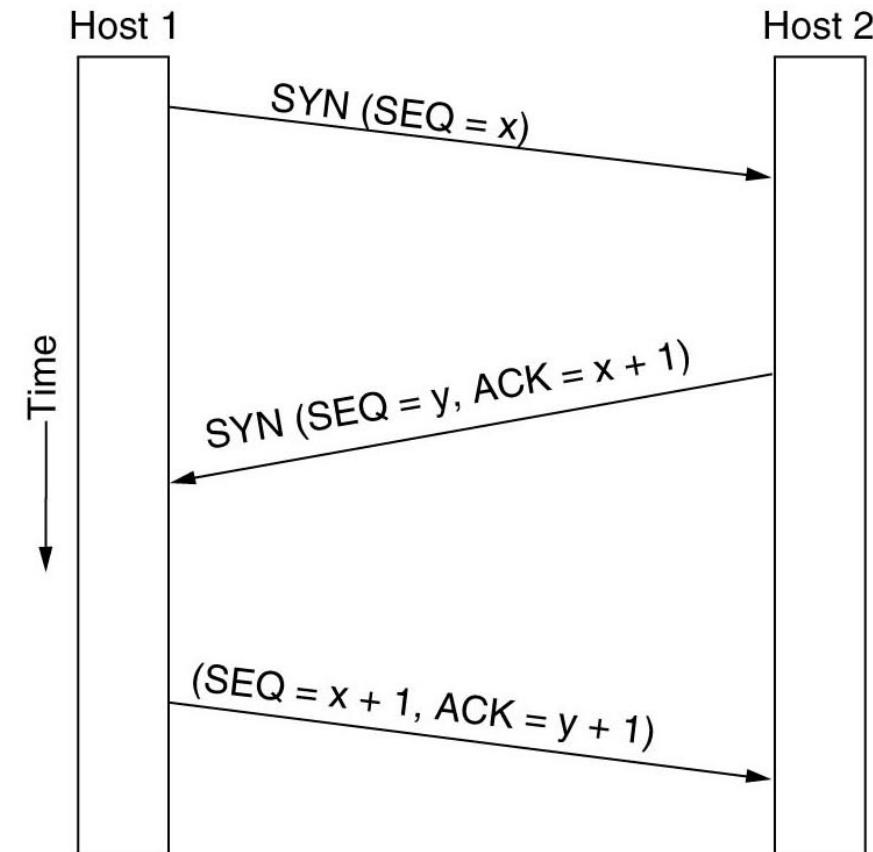
TCP: Header – cont.

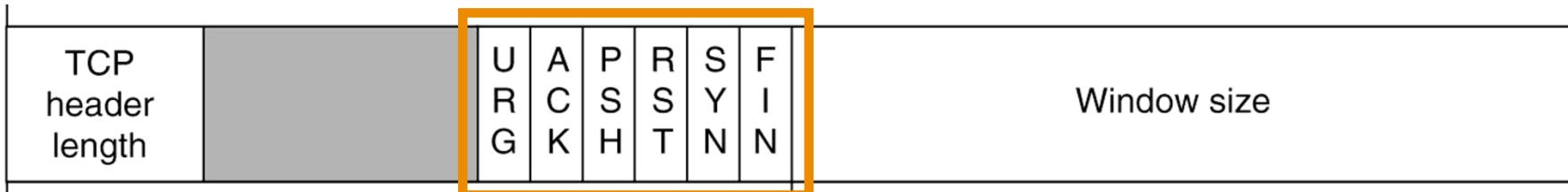
▪ Sequence Number (32-bit)

- sequence number of the first data octet (byte) in this segment
- if SYN is set this field is the initial sequence number (ISN) and the first data octet is ISN+1

▪ Acknowledgement Number (32-bit)

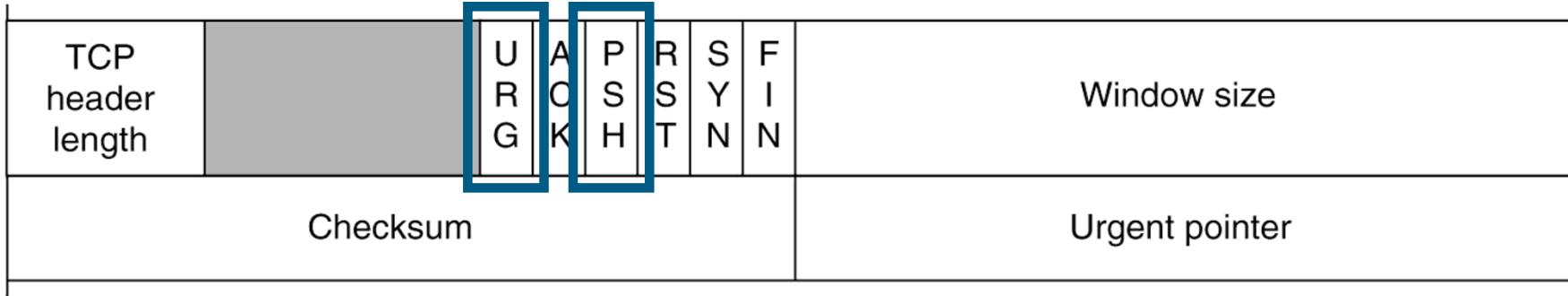
- sequence number of the next data octet the TCP entity expects to receive. May be piggybacked!!
- NOTE that TCP is byte or stream oriented.





- ✓ **URG:** Urgent pointer field significant. Inform the destination TCP user that 'urgent' data is arriving.
- ✓ **ACK:** Acknowledgement field significant.
- ✓ **PSH:** Push function. A TCP user can require TCP to send (receive) all outstanding data up to and including that labelled with a **PUSH** flag.
- ✓ **RST:** Reset the connection.
- ✓ **SYN:** Synchronize the sequence numbers. Used to establish connections.
- ✓ **FIN:** No more data from sender. Used to release connections.

FLAGS



*When application passes data to TCP, TCP may **send it immediately** or **buffer it (at both sides)***

(in order to collect a larger amount to send at once)

FLAGS

- **PUSH Flag – (used by application) ->**

Force/Flush data out



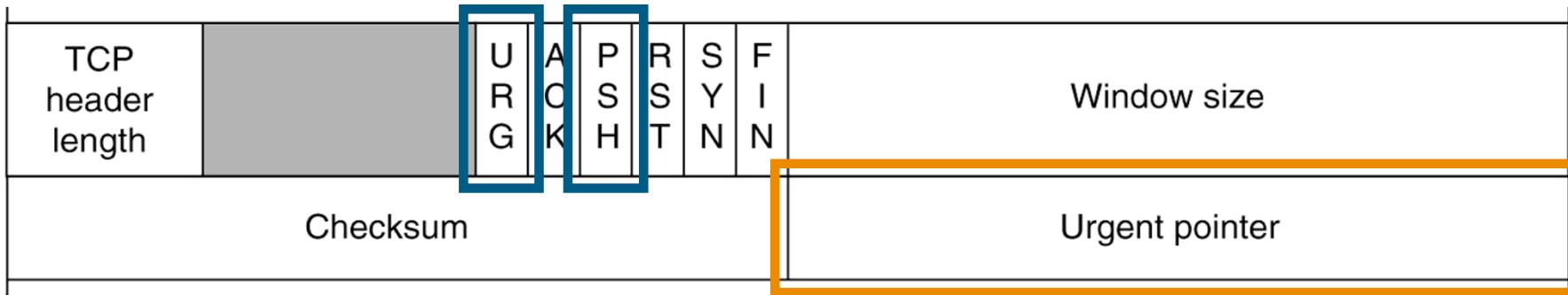
at both sides!

- **URGENT Flag – (used by application) ->**

Cause TCP to stop accumulating data and transmit everything it has for the connection immediately



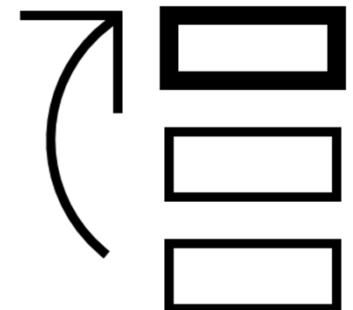
when urgent data is received, receiving application is **interrupted** !
 (stops whatever it was doing) & read the data stream to find the urgent data

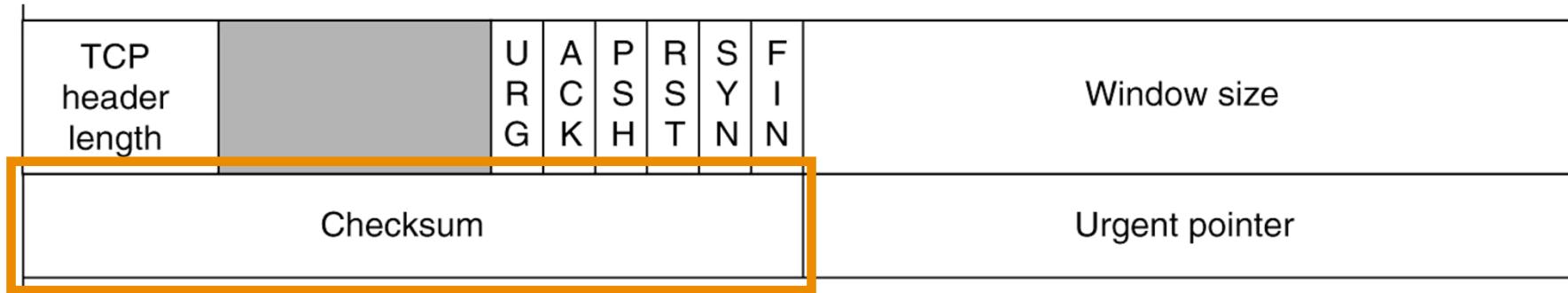


URG POINTER

- Urgent Pointer

- ✓ **Points to the last octet** in a sequence of urgent data.
Allows the receiver to know how much urgent data is coming



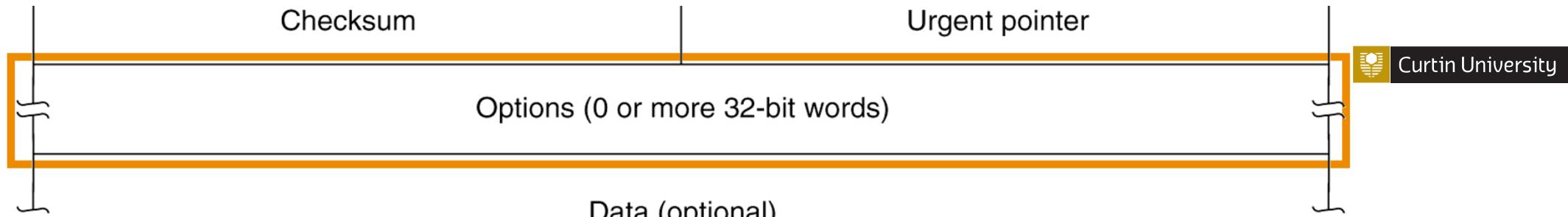


▪ **Checksum:**

- ✓ **header**
- ✓ **data**
- ✓ conceptual **pseudo-header**

CHECKSUM

source & destination addresses, segment length.
This provides protection from **mis-delivery**



✓ Maximum Segment Size (MSS)

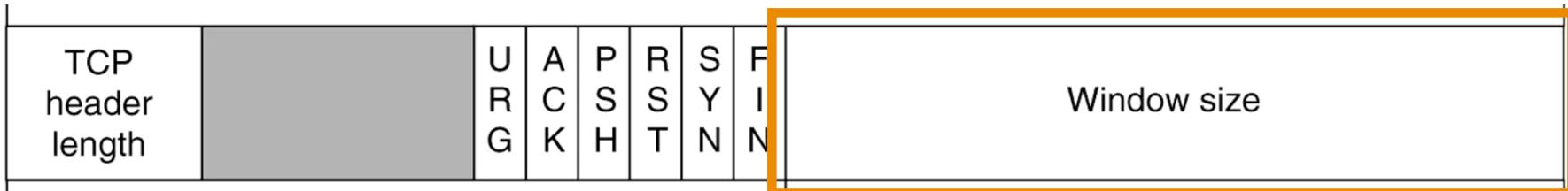
OPTIONS

✓ Window Scale Factor:

Window is multiplied by 2^F where F is the window scale factor.

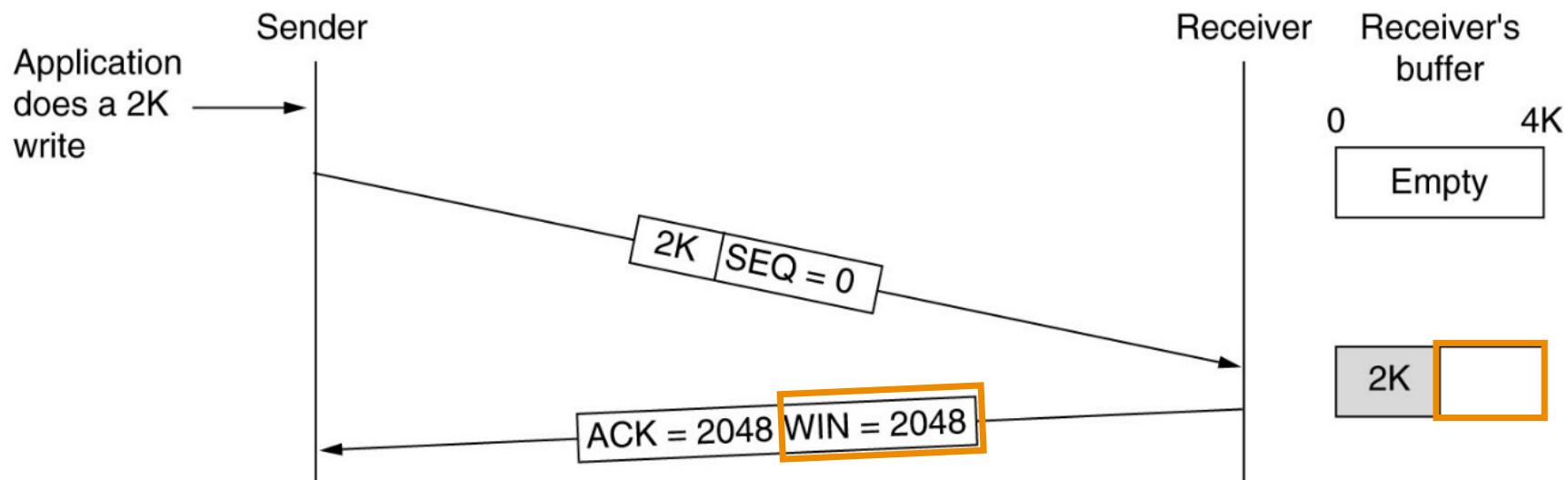
✓ Timestamp

Any outgoing packet with a timestamp will cause the ACK to carry a timestamp echo with the same value. Can be used to calculate round trip time



- ✓ Used in flow control
- ✓ **Variable-sized Sliding Window**

WND SIZE



Window Size

- **Window Management:** not tied to ACKs
(differs from data link protocols)
- Receive entity will **advertise** window segment that it can receive & buffer
- Each entity can **alter size** of the other's sending window **dynamically** using the segment's Window field.

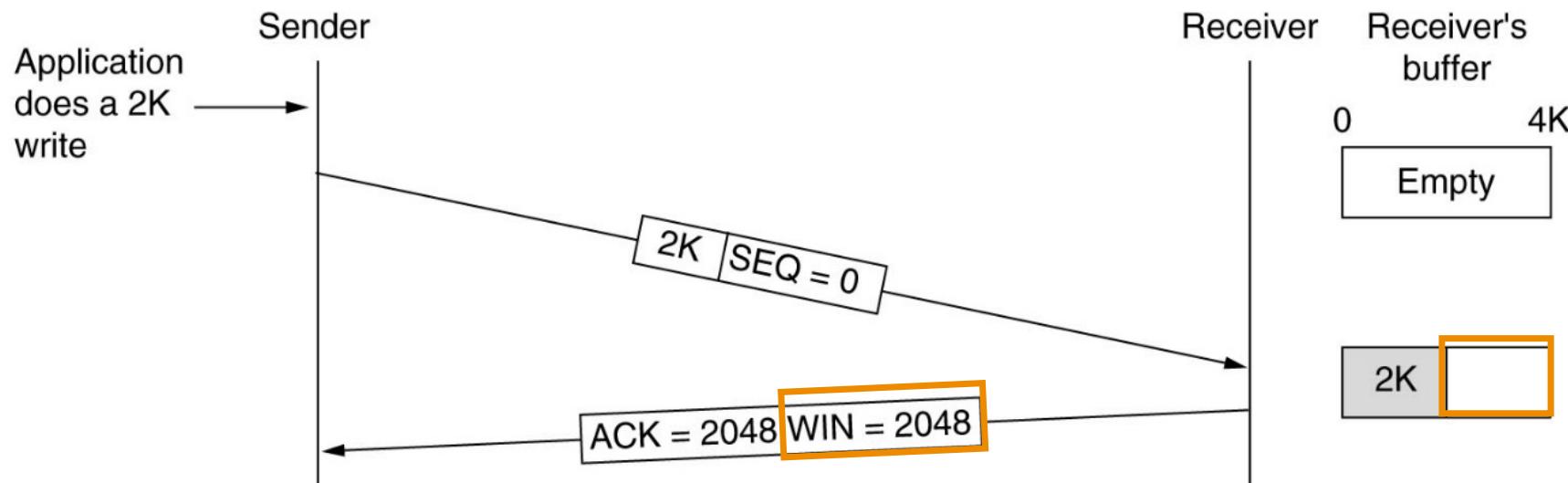
DYNAMIC
BUFFER
MANAGEMENT

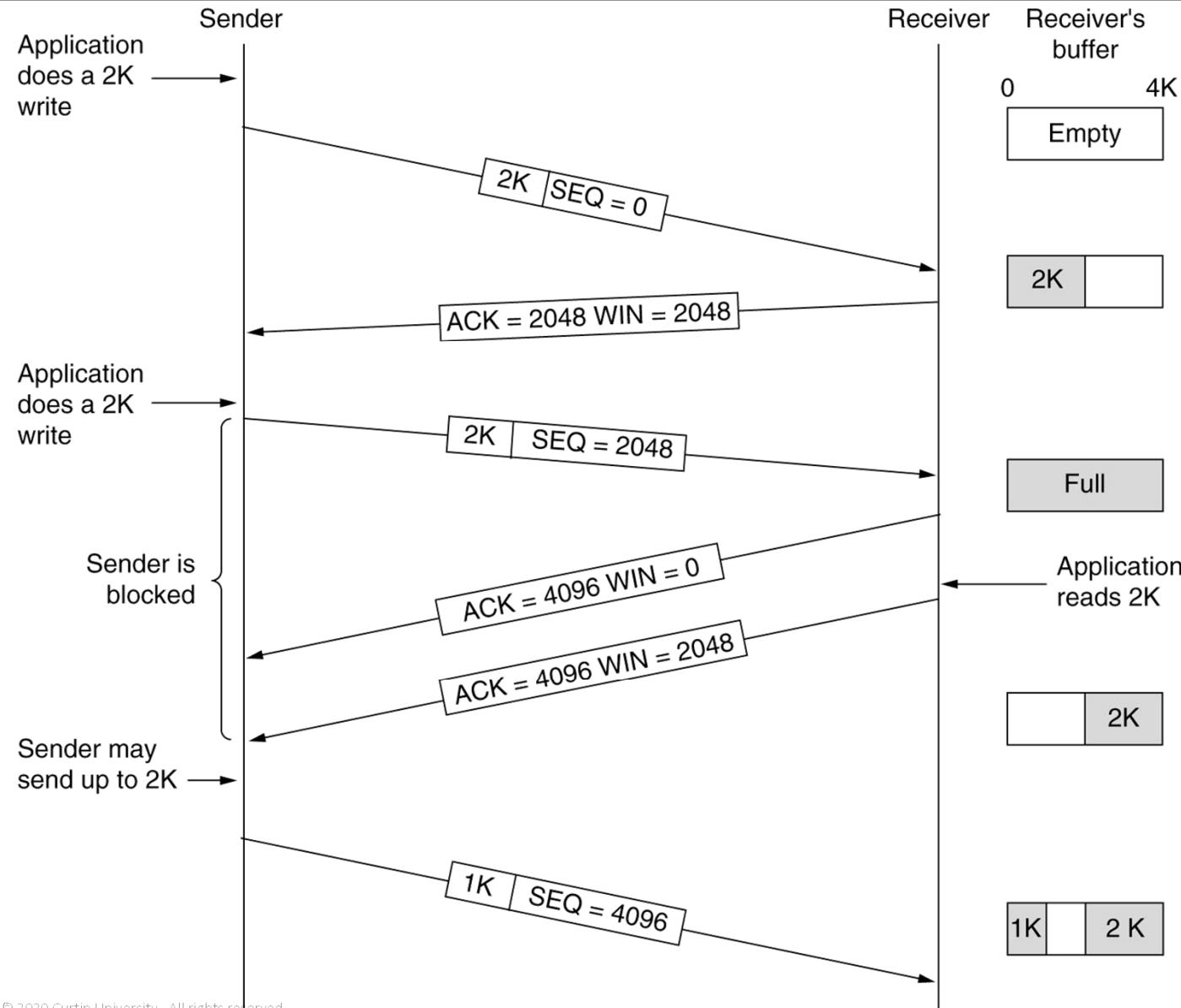
TCP: Flow Control

Each entity implement flow control using a **credit mechanism**, also called a **window advertisement**.

A **credit specifies** the maximum number of bytes the entity sending this segment can receive and buffer from the other entity

DYNAMIC
BUFFER
MANAGEMENT





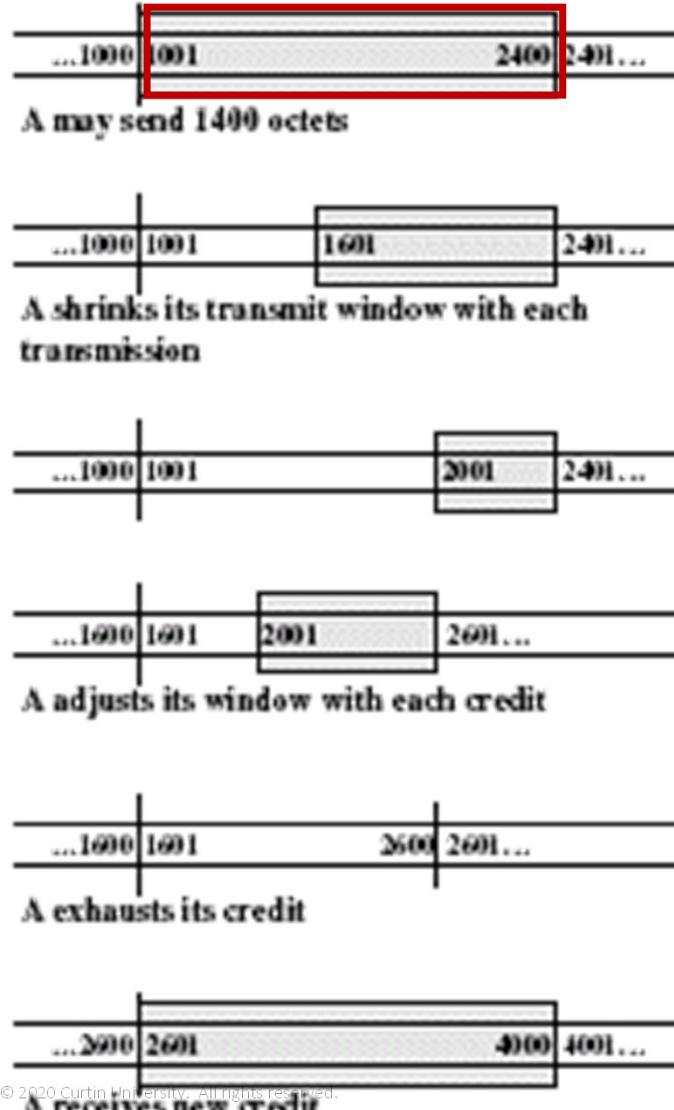
DYNAMIC BUFFER MANAGEMENT



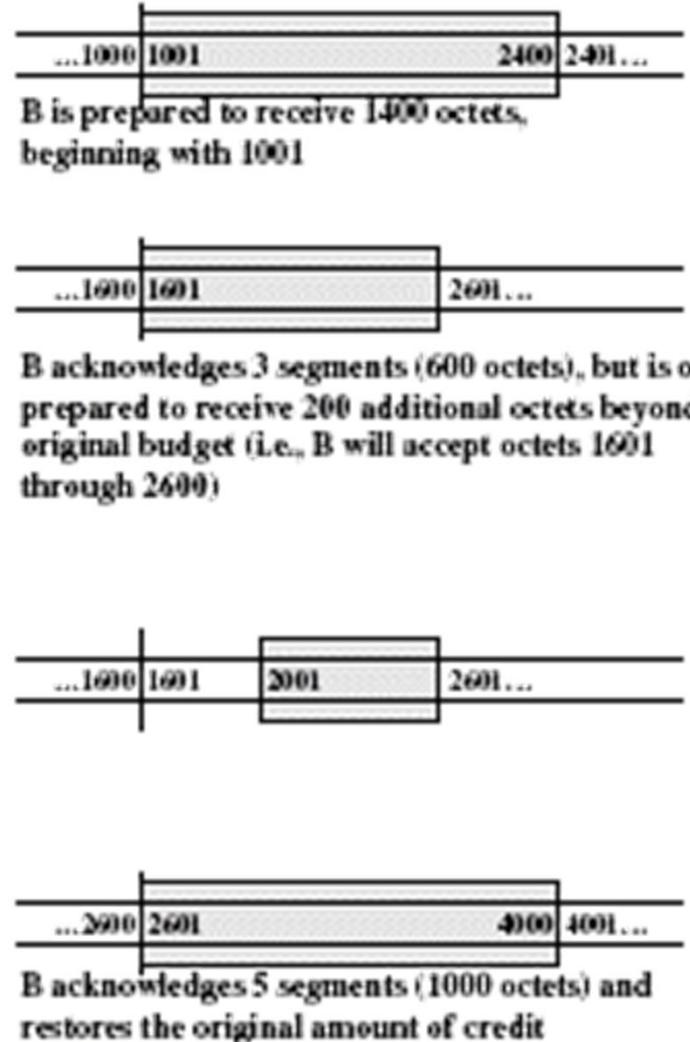
TCP flow control the **sequence number** refers to byte sequence instead of packet (or segment) sequences.



Transport Entity A



Transport Entity B





TCP Implementation Policies

- Send Policy
 - Silly Window Syndrome
- Deliver Policy
- Accept Policy
- Retransmit Policy
- Acknowledge Policy

TCP: Implementation Policies

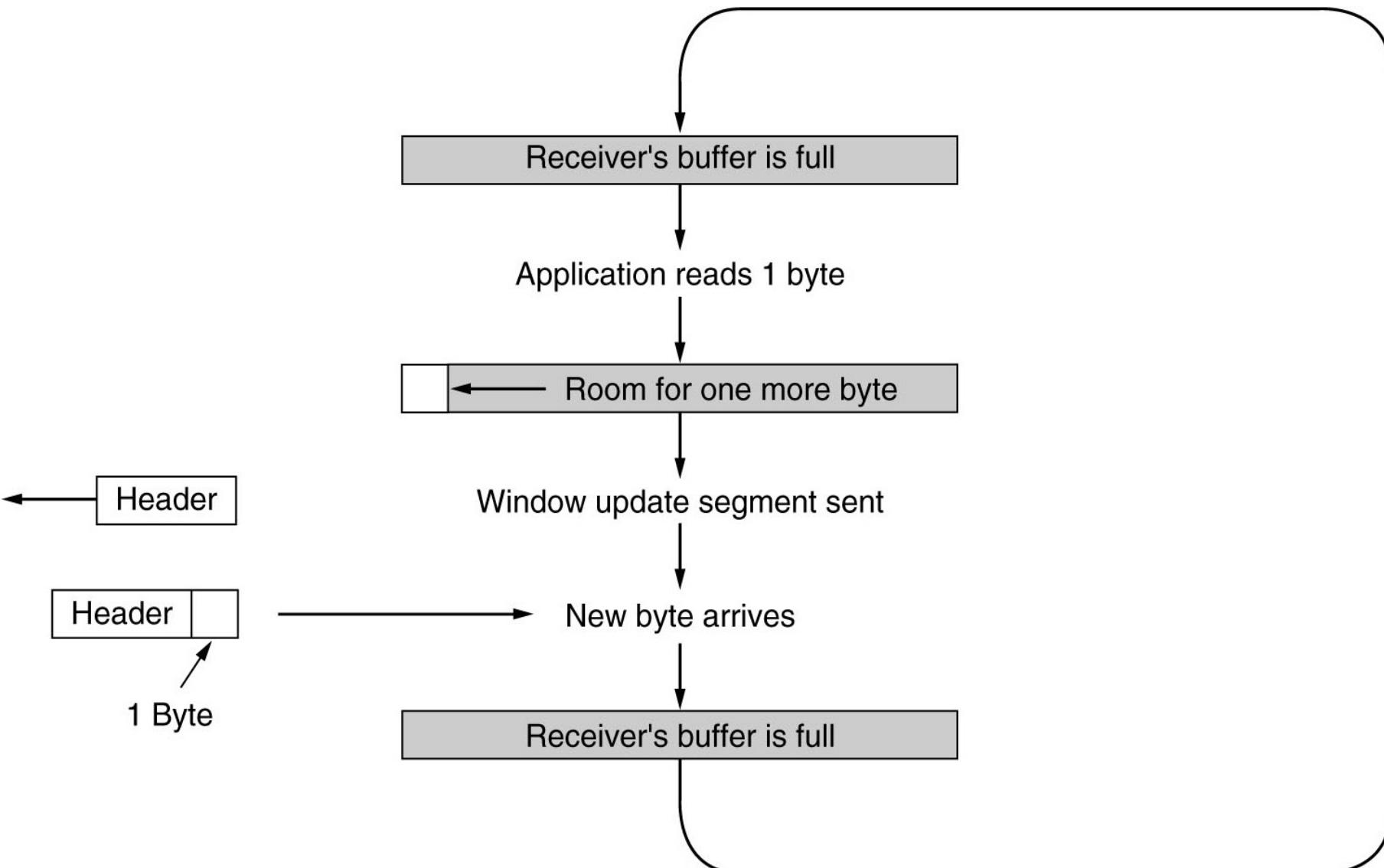
- The standard provide a precise specification of the protocol to be used between TCP entities
- Policies defined for the protocol
 1. **Send policy**
 2. **Deliver policy**
 3. **Accept policy**
 4. **Retransmit policy**
 5. **Acknowledge policy**

1. Send Policy

In the **absence of PUSH** data and a closed transmission window, a sending TCP entity is free to **transmit data at its own convenience**

- depend on performance considerations
 - ✓ if **infrequent and large** (*buffer @ sender*)
 - low overheads
 - slow response
 - ✓ if **frequent and small** (*buffer @ receiver*)
 - quick response
 - high overheads
 - silly window syndrome

Silly Window Syndrome



2. Deliver Policy

Too frequent delivery means too many OS interrupts

- Arriving data are stored in deliver buffer

- ✓ if PUSH flag is set**

Data along with any other data in the deliver buffer are submitted to the destination application in a *RECEIVE* command.

- ✓ if PUSH flag is not set**

TCP may wait, e.g. to avoid excess *interrupts*

- ✓ if URG flag is set**

The receiving application is signaled that urgent data is present

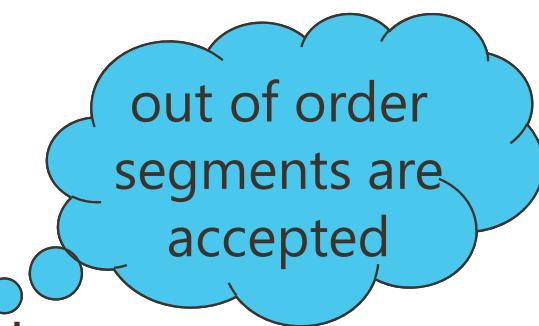
3. Accept Policy

▪ In Order

- ✓ **Discard** out of order segments

▪ In Window

- ✓ **Accept** all segments within the receive window
- ✓ Complex acceptance test and sophisticated storage



4. Retransmit Policy

TCP maintains a **queue of segments** that have been sent but **not ACKed**

- **First Only** suited for **in-window accept policy**

- ✓ one retransmission timer for the entire queue
- ✓ if an ACK is received, the segment/s removed and timer reset
- ✓ if timer expires, first segment in the queue is retransmitted

Selective-
Repeat ARQ

- **Batch** suited for **in-order accept policy**

- ✓ same as above, except when timer expires, retransmit all segments in the queue

Go-Back-N
ARQ

- **Individual** suited for **in-window accept policy**

- ✓ one timer for each segment

Selective-
Repeat ARQ

5. Acknowledge Policy

- **Immediate**

- **Cumulative**

- ✓ wait for an outbound segment, piggyback the ACK
- ✓ Timer to avoid long delay

SUMMARY



▪ Transport Layer

- Fundamentals
- Transport Entity

▪ Transport Layer Elements

- Addressing
- Connection Establishment
- Connection Release
- Flow Control and Buffering
- Multiplexing
- Crash Recovery

▪ TCP

- TCP Header
 - Flags (SYN, FIN, ACK, RST)
 - Flag (URG, PSH) – *in depth*
 - TCP Options
 - Window Size (Dynamic Buffer Management)
- TCP Flow Control
 - Dynamic Buffer Management

▪ TCP - Implementation Policies

- Send Policy
 - Silly Window Syndrome
- Deliver Policy
- Accept Policy
- Retransmit Policy
- Acknowledge Policy



Curtin University

THANK YOU

Make tomorrow better.

Transport Layer II

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 07

Semester 1, 2021



Flow and Congestion Control

- Flow Control
 - Transport Layer
 - Data Link Layer
- Congestion Control
 - Warning-bit
 - Choke-packets
 - Load-shedding
 - RED

Flow Control – Data Link Layer

- Common flow and error control **techniques** at the **Data Link Layer**:

1. **Stop-and-Wait ARQ**

2. **Go-back-N ARQ**

3. **Select-Reject ARQ**

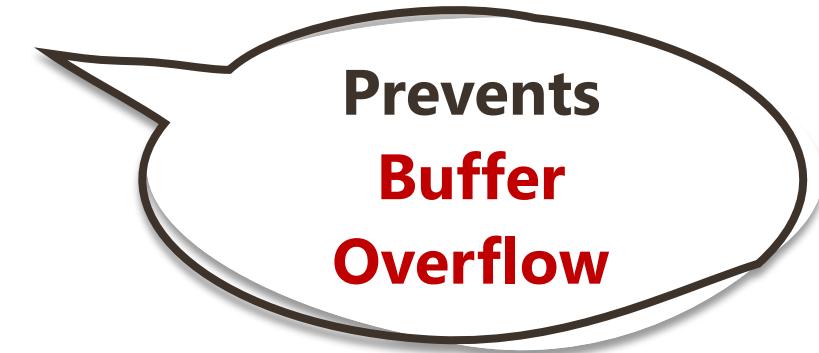


Sliding Window
Methods

Flow Control – Transport Layer

■ Why limit flow?

- Source may send frames/packets at a rate that is faster than the processing speed of the destination host
- The buffer at the destination could be full because
 - ✓ Higher level protocol is not ready
 - ✓ Outgoing I/O port not ready
 - ✓ Destination protocols cannot process PDU as fast



"limits the amount of data being transmitted so that destination host can 'cope' "

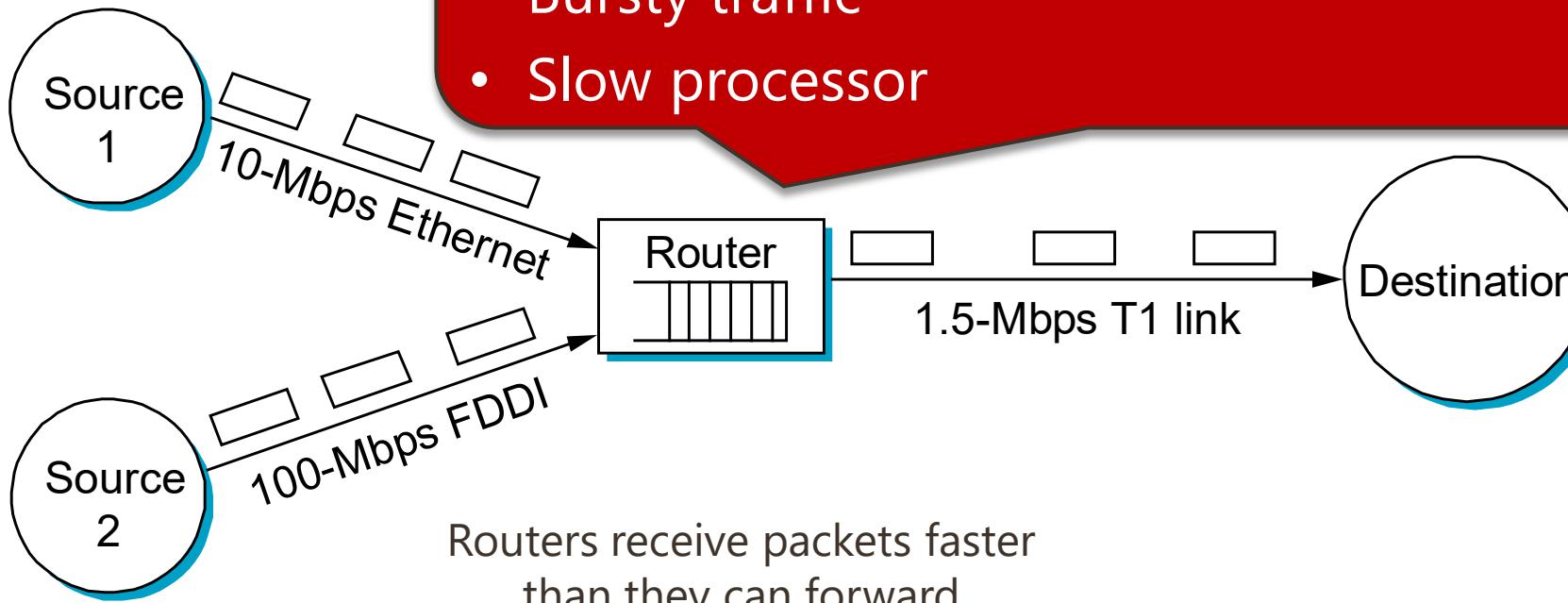
Flow Control – Transport Layer

- Also used Sliding Window methods
- **Decouple Acknowledgement with available buffer**
- E.g., TCP implementation

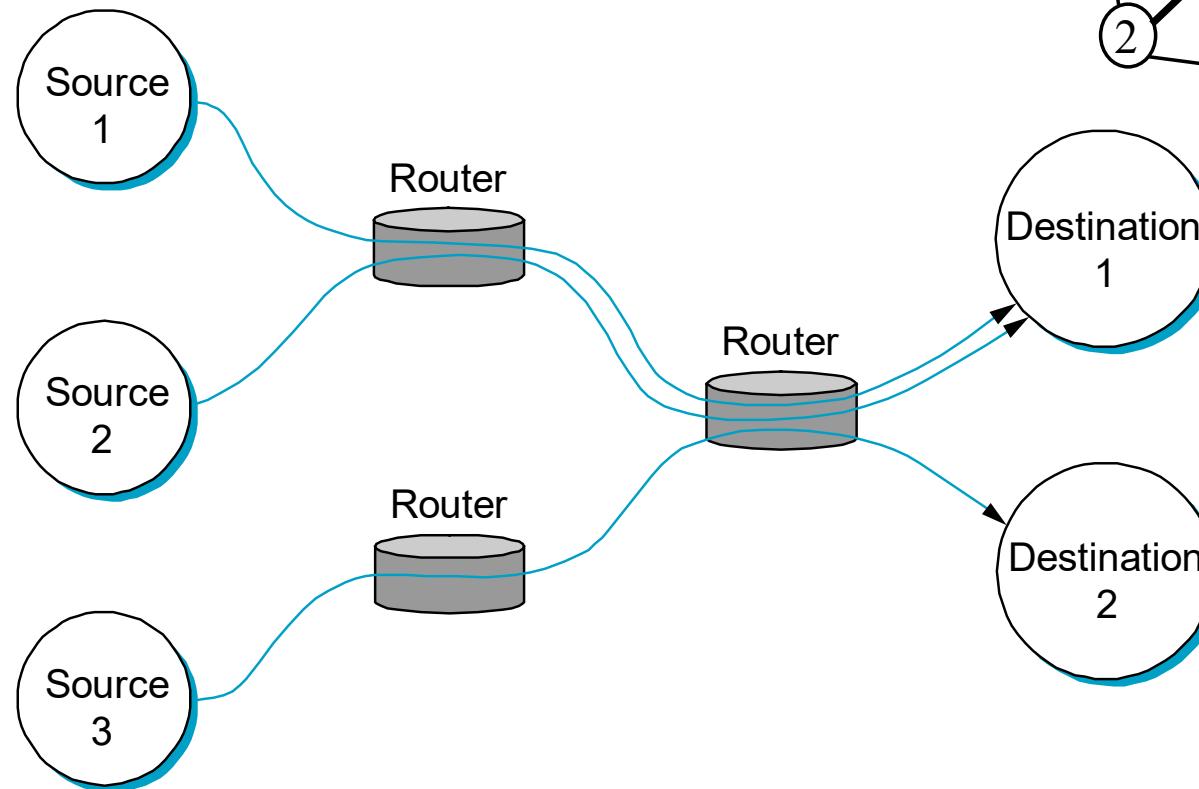
Congestion

- One part of the subnet (e.g. one or more routers in an area) becomes overloaded, **congestion** occurs

- Packet arrival rate exceeds the outgoing link capacity
- Insufficient memory to store arriving packets
- Bursty traffic
- Slow processor



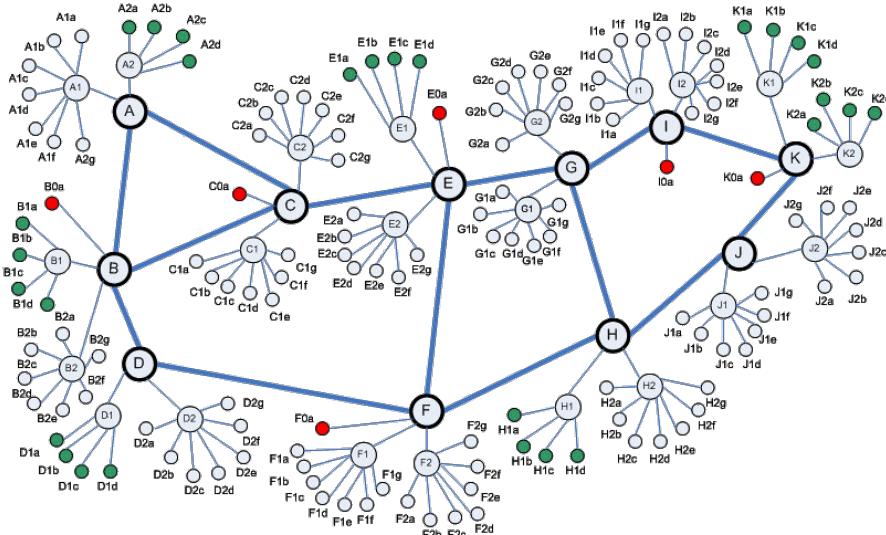
Congestion



Congestion control can be distinguished from **routing** in that sometimes there is no way to '*route around*' a congested router.

Flow vs Congestion Control

- **Congestion Control - Global Issue**

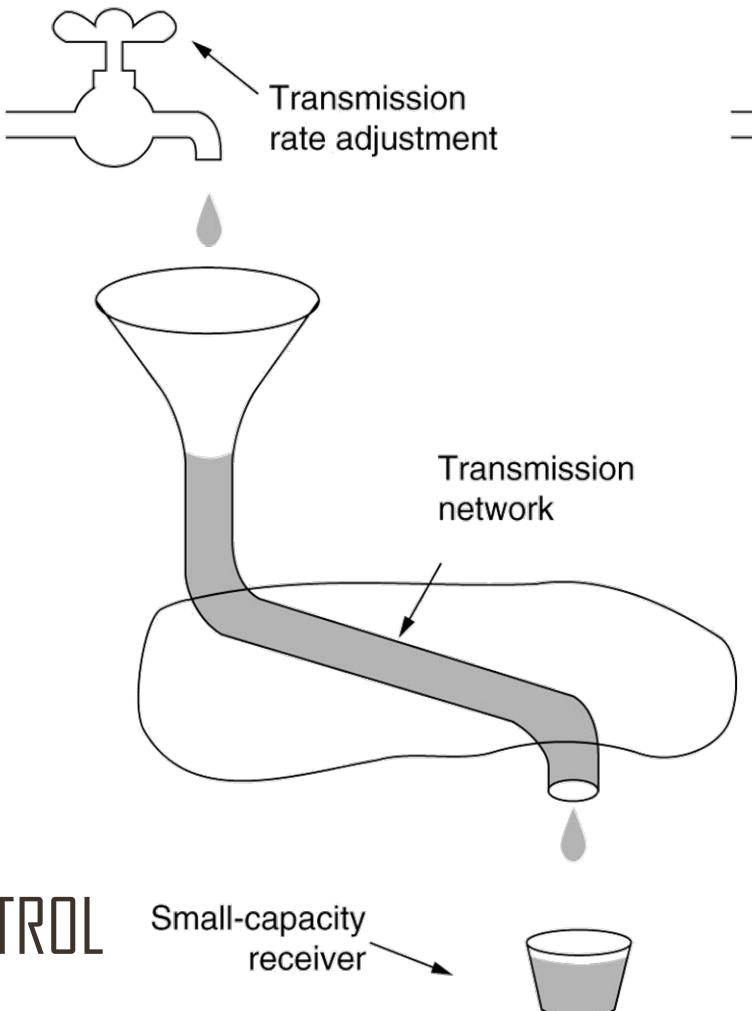


Make sure the subnet is able to carry the offered traffic; **involves** all the **host, routers, store-and-forwarding** processing, etc. within the subnet

1

- **Flow Control – Scope is point-to-point;**
 - involves just **sender** and **receiver**

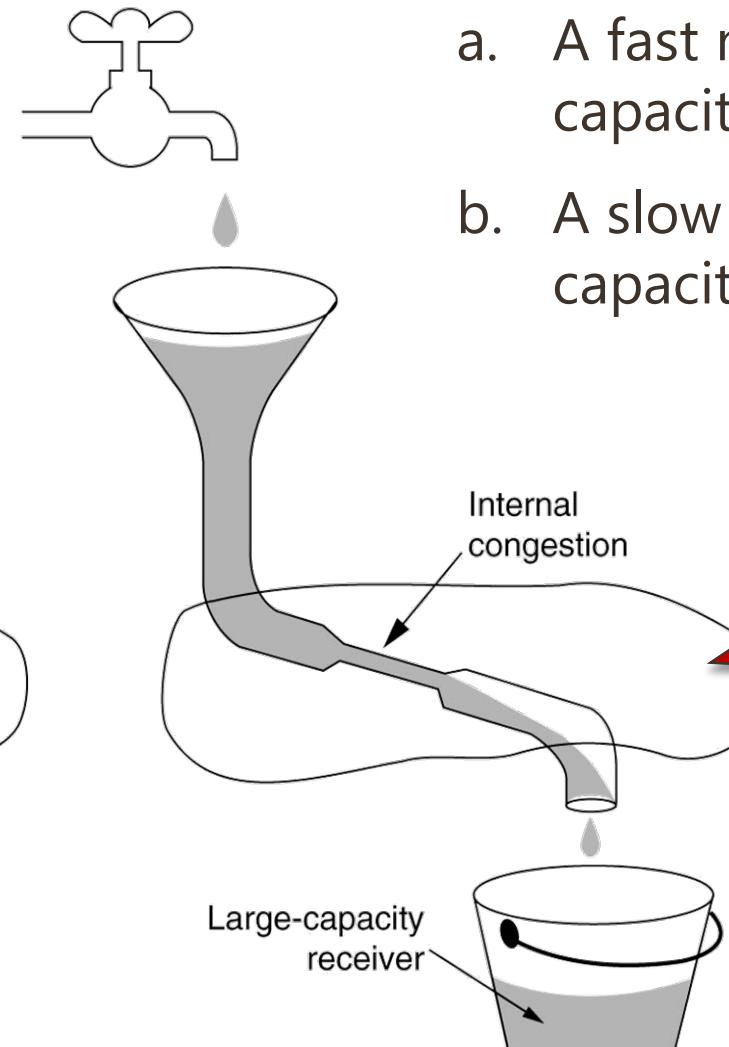
Flow vs Congestion Control



FLOW CONTROL

Small-capacity
receiver

(a)



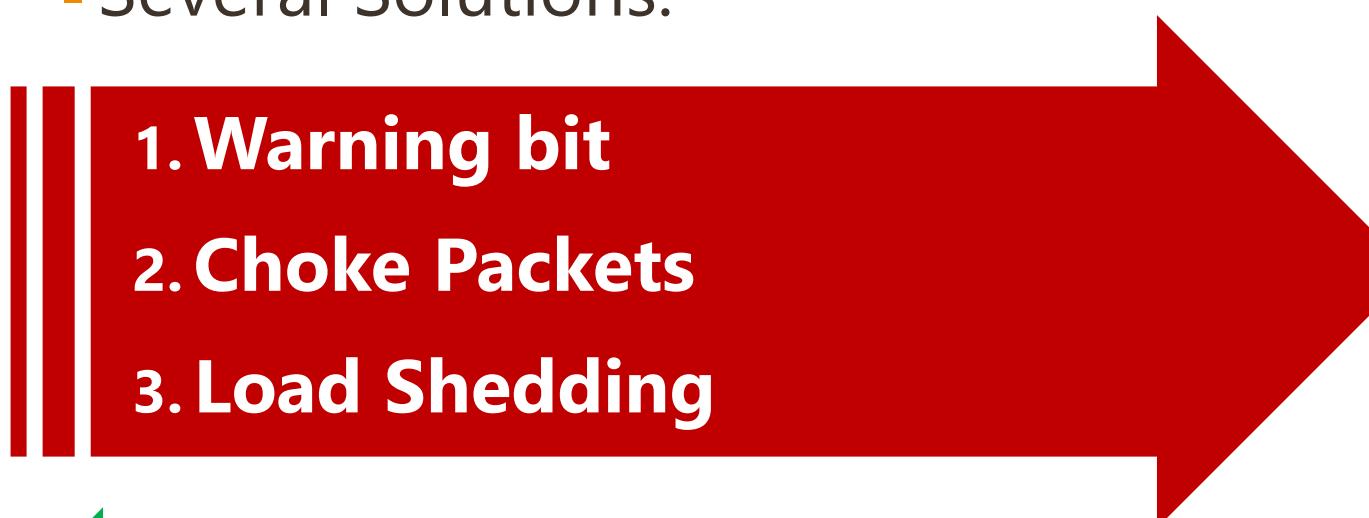
Congestion CONTROL

Large-capacity
receiver

(b)

Congestion Control - Solutions

- Congestion Control is concerned with efficiently using a network at high load
- Several Solutions:

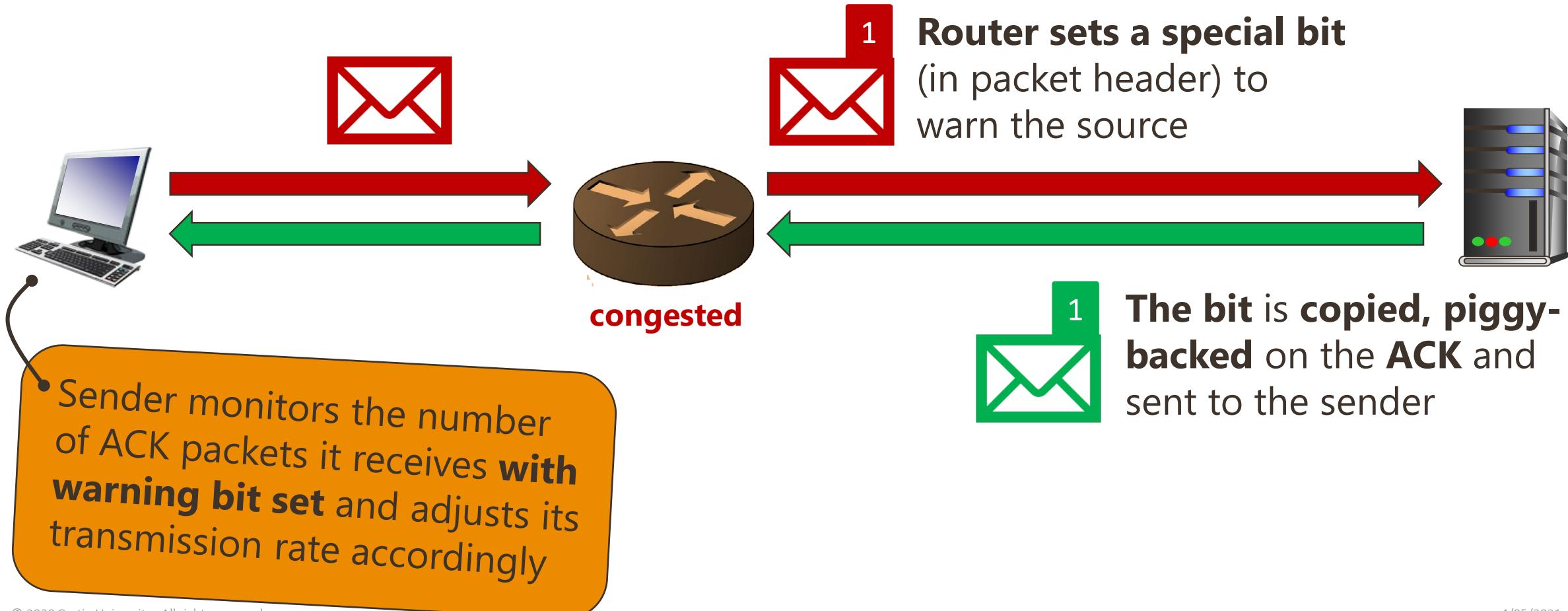
- 
1. Warning bit
 2. Choke Packets
 3. Load Shedding

**Congestion
Detection &
Recovery**

4. Random Early Discard (RED)

**Congestion
Avoidance**

1. Warning Bit



2. Choke Packets

- A **more direct way** of telling the **source to slow down**

ICMP Source
Quench Packet

- A choke packet is a **control packet** generated at a congested node, transmitted to restrict traffic flow

- On receiving the choke packet,
 - Source will **reduce transmission rate** by a



2. Choke Packets - Hop-by-hop

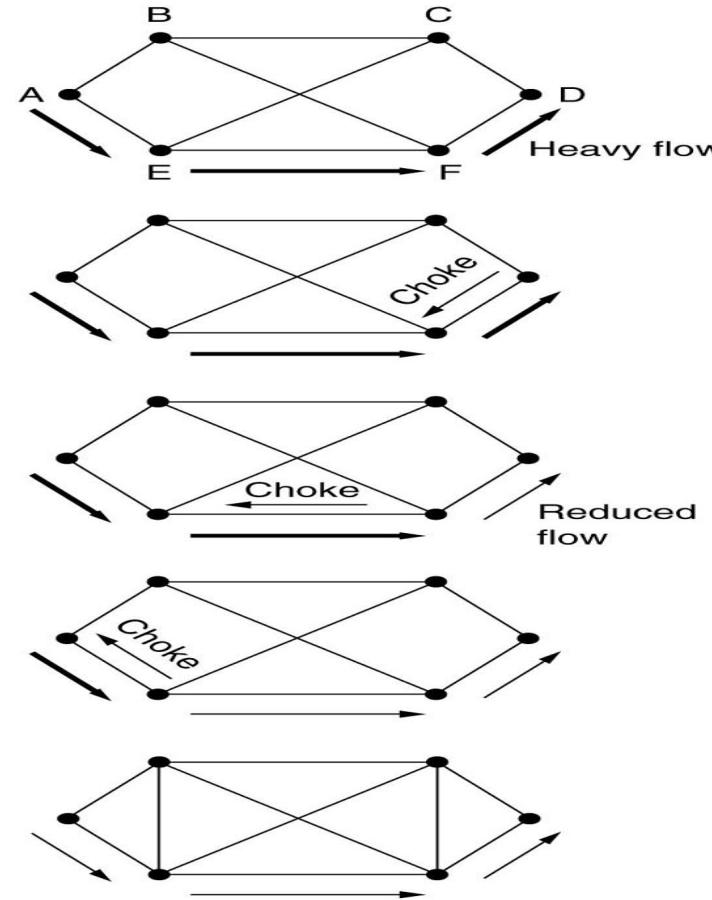
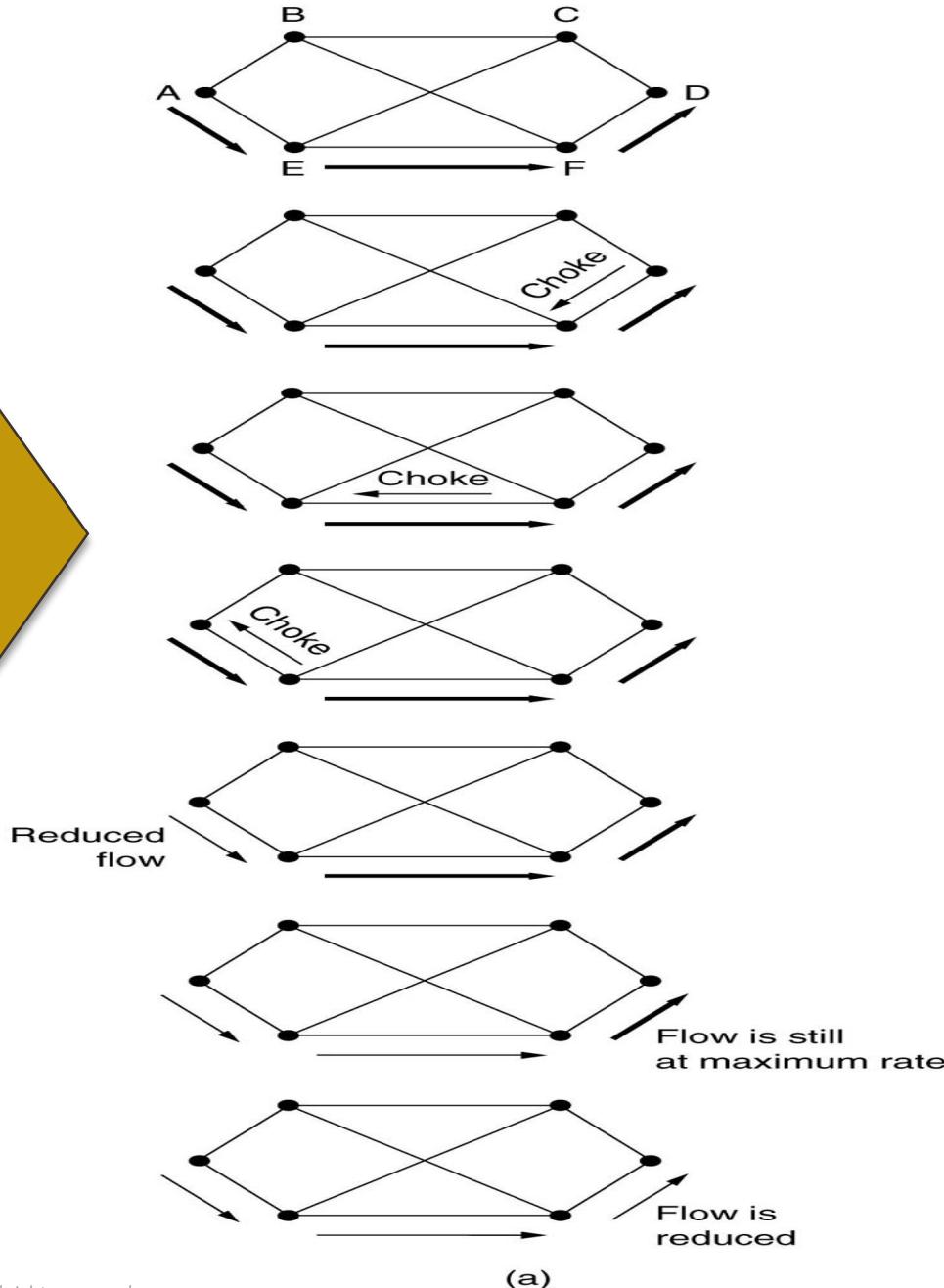
- Over long distances or at high speeds choke packets are not very effective.
- A more efficient method: Send the choke packets hop-by-hop



- This requires each hop to reduce its transmission even before the choke packet arrive at the source

Hop-by-Hop

Typical Choke Packet



Hop-by-Hop Choke

- a. A choke packet that **affects only the source**
- b. A choke packet that **affects each hop** it passes through (Tanenbaum)

3. Load Shedding

- When **buffers become full**, routers simply **discard packets**
- Which packet is chosen to be the victim depends on:

1. Wine or Milk policy

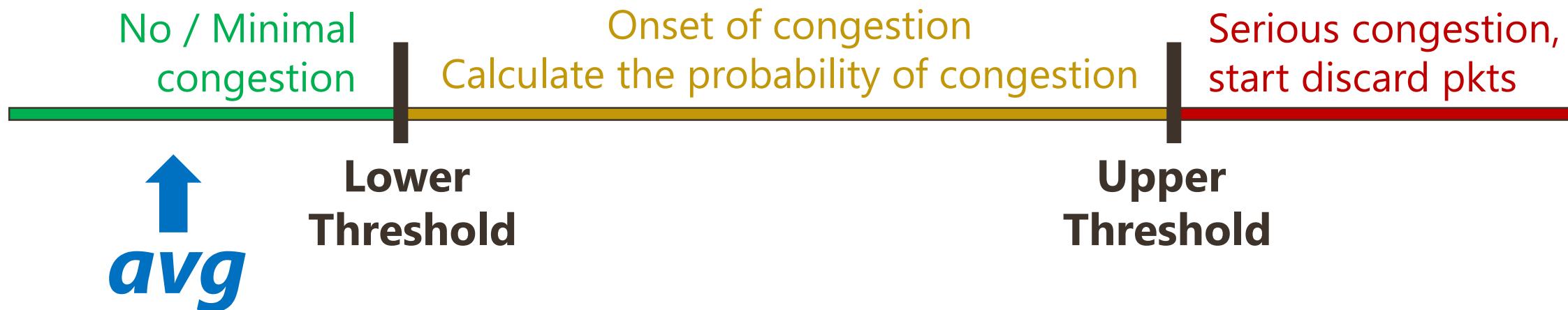


- ✗ File transfer cannot discard older packets: *will cause a gap in the received data.*
- ✓ Real-time voice or video may throw away *old data and keep new packets.*

2. Alternatively, implement an **Intelligent Discard Policy** or get the application to mark packets with discard priority

4. Random Early Discard (RED)

- This is a proactive approach in which the router **discards** one or more packets ***before the buffer becomes*** completely **full**
- Each time a packet arrives, the **RED algorithm** computes the average **queue length**, **avg**





Transport Control Protocol **(TCP)**

- Fundamentals
- TCP Header
 - Flags (SYN, FIN, ACK, RST)
 - Flag (URG, PSH) – in depth
 - TCP Options
 - Window Size (Dynamic Buffer Management)
- TCP Flow Control
 - Dynamic Buffer Management

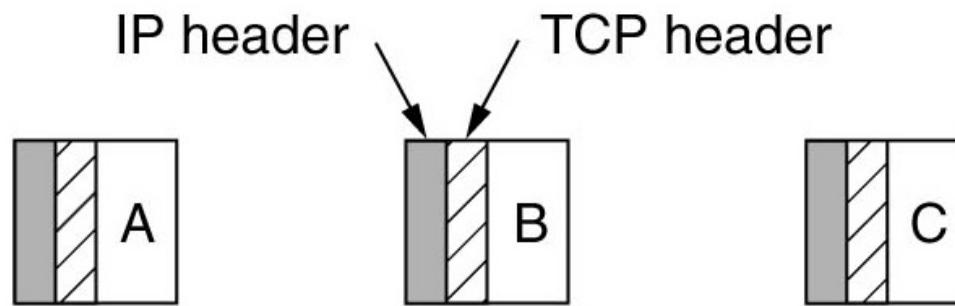
TCP

- The (other) **main** transport **protocol** used in the **Internet**
 - ✓ **Connection-oriented** protocol
 - ✓ RFC 793 (formal), RFC 1122 & 1323 (bug fixes)
 - ✓ Provide a reliable end-to-end communication over an unreliable internetwork
- **Connections** are:
 - ✓ **Full duplex** and point-to-point
 - ✓ A **byte stream** not a message stream



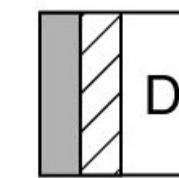
No support for Multicasting or Broadcasting !

TCP: Header



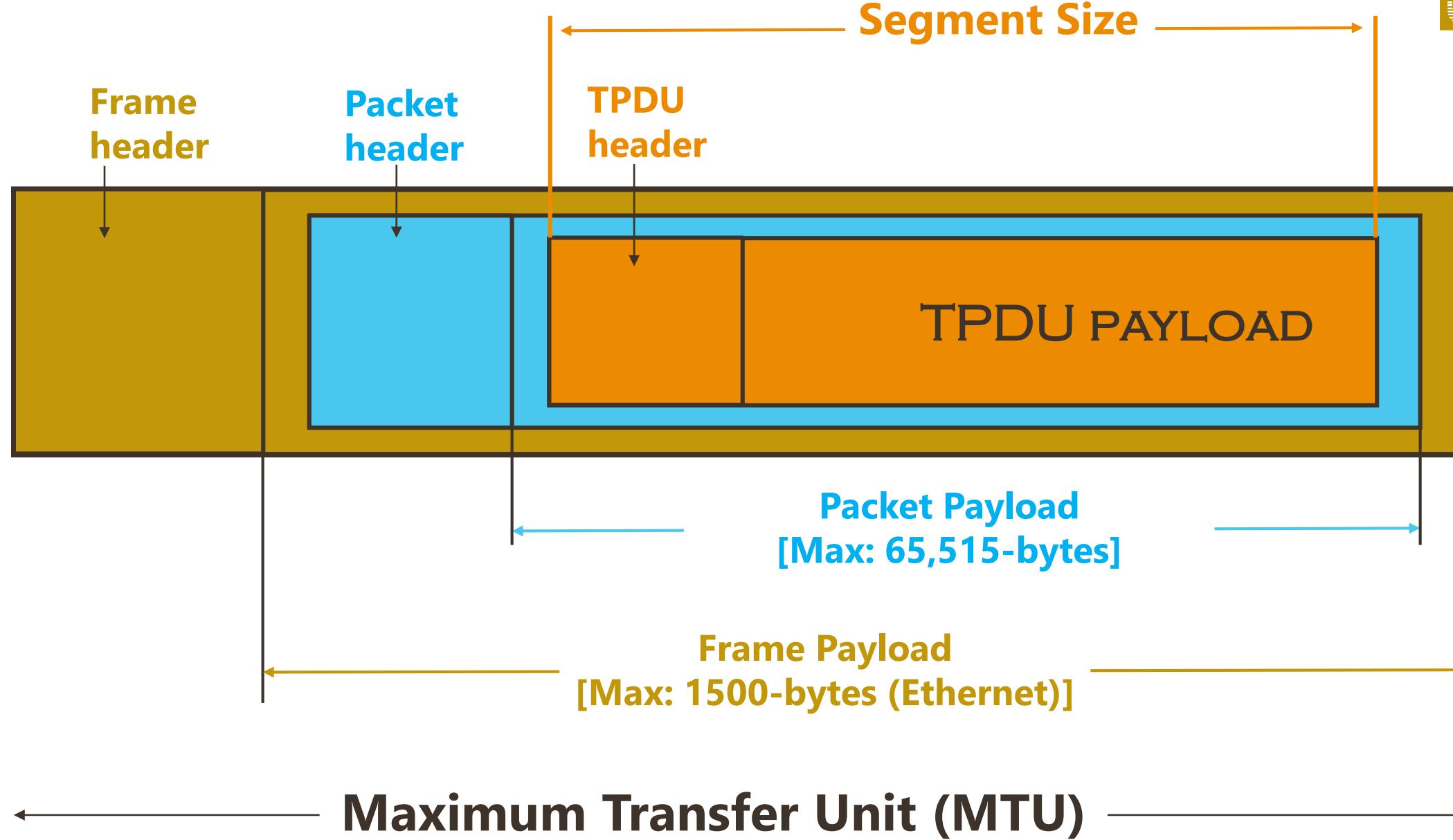
(a)

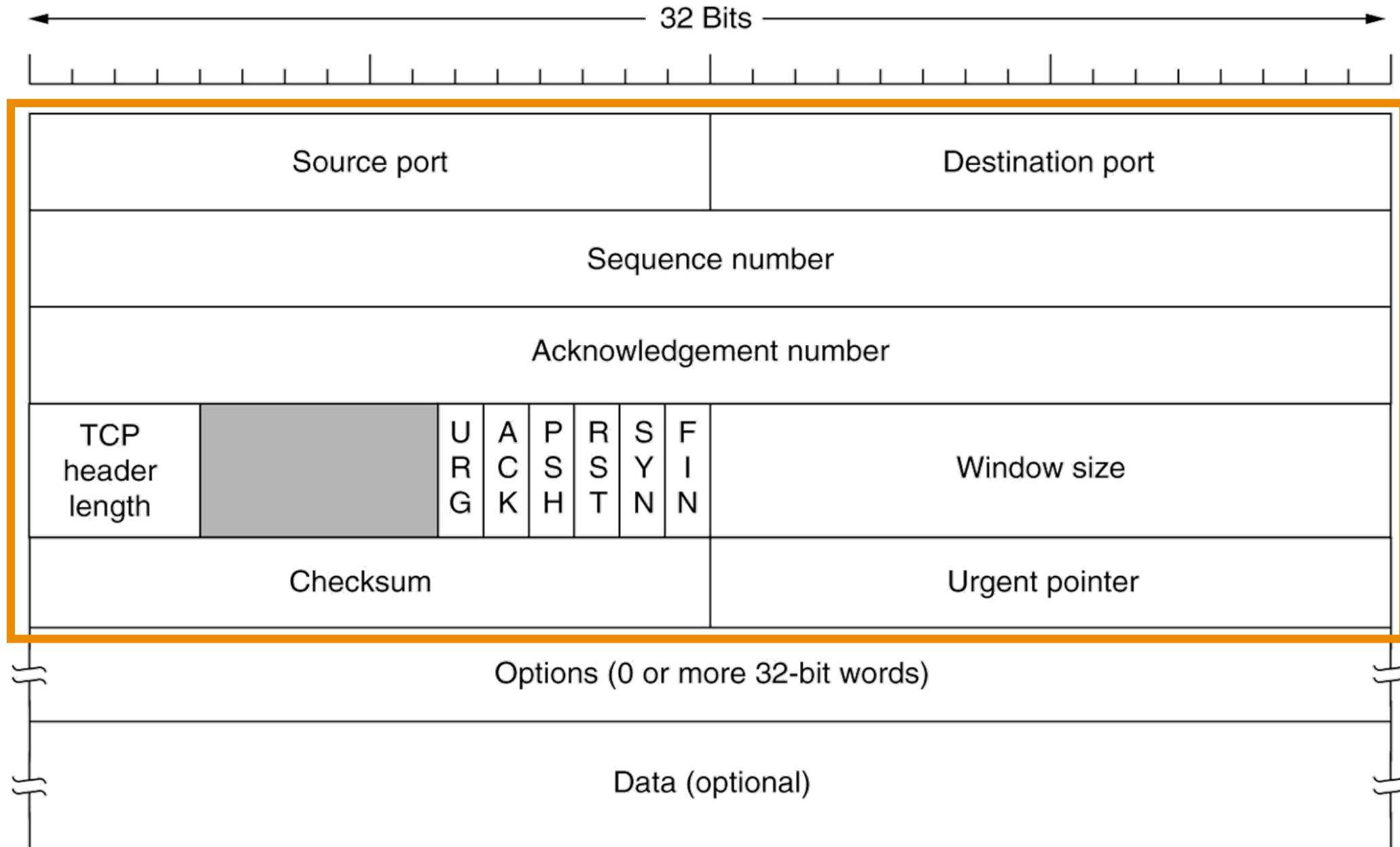
512-byte segments
sent as a separated IP
datagrams



(b)

2048-bytes of data
received to the application
in a single **READ** call





TCP HEADER

Fixed
20-bytes
header

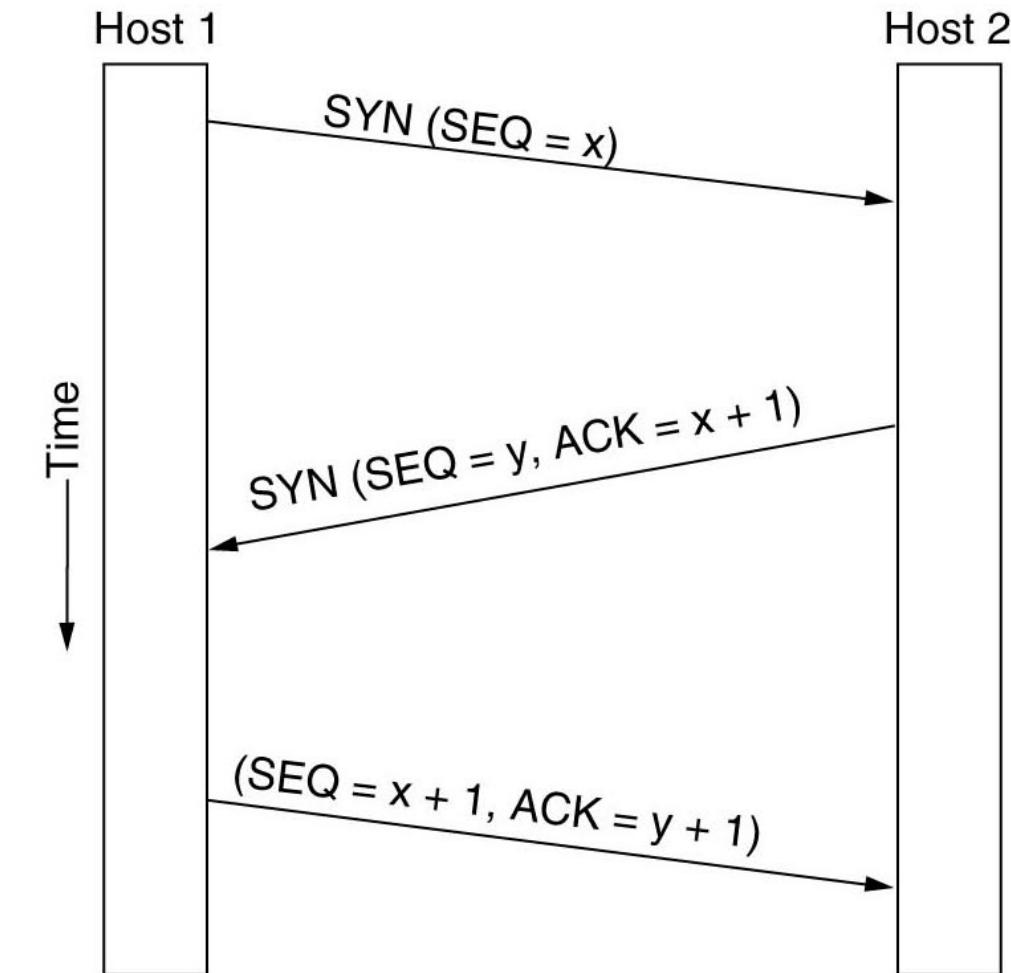
TCP: Header – cont.

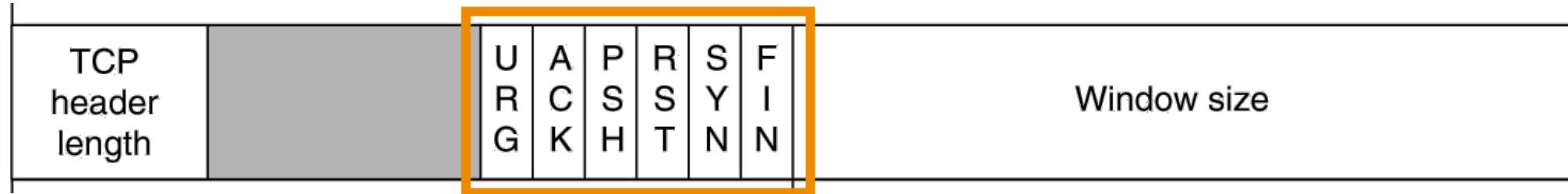
▪ Sequence Number (32-bit)

- sequence number of the first data octet (byte) in this segment
- if SYN is set this field is the initial sequence number (ISN) and the first data octet is ISN+1

▪ Acknowledgement Number (32-bit)

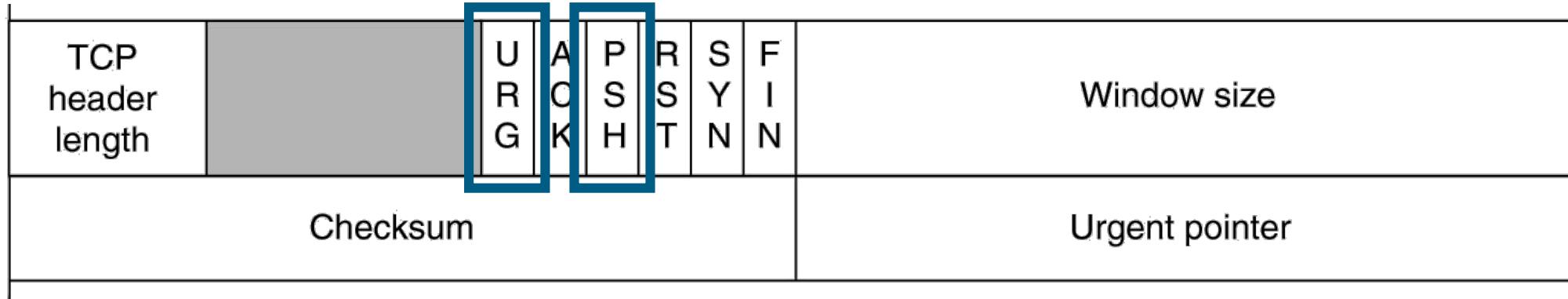
- sequence number of the next data octet the TCP entity expects to receive. May be piggybacked!!
- NOTE that TCP is byte or stream oriented.





- ✓ **URG:** Urgent pointer field significant. Inform the destination TCP user that 'urgent' data is arriving.
- ✓ **ACK:** Acknowledgement field significant.
- ✓ **PSH:** Push function. A TCP user can require TCP to send (receive) all outstanding data up to and including that labelled with a **PUSH** flag.
- ✓ **RST:** Reset the connection.
- ✓ **SYN:** Synchronize the sequence numbers. Used to establish connections.
- ✓ **FIN:** No more data from sender. Used to release connections.

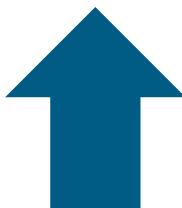
FLAGS



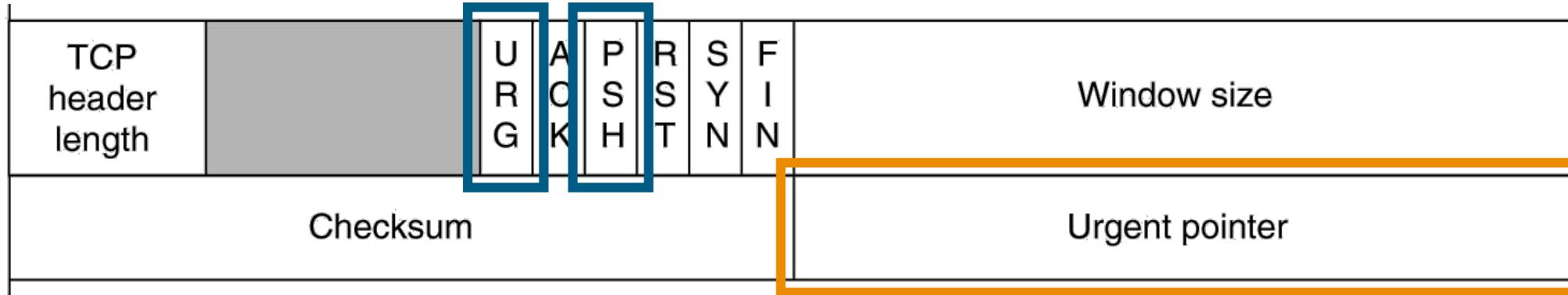
*When application passes data to TCP, TCP may **send it immediately or buffer it (at both sides)**
(in order to collect a larger amount to send at once)*

FLAGS

- **PUSH Flag – (used by application) ->** Force/Flush data out  **at both sides!**
- **URGENT Flag – (used by application) ->** Cause TCP to stop accumulating data and transmit everything it has for the connection immediately



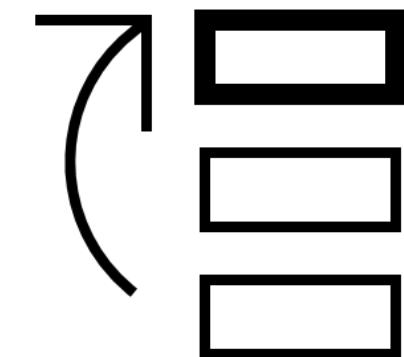
when urgent data is received, receiving application is **interrupted** !
(stops whatever it was doing) & read the data stream to find the urgent data

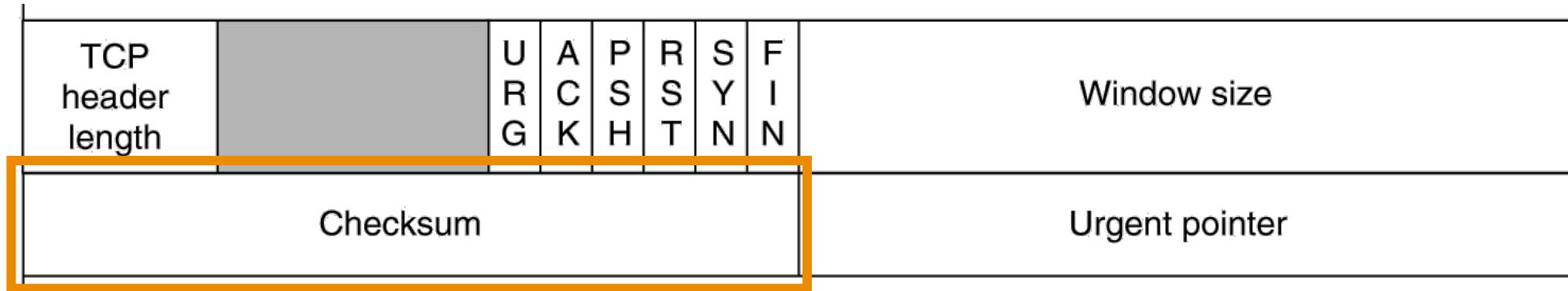


▪ Urgent Pointer

- ✓ **Points to the last octet** in a sequence of urgent data.
Allows the receiver to know how much urgent data is coming

URG POINTER



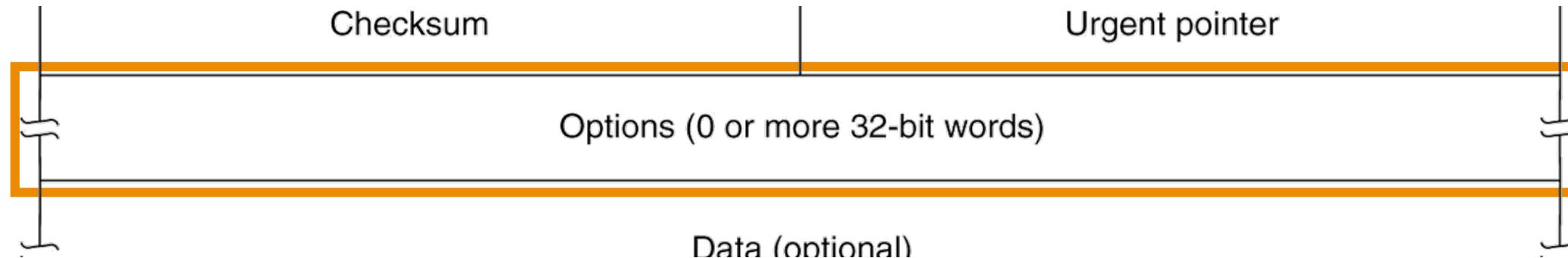


▪ **Checksum:**

- ✓ **header**
- ✓ **data**
- ✓ **conceptual pseudo-header**

CHECKSUM

source & destination
addresses, segment length.
This provides protection
from mis-delivery



- ✓ **Maximum Segment Size (MSS)**

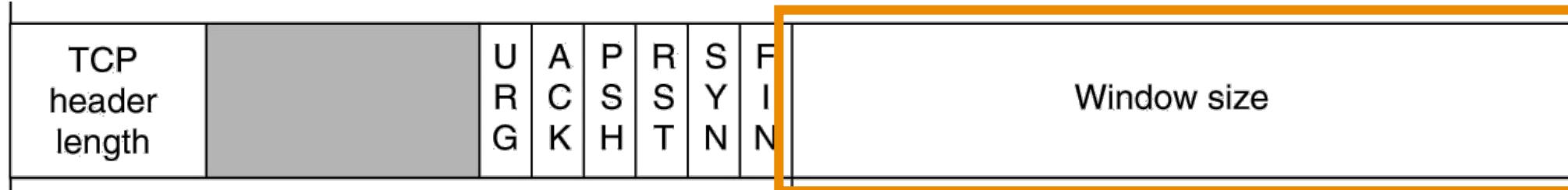
OPTIONS

- ✓ **Window Scale Factor:**

Window is multiplied by 2^F where F is the window scale factor.

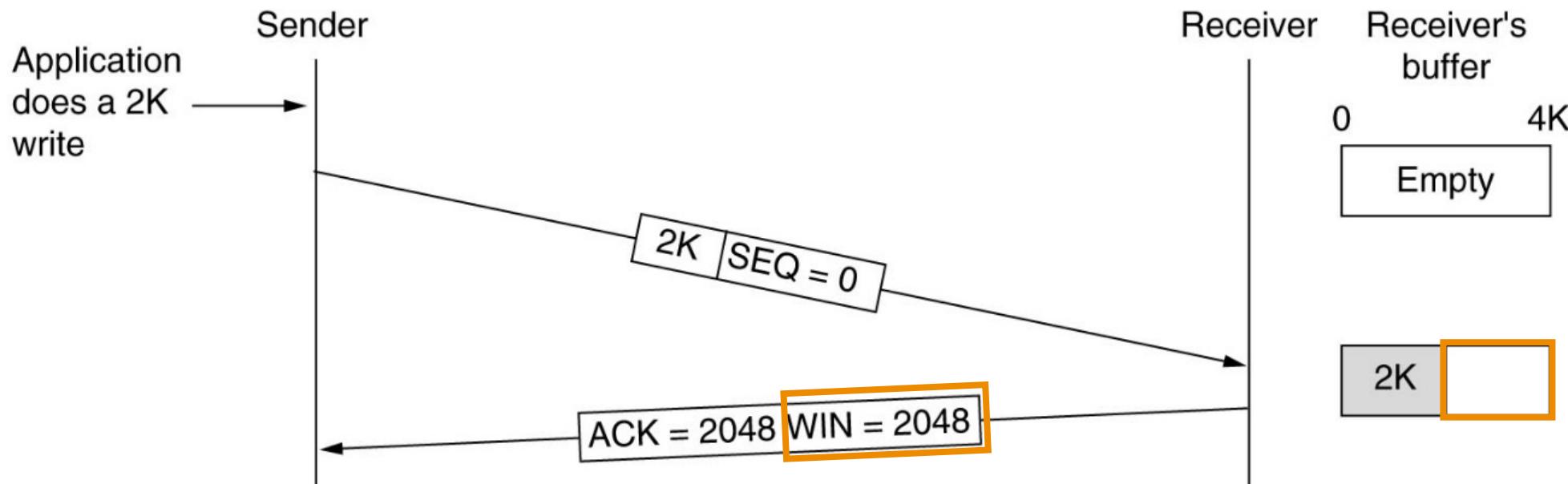
- ✓ **Timestamp**

Any outgoing packet with a timestamp will cause the ACK to carry a timestamp echo with the same value. Can be used to calculate round trip time



- ✓ Used in flow control
- ✓ **Variable-sized Sliding Window**

WND SIZE



Window Size

- **Window Management:** not tied to ACKs
(differs from data link protocols)
- Receive entity will **advertise** window segment that it can receive & buffer
- Each entity can **alter size** of the other's sending window **dynamically** using the segment's Window field.

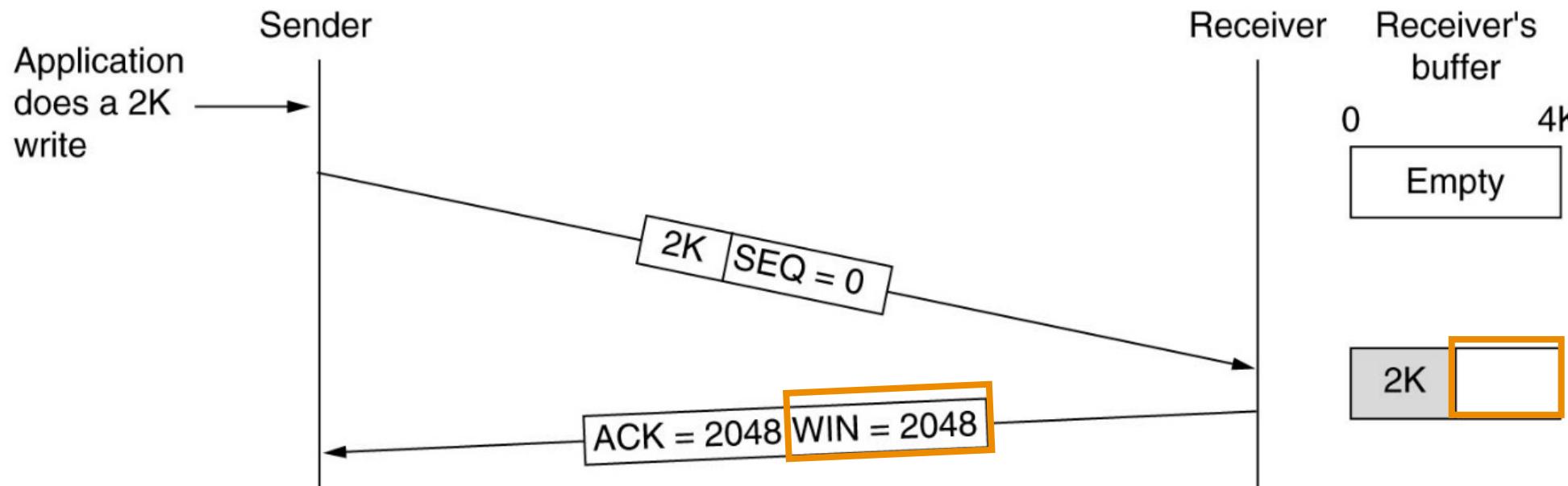
DYNAMIC
BUFFER
MANAGEMENT

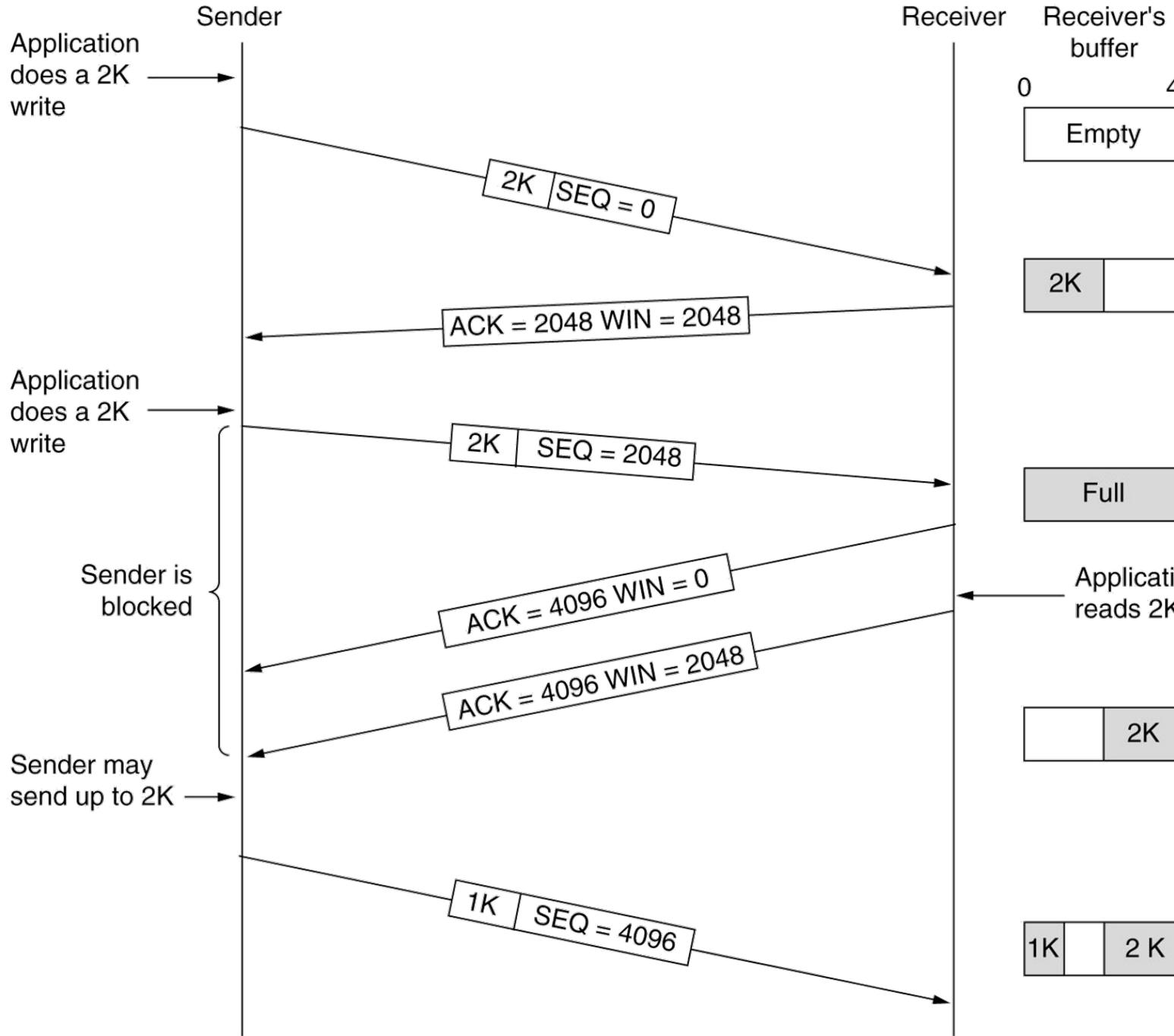
TCP: Flow Control

Each entity implement flow control using a **credit mechanism**, also called a **window advertisement**.

A **credit specifies** the maximum number of bytes the entity sending this segment can receive and buffer from the other entity

DYNAMIC
BUFFER
MANAGEMENT

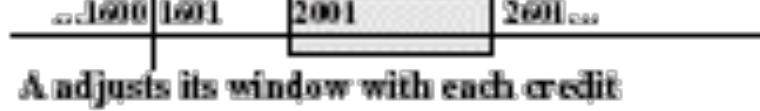
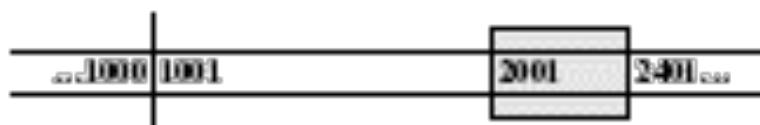
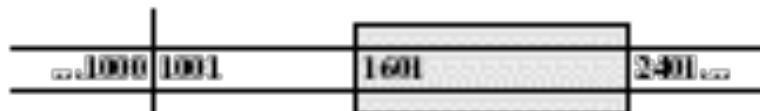
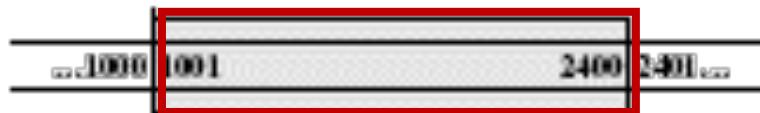




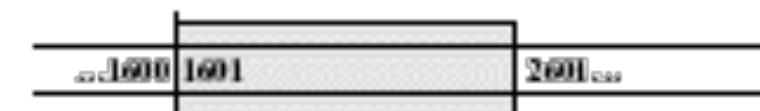
DYNAMIC BUFFER MANAGEMENT



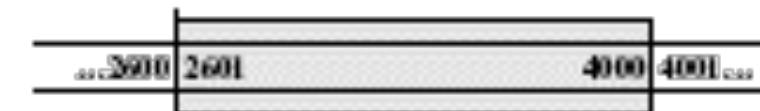
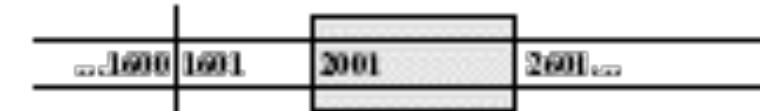
TCP flow control the **sequence number refers to byte sequence** instead of packet (or segment) sequences.

Transport Entity A**Transport Entity B**

B is prepared to receive 1400 octets, beginning with 1001



B acknowledges 3 segments (600 octets), but is only prepared to receive 200 additional octets beyond the original budget (i.e., B will accept octets 1601 through 2600).



B acknowledges 5 segments (1000 octets) and restores the original amount of credit



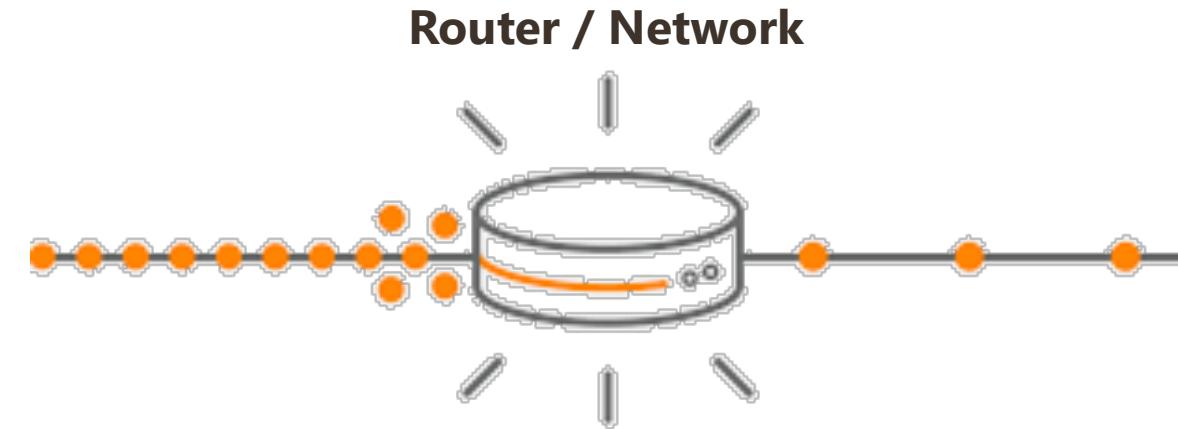
TCP Congestion Control

- Network layer *also tries* to manage congestion,
but difficult

- TCP does the **heavy lifting**



**“The law of conservation of packets:
cannot inject new packets into the
network until the old one leaves”**



TCP Congestion Control

- **TCP dynamically manipulates the window size**

- ✓ Detecting congestion
- ✓ Prevent congestion (try)
- ✓ React to congestion

- **Detecting Congestion, Difficult?**

- Old days  : **transmission error or packet discard** 
- Nowadays  : **most transmission timeouts** on the Internet are due to congestion 

Prevent Congestion

“ultimately, congestion can only be controlled by limiting the total amount of data entering the internet to the amount that the internet can carry”

- In Data Link Control Protocol

- ✓ Sliding window mechanism helps **to pace the sender**

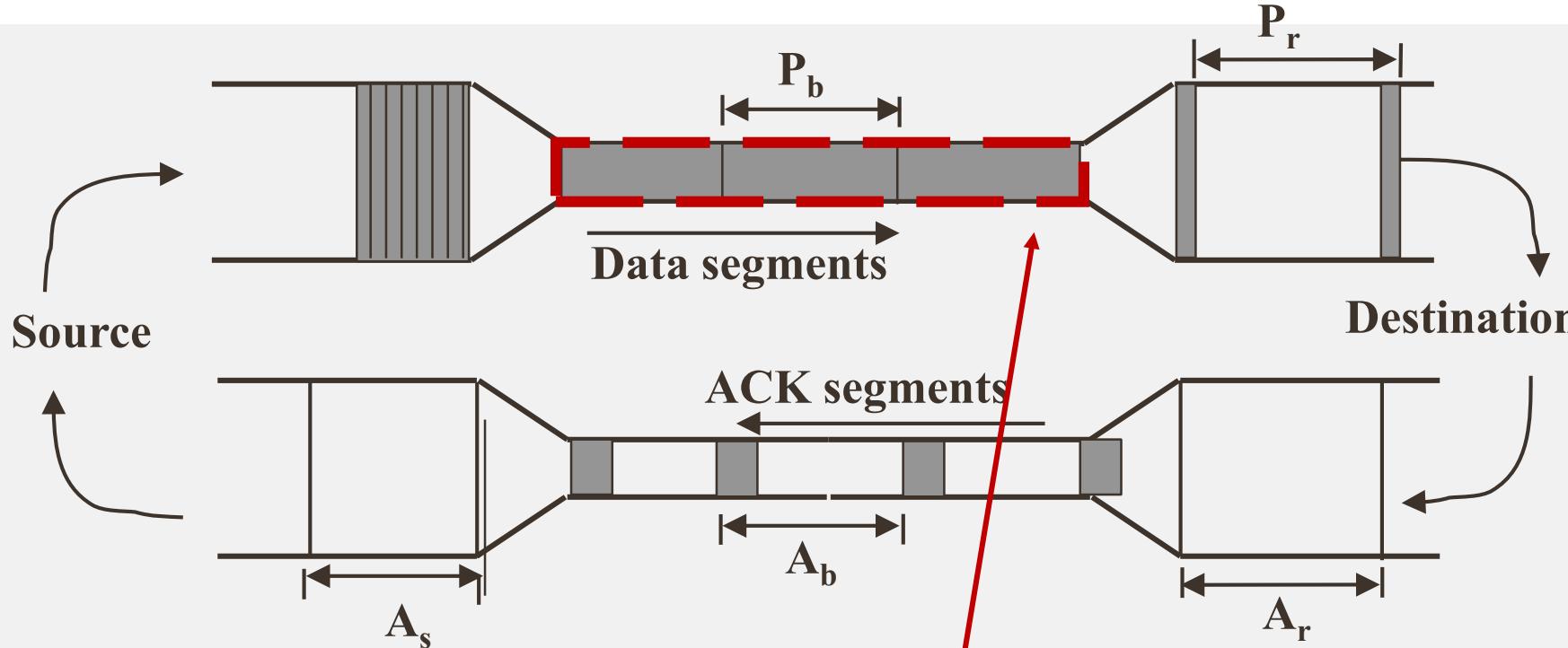
- In TCP

- same **pacing effect** as data link protocol
 - **Data Sending Rate = Ack Arrival Rate**, once any initial credit is used up



Flow Determined by Network

- P_b = time of minimum segment spacing on the slowest link



Rate of ACK arrival is dependent on round trip time

1. bottlenecks in the **network**
2. bottlenecks at the **destination**

TCP Segment Pacing

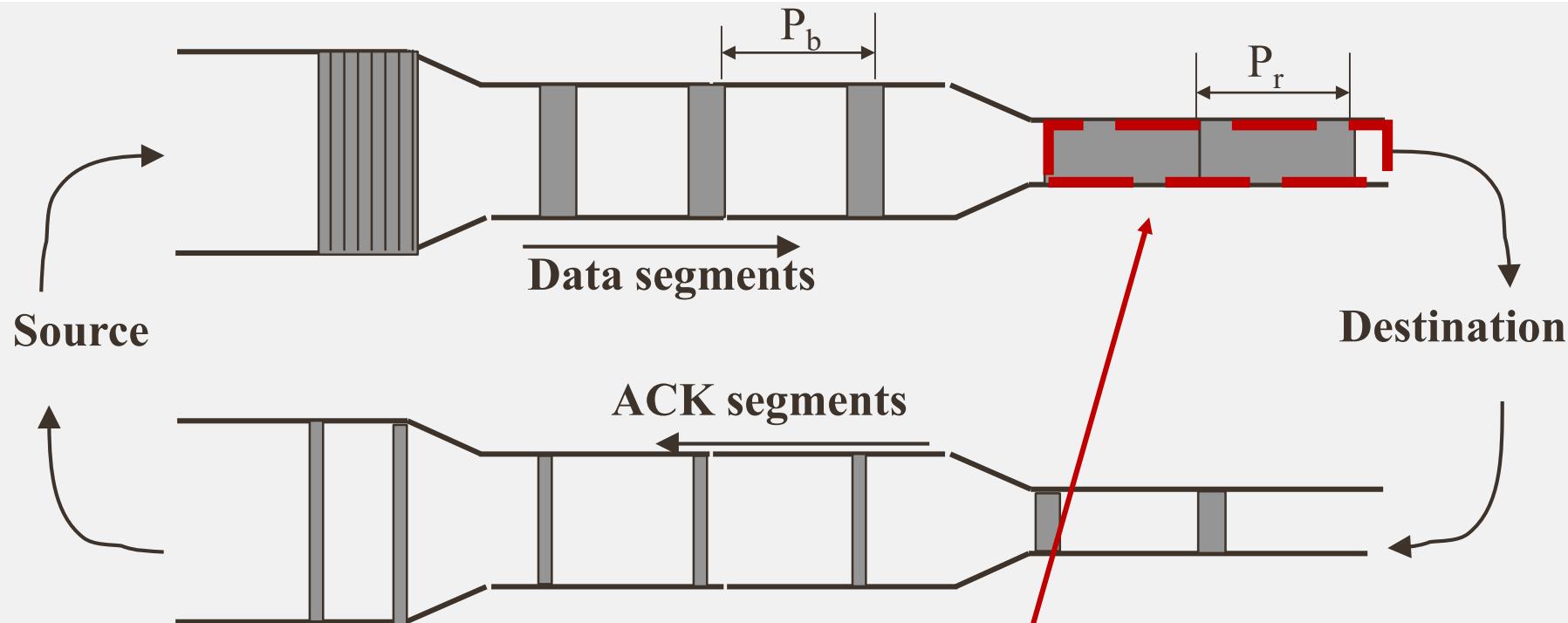
- The **thickness** of the pipe is proportional to the data rate
- Source and destination are on **high capacity networks**
- **Lower speed links create bottlenecks**
- Segment represented by rectangle areas - spreads out if data rate is low
- The wider spacing is preserved at the destination even though the data rate on the final link is high, therefore the segment spacing at the receiver is: **Pr = Pb**

TCP Segment Pacing – cont.

- In **steady state**,
 - ✓ sender's segment rate = arrival rate of ACKs
 - ✓ sender's rate is determined by the slowest link on the path
- Sending TCP entity automatically **senses and regulates flow**
 - ✓ self-clocking
- Self-clocking works equally well with the bottleneck at receiver
 - ✓ ACKs are sent out at a rate equal to the absorption capacity of the destination

Flow Determined by Destination system

- P_b = time of minimum segment spacing on the slowest link



Rate of ACK arrival is dependent on round trip time

1. bottlenecks in the **network**
2. bottlenecks at the **destination**

TCP Congestion - Solutions

1. Segment Pacing Effect – *discussed before*
2. Slow Start
3. Dynamic Window Sizing on Congestion

2. Slow Start

$$\text{awnd} = \min\{\text{credit}, \text{cwnd}\}$$

- **Each sender maintains two windows**
 - **awnd** - Window the receiver has granted
 - **cwnd** - Congestion window
 - each window reflects the number of bytes the sender may transmit
 - **credit** - amount of unused credit granted in the most recent ACK in segments

2. Slow Start – cont.

- **After connection is established**

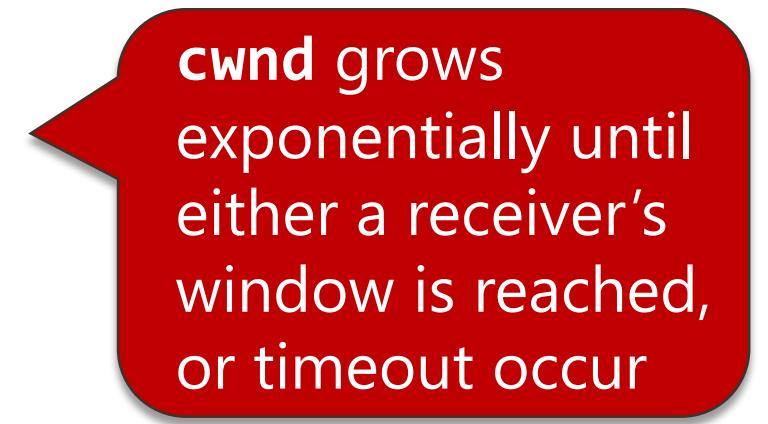
- sender initializes **cwnd** to the size of the maximum segment (normally **cwnd=1**) and sends one maximum segment
- if segment is ACKed before timeout, **cwnd** is increased to two maximum segment size and two segments are sent

- **As each segments is ACKed,**

- the **cwnd** is increased by one maximum segments size

- **The idea:**

- if bursts of size **n** bytes work fine but a burst of **n+n** bytes gives timeout, the **cwnd** is set to n bytes to avoid congestion



cwnd grows exponentially until either a receiver's window is reached, or timeout occurs

3. Dynamic Window Sizing on Congestion

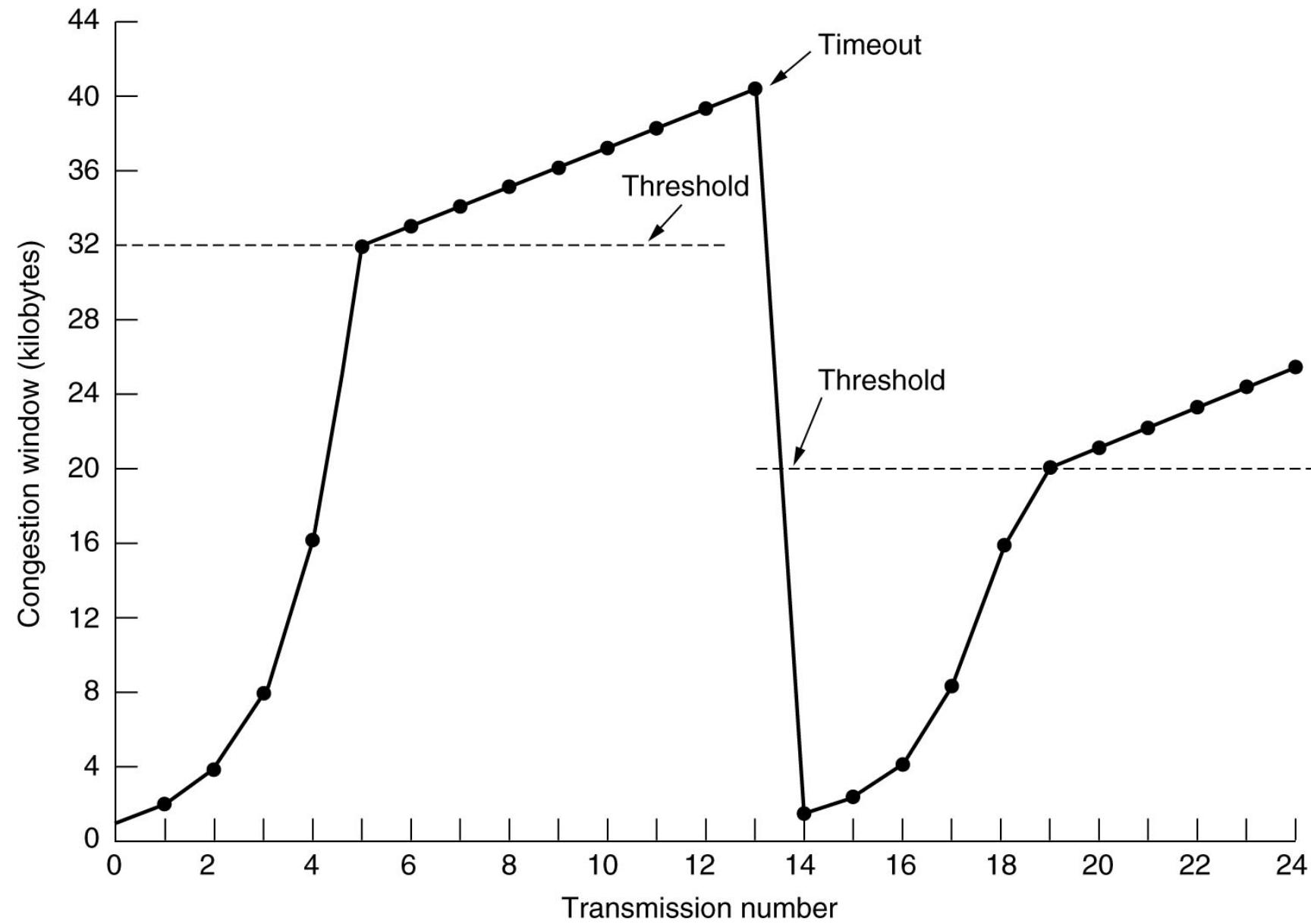
- An internet congestion control algorithm
- if **congestion causes** a timeout, **cut back flow, ramp up slowly**

- **when timeout occurs**



- **ssthresh** = slow start threshold
- set **ssthresh** = **cwnd**/2
- set **cwnd=1** and **performs slow-start until cwnd = ssthresh**
- from then on **linear grow**, increase **cwnd by 1** (segment) **after every ACK** until the receiver's window is reached or timeout occur

Dynamic Window Sizing



TCP Timer Management

- TCP uses multiple timer (at least conceptually) to do its work.
- Retransmission timer
 - How long should the timeout interval be?
- Determine the Round-Trip Time (**RTT**), time between sending a segment and receiving its ACK, is tricky.
- Even if **RTT** is known, deciding the timeout interval is also difficult
 - if timeout is set too short (say T1) - unnecessary retransmission
 - if timeout is set too long (say T2) – performance suffer due to long retransmission delay

TCP Timer Management – cont.

- Highly dynamic algorithm that constantly adjust the timeout interval, based on continuous measurements of the network performance.
- Jacobson's algorithm (1988) – RTT variance estimation:
 - ✓ generally used by TCP
 - ✓ for each connection, determine best retransmission timer (RTO) by using an estimate of RTT which includes an estimate of the variance
 - ✓ use the *mean deviation* (not standard deviation) and *exponential averaging*

TCP Timer Management – cont.

- for each connection, TCP entity maintains a variable $SRTT$, that is the best current estimate of RTT to the destination.
- if an ACK gets back before timeout, TCP measures RTT for the ACK, and updates $SRTT$:

$$SRTT(k+1) = (1-\alpha) \times SRTT(k) + \alpha \times RTT(k+1)$$

where α is a smoothing factor that determines how much weight is given to the old value. Typically $\alpha = 1/8$

- even with good value of $SRTT$, selecting the retransmission timeout is still difficult

- Normally, TCP uses $\beta \times SRTT$
 - Trick is selecting β
 - it was initially constant value $\beta = 2$ (inflexible)
- Jacobson proposed making β roughly proportional to the *mean deviation* (not standard deviation) of the acknowledgment arrival time probability density function
- keeping track of another smooth variable, D

$$D = \alpha \times D + (1-\alpha) |SRTT - RTT|$$

where α may or may not be the same value used to smooth $SRTT$

- most TCP implementations now uses Jacobson's algorithm to set the timeout interval.

$$RTO = SRTT + 4 \times D$$

Karn's Algorithm



What happen during timeout retransmission

Which RTT samples should be used as input to Jacobson's algorithm

- **Karn's proposed a simple fix to Jacobson algorithm**

- Do not update **SRTT** on any segments that have been retransmitted
- Timeout is doubled on each failure until the segments get through the first time



Exponential Backoff

- What retransmission timer (RTO) value should be used on a retransmitted segment?

TCP source increases its RTO value each time the same segment is retransmitted - backoff process

- After the first retransmission of a segment
 - wait for a longer time before performing a second retransmission

$$\text{RTO} = q \times \text{RTO}$$

- RTO grows exponentially after each backoff (common value of q is 2)



TCP Implementation Policies

- Send Policy
 - Silly Window Syndrome
- Deliver Policy
- Accept Policy
- Retransmit Policy
- Acknowledge Policy

TCP: Implementation Policies

- The standard provide a precise specification of the protocol to be used between TCP entities
- Policies defined for the protocol
 1. **Send policy**
 2. **Deliver policy**
 3. **Accept policy**
 4. **Retransmit policy**
 5. **Acknowledge policy**

1. Send Policy

In the **absence of PUSH** data and a closed transmission window, a sending TCP entity is free to **transmit data at its own convenience**

- depend on performance considerations

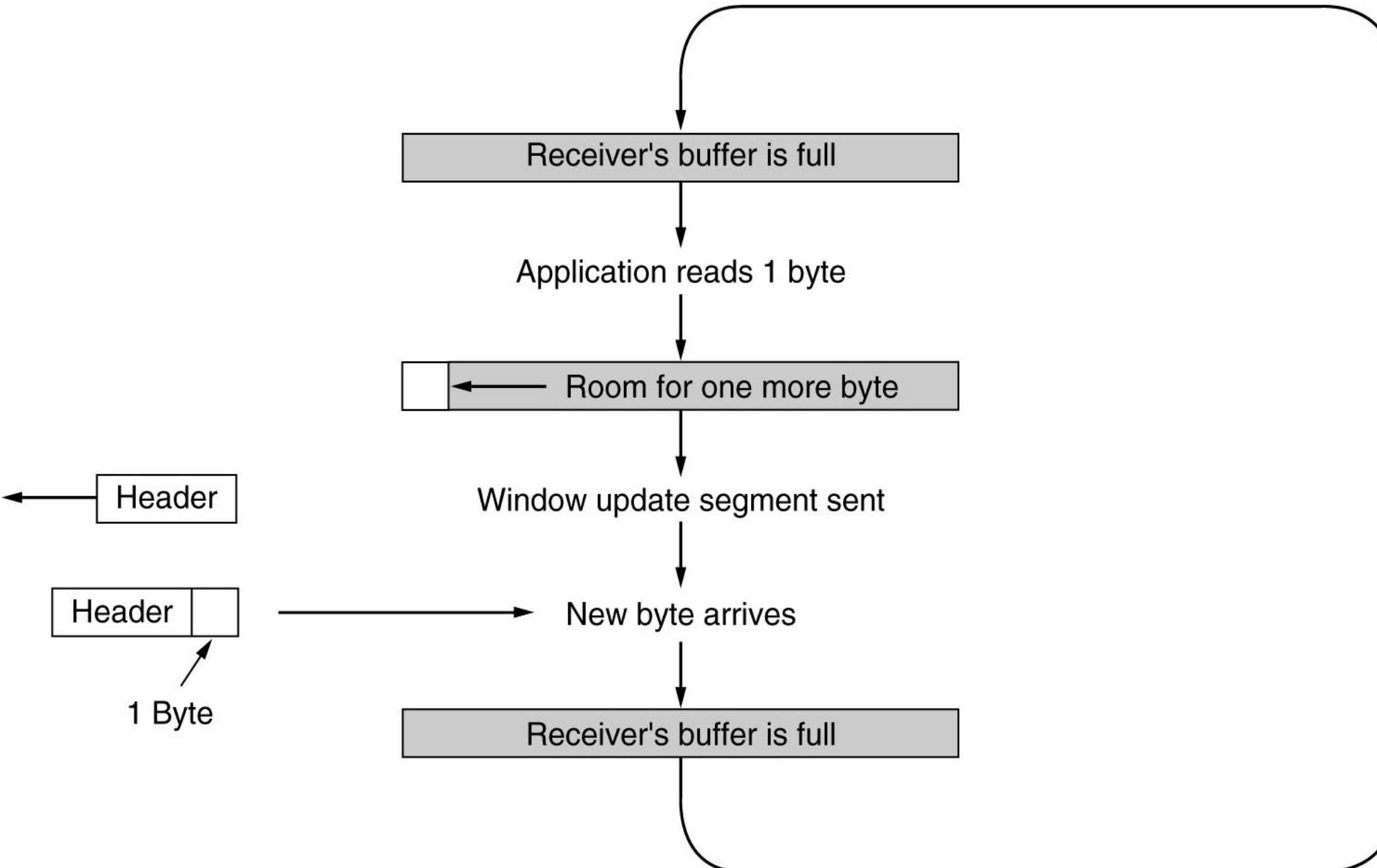
✓ if **infrequent and large** (*buffer @ sender*)

- low overheads
- slow response

✓ if **frequent and small** (*buffer @ receiver*)

- quick response
- high overheads
- silly window syndrome

Silly Window Syndrome



2. Deliver Policy

Too frequent delivery means too many OS interrupts

- Arriving data are stored in deliver buffer

- ✓ **if PUSH flag is set**

Data along with any other data in the deliver buffer are submitted to the destination application in a *RECEIVE* command.

- ✓ **if PUSH flag is not set**

TCP may wait, e.g. to *avoid excess interrupts*

- ✓ **if URG flag is set**

The receiving application is signaled that urgent data is present

3. Accept Policy

- In Order

- ✓ **Discard** out of order segments

- In Window

- ✓ **Accept** all segments within the receive window
 - ✓ Complex acceptance test and sophisticated storage



4. Retransmit Policy

TCP maintains a **queue of segments** that have been sent but **not ACKed**

- **First Only** suited for **in-window accept policy**

- ✓ one retransmission timer for the entire queue
- ✓ if an ACK is received, the segment/s removed and timer reset
- ✓ if timer expires, first segment in the queue is retransmitted

Selective-
Repeat ARQ

- **Batch** suited for **in-order accept policy**

- ✓ same as above, except when timer expires, retransmit all segments in the queue

Go-Back-N
ARQ

- **Individual** suited for **in-window accept policy**

- ✓ one timer for each segment

Selective-
Repeat ARQ

5. Acknowledge Policy

- **Immediate**
- **Cumulative**

- ✓ wait for an outbound segment, piggyback the ACK
- ✓ Timer to avoid long delay



User Datagram Protocol (UDP)

- Fundamentals
- Applications
 - DNS
 - RPC

Protocols: UDP [RFC 768]

- **TCP: Reliable ordered delivery** of packets

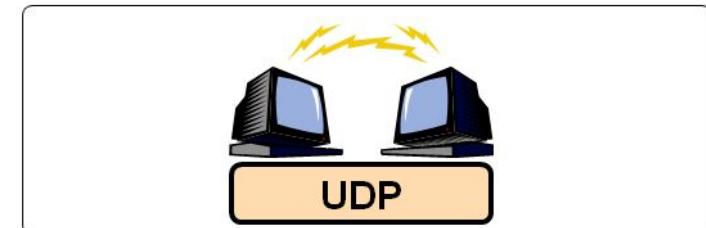
- Error detection, retransmissions and acknowledgements.
- TCP is strictly used for **point to point**
- TCP segments the data before sending to Network layer – Stream oriented

- **UDP: Unreliable delivery** of packets

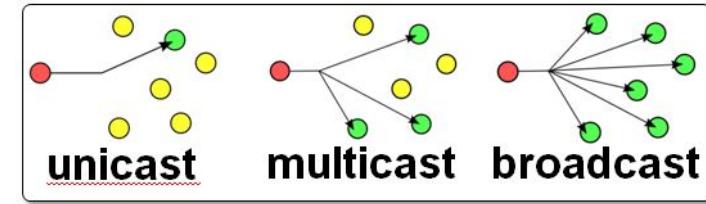
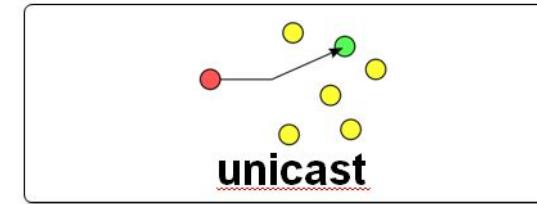
- Connectionless protocol
- No retransmissions, acknowledgements
- UDP does not segment the data – message oriented



- Slower but reliable transfers
- Typical applications:
 - Email
 - Web browsing



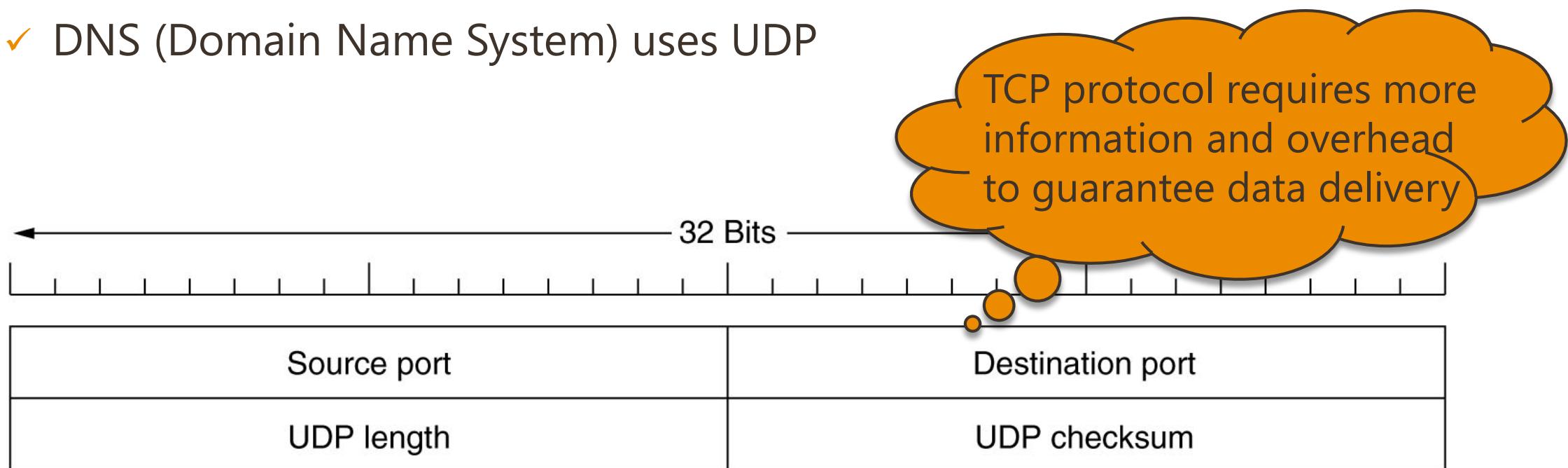
- Fast but non-guaranteed transfers (“best effort”)
- Typical applications:
 - VoIP
 - Music streaming



No flow control, error control, or retransmission upon receipt of bad segments

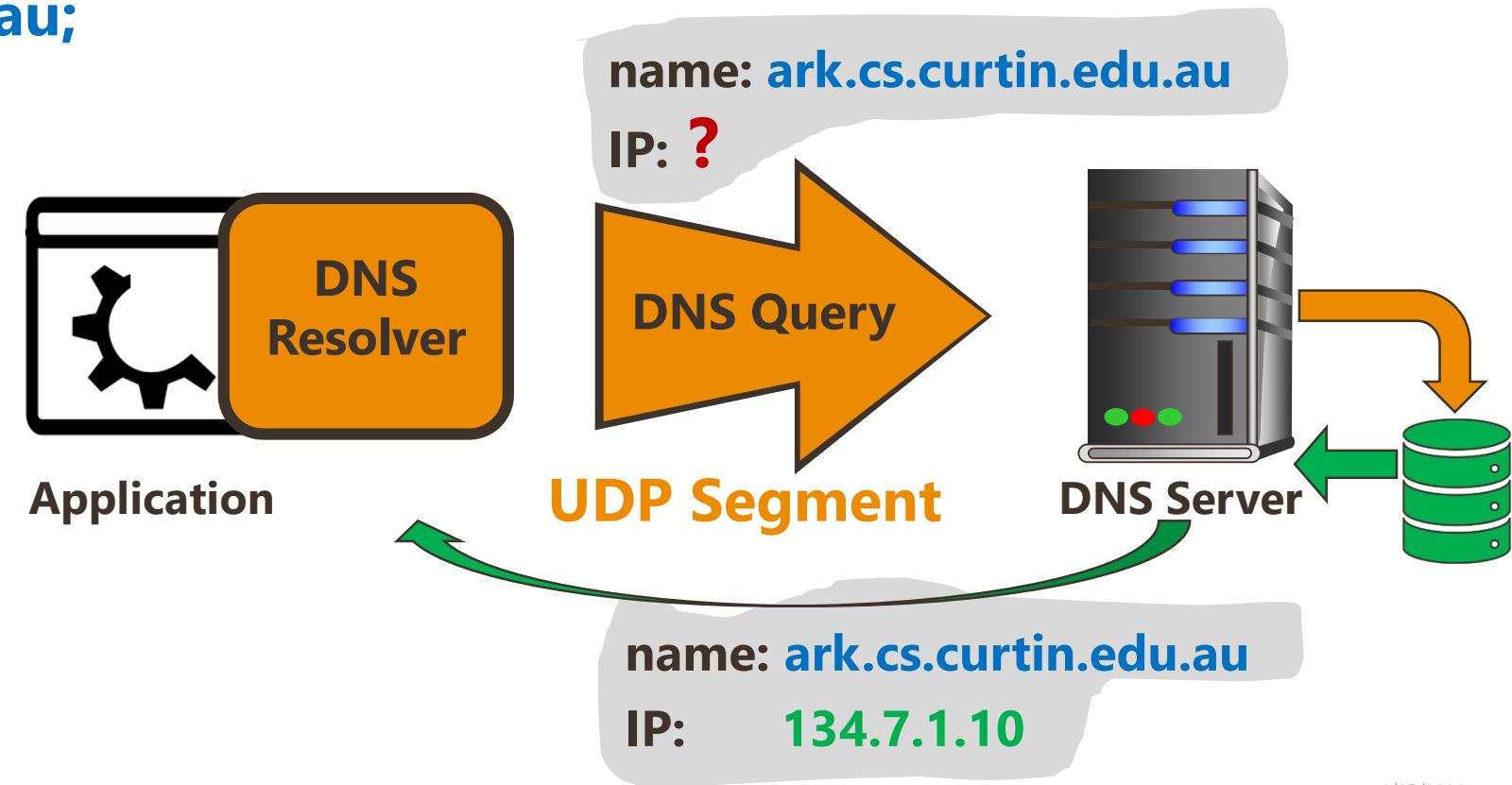
Why UDP?

- Lower overhead enables faster transmissions
- **Unicast, Multicast, Broadcast**
- UDP is especially useful in client-server situation
 - ✓ DNS (Domain Name System) uses UDP



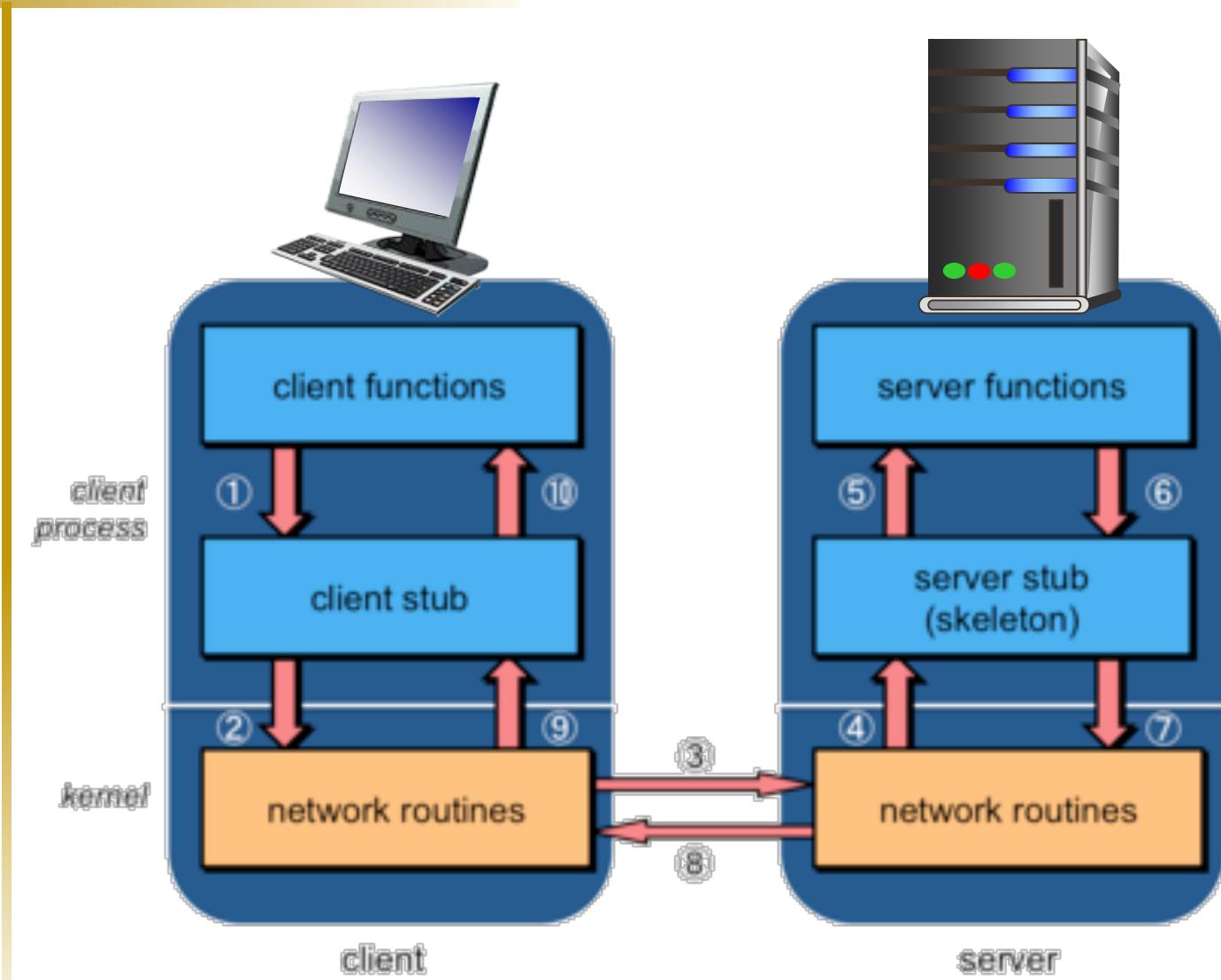
UDP: DNS (Domain Name System)

- How can the **IP address of the remote host** be found?
- Hierarchical, symbolic addresses are used
 - e.g. **ark.cs.curtin.edu.au**;
 - **curtin.edu.au**



Remote Procedure Call (RPC) - 1984

- Allows program to call procedures located on remote hosts
- Information is transported from the caller to the callee (remote host) in the parameters and can come back in the procedure result
- **Message passing is invisible** to the programmer



RPC – cont.

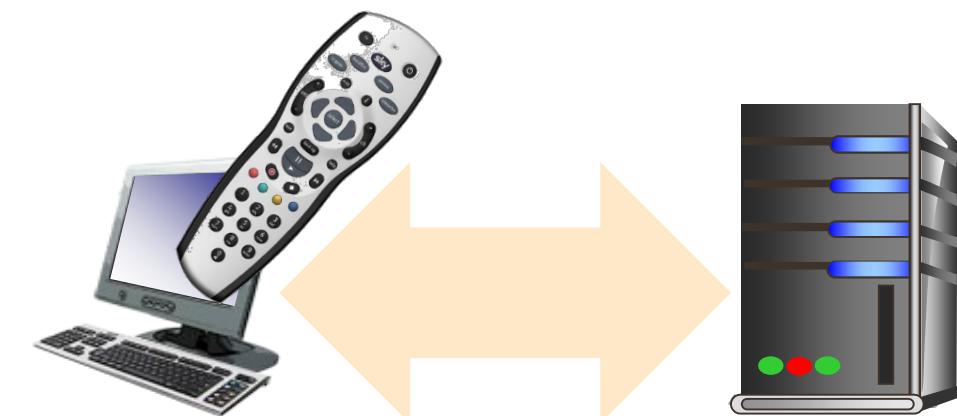
- **Client Stub**

- **a small library procedure** that represents the server procedure in the client's address space that the Client program must be bound with.

- **Server Stub**

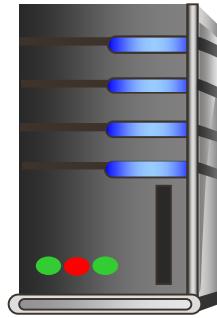
- **a procedure call that represents the client procedure call** in the server's address space that the Server program must be bound with.

- **UDP is commonly used for RPC**

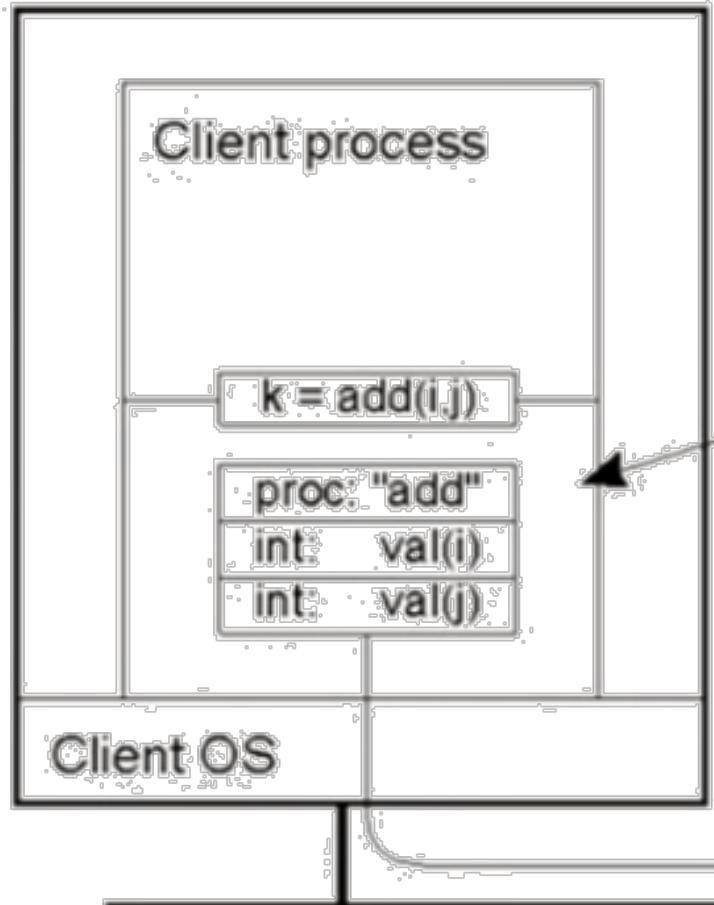




Client machine



Server machine



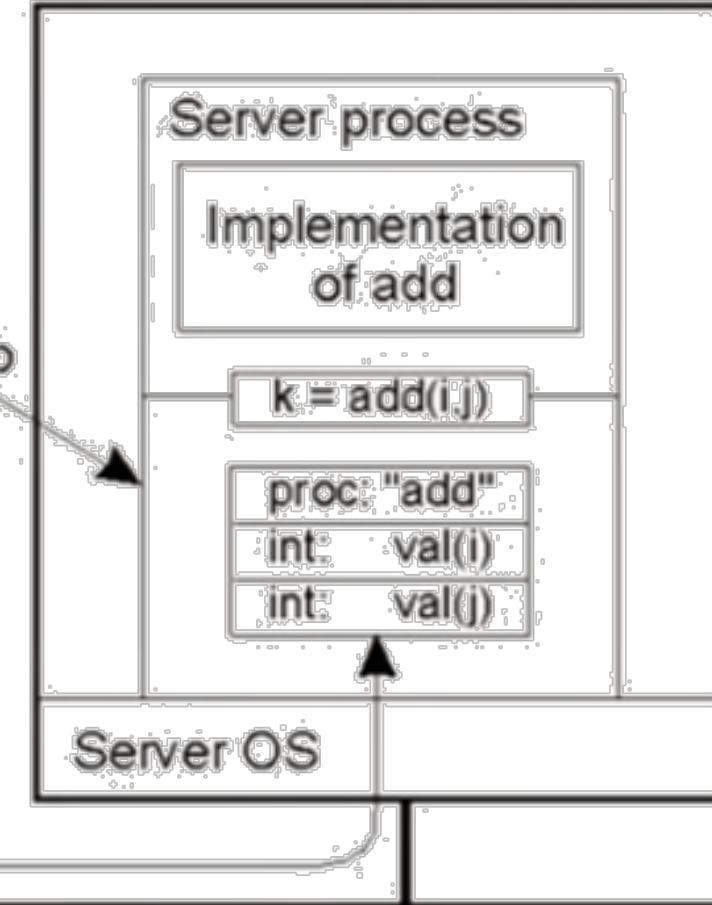
1. Client call to procedure

Server stub
Client stub

2. Stub builds message



3. Message is sent across the network



6. Stub makes local call to "add"

5. Stub unpacks message

4. Server OS hands message to server stub



▪ **Flow Control and Congestion Control**

- Flow Control
 - Transport Layer
 - Data Link Layer
- Congestion Control
 - Warning-bit
 - Choke-packets
 - Load-shedding
 - RED

▪ **TCP**

- TCP Header
 - Flags (SYN, FIN, ACK, RST)
 - Flag (URG, PSH) – *in depth*
 - TCP Options
 - Window Size (Dynamic Buffer Management)
- TCP Flow Control
 - Dynamic Buffer Management
- TCP Congestion Control
 - TCP Segment Pacing Effect
 - Slow Start
 - Dynamic Window Sizing
- TCP Timer Management

▪ **TCP - Implementation Policies**

- Send Policy
 - Silly Window Syndrome
- Deliver Policy
- Accept Policy
- Retransmit Policy
- Acknowledge Policy

▪ **UDP**

- Fundamentals
- Applications
 - DNS
 - RPC

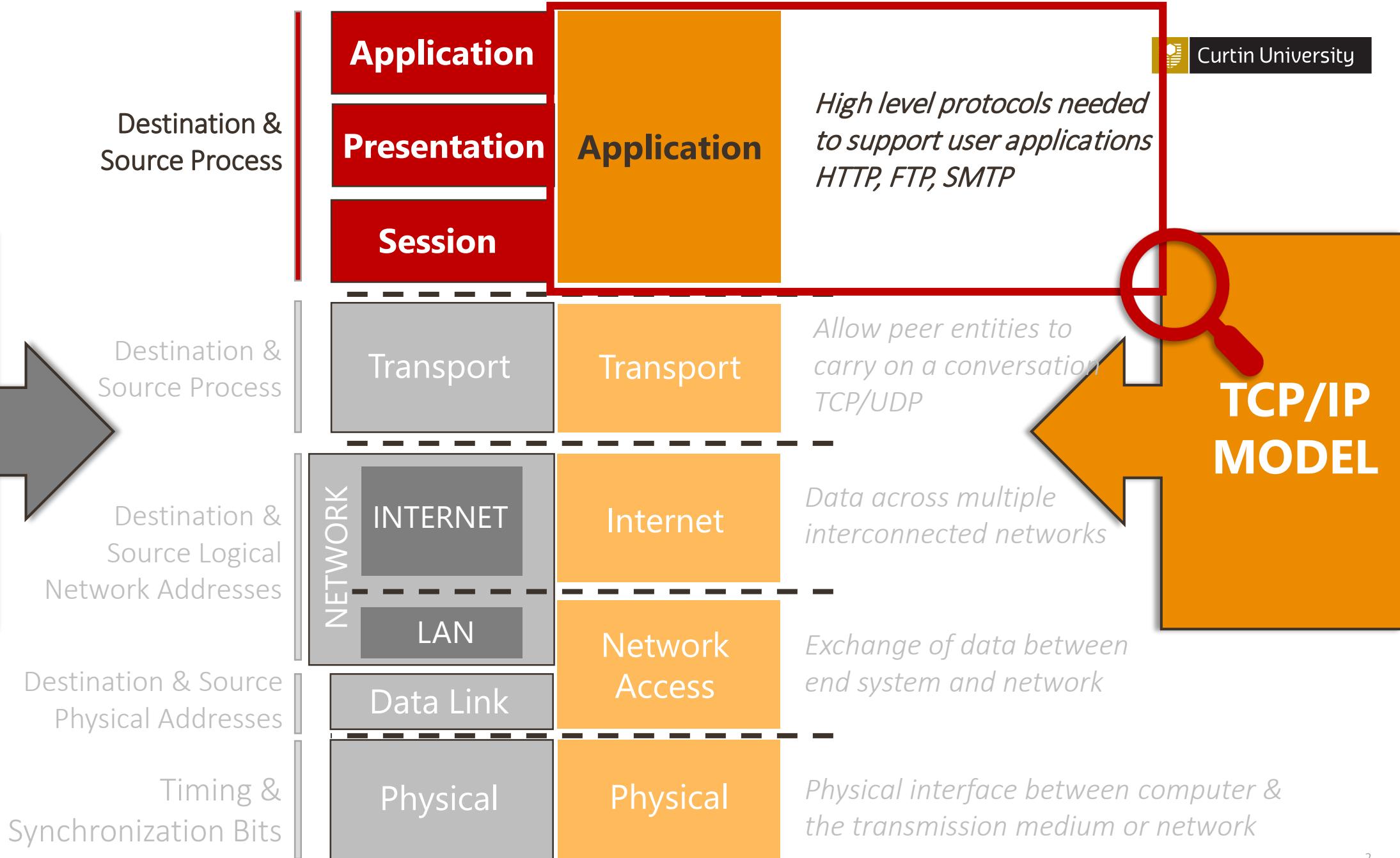
THANK YOU

Make tomorrow better.

Application Layer I

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 09

Semester 1, 2021



Application Layer



Defines communications protocols and interface methods used in process-to-process communications

Overview

- **Network Application Architectures**

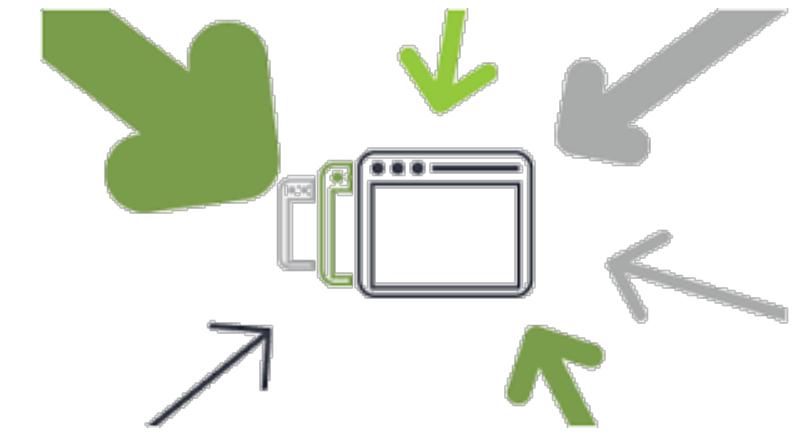
- Client-Server
- Peer-to-peer

- **Application Layer Protocols**

- Telnet
- FTP
- HTTP
- SMTP/POP3/IMAP
- DHCP

Application Layer

- All the communication applications / processes
- **Layers below** are there **to provide reliable transport**
- What we appreciate is how these applications are build on **top of the lower layers**
- Application layer in TCP/IP mainly cover the Session, Presentation, and Application Layers of the OSI Reference Model



Some Applications

- E-mail



- Web



- Remote login



TeamViewer

- P2P file sharing



BitTorrent™

- Streaming stored video



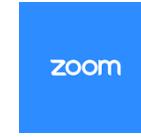
NETFLIX



- Voice over IP
e.g. Skype



- Real-time video conferencing



- Social networking



- Search



- Multi-user network games



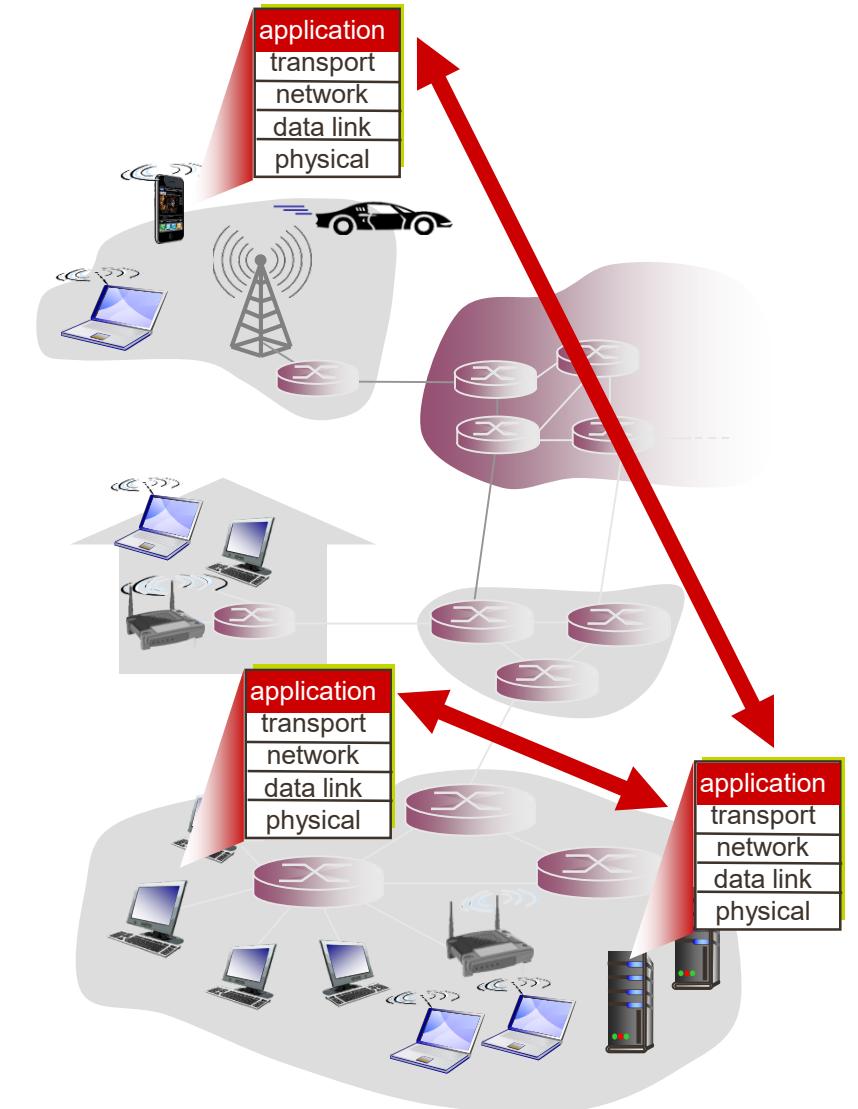
Creating a Network Application

- **Write programs** that:

- ✓ run **on** (different) **end systems**
- ✓ communicate over network
- ✓ i.e. web server software <-> browser software

- **No need to write software for network-core devices**

- network-core devices do not run user applications
- applications on end systems allows for rapid app development, propagation



App-layer Protocol Defines

- **Message Type** exchanged

- e.g., request, response

- **Message Syntax**

- what fields in messages & how fields are delineated

- **Message Semantics**

- meaning of information in fields

- **Rules** for **when and how** processes **send & respond** to messages



Open Protocols:

- defined in RFCs
- allows for interoperability
- e.g., **HTTP, SMTP**



Proprietary Protocols:

- i.e. **Skype**



Transport Services Required

- **Data integrity**

- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer
- other apps (e.g., audio) can tolerate some loss

- **Timing**

- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

- **Throughput**

- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

- **Security**

- encryption, data integrity, ...

Transport Services vs App

Application	Data Loss	Throughput	Time Sensitive
File Transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web Documents	No loss	Elastic	No
Real-time Audio/Video	Loss-tolerant	Audio: 5kbps- 1Mbps Video: 10kbps- 5Mbps	Yes, 100's msec
Stored Audio/Video	Loss-tolerant	Same as above	Yes, few secs
Interactive Games	Loss-tolerant	Few kbps	Yes, 100's
Text Messaging	No loss	Elastic	msec (yes and no)

Apps **underlying** TCP Protocols

application	application layer protocol	underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	TCP or UDP



traceroute

ping



telnet

FTP

SMTP

HTTP

DNS

TFTP

TCP

Transport

UDP

ICMP

Network

IGMP

ARP

Data Link

RARP

Secure Communication

- **TCP & UDP**

- no encryption
- cleartext passwords sent into port (socket) traverse Internet in clear text



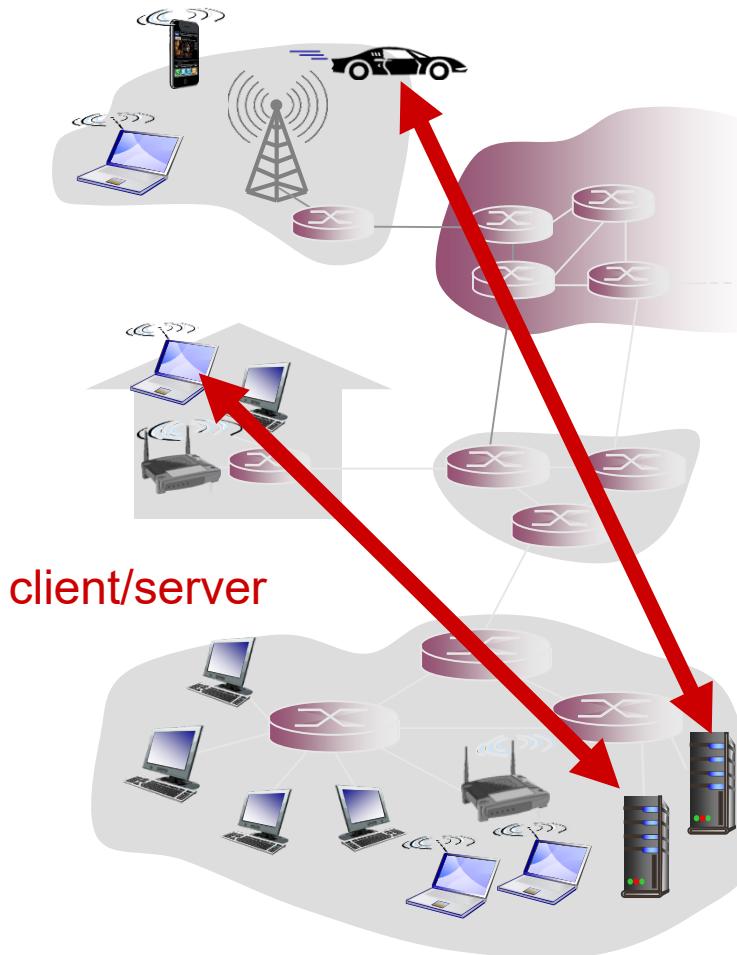
- **SSL**

- provides encrypted TCP connection
- data integrity
- end-point authentication

- **SSL is at app layer**

- Apps use SSL libraries, which “talk” to TCP

Client-Server Architecture



■ Server:

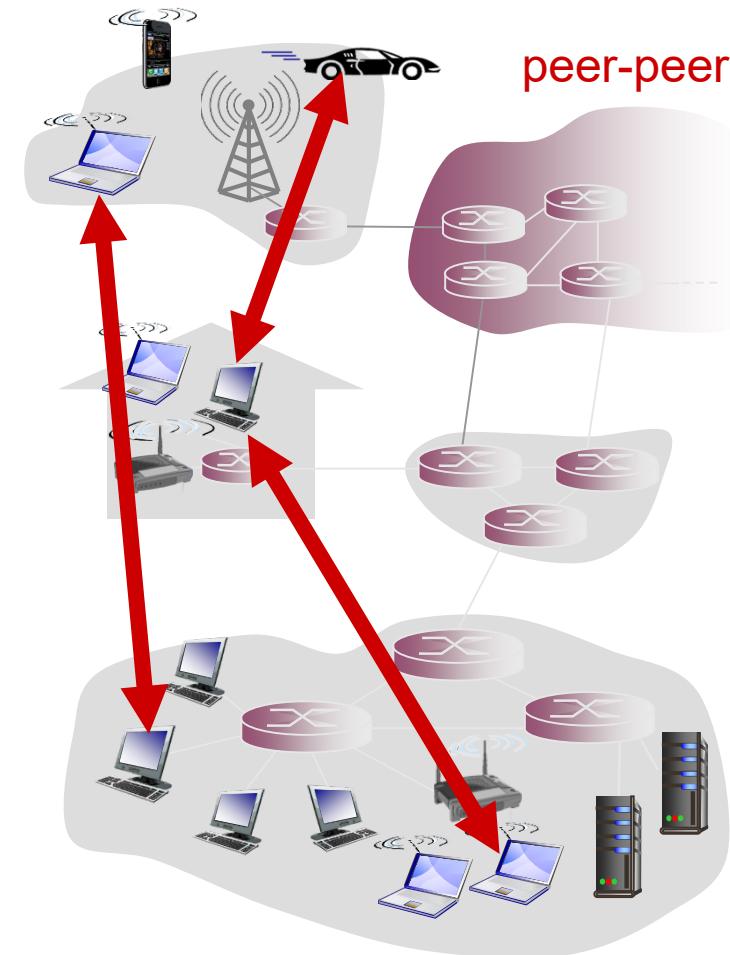
- ✓ always-on host
- ✓ permanent IP address
- ✓ data centers for scaling

■ Clients:

- ✓ communicate with server
- ✓ may be intermittently connected
- ✓ may have dynamic IP addresses
- ✓ do not communicate directly with each other

P2P Architecture

- **No always-on server**
- **Arbitrary end systems** directly communicate
- Peers request service from other peers, provide service in return to other peers
 - ✓ ***self scalability*** – new peers bring new service capacity, as well as new service demands
- Peers are intermittently connected and change IP addresses
 - ✓ ***complex management***





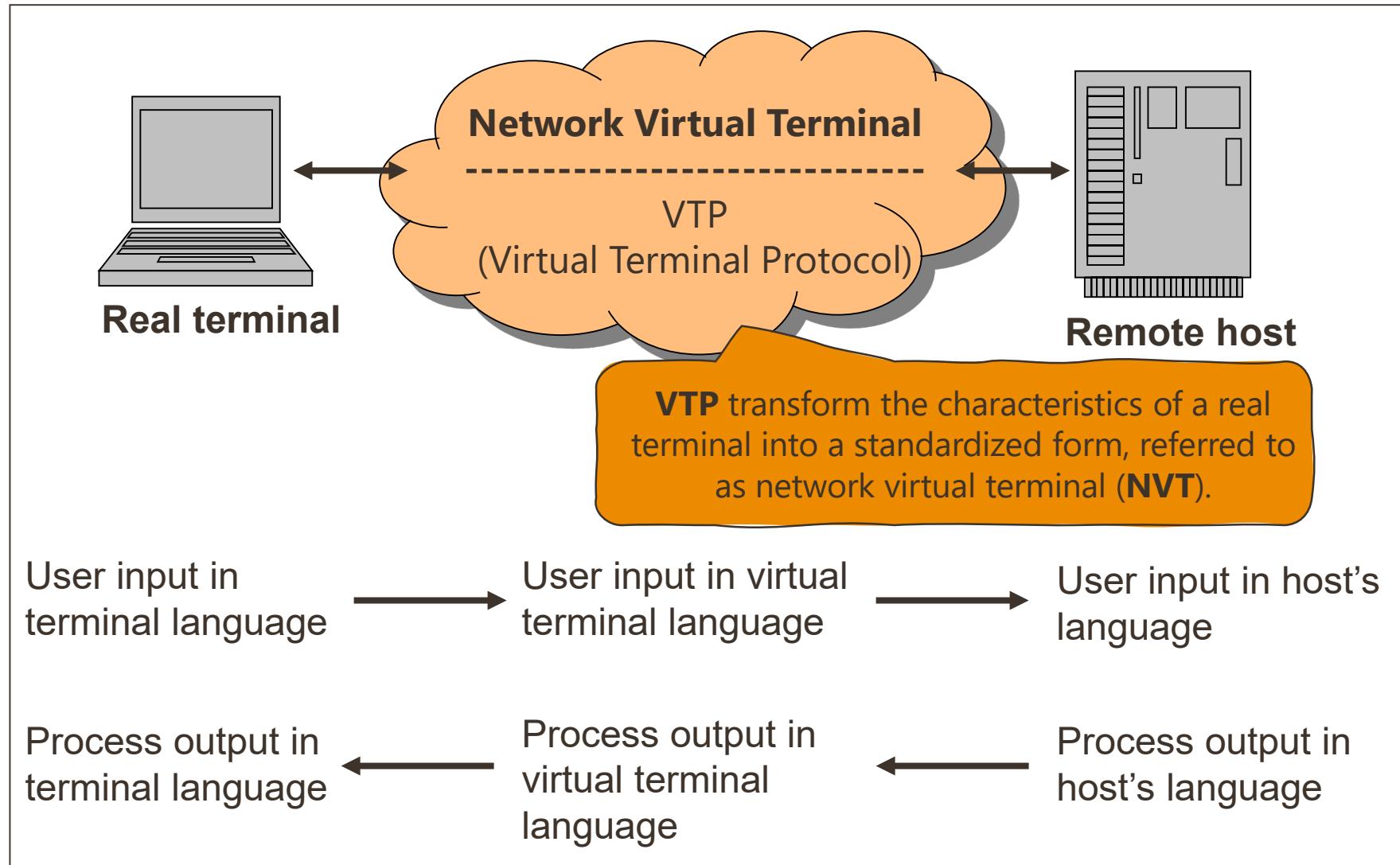
Application Layer Protocols - **Telnet**

- Fundamentals
- NVT, VTP
- Connections
- Highlights

Terminal Access – Telnet

- One of the oldest application (1969)
- Basis of many newer protocols
- Telnet is a **remote logon facility** based on the use of a virtual terminal protocol (**VTP**) and a network virtual terminal (**NVT**).
- Both real terminal's characteristics and a host's representation of a terminal are mapped into a network virtual terminal for data transfer

Using TCP connection, Telnet can be used between two terminals, two processes, or a terminal and a process



Telnet Transfer Protocol

- Data sent **half-duplex**
- **Terminal-to-process:**
 - newline signifies end of user input.
- **Process-to-terminal:**
 - Telnet **Go Ahead Command** is used
(Returns the prompt to the user)
- Underlying TCP **full duplex**
 - **Control Signals** sent any time regardless of current data direction
- Data are sent as a **stream of 8-bit bytes**
 - no other formatting to the data
 - control data and other non-data information are sent as **telnet commands**:
 - **Interrupt process** (IP) – code 244
 - **Break** (BRK) – code 243
 - **Interpret as Command** (IAC) – code 255

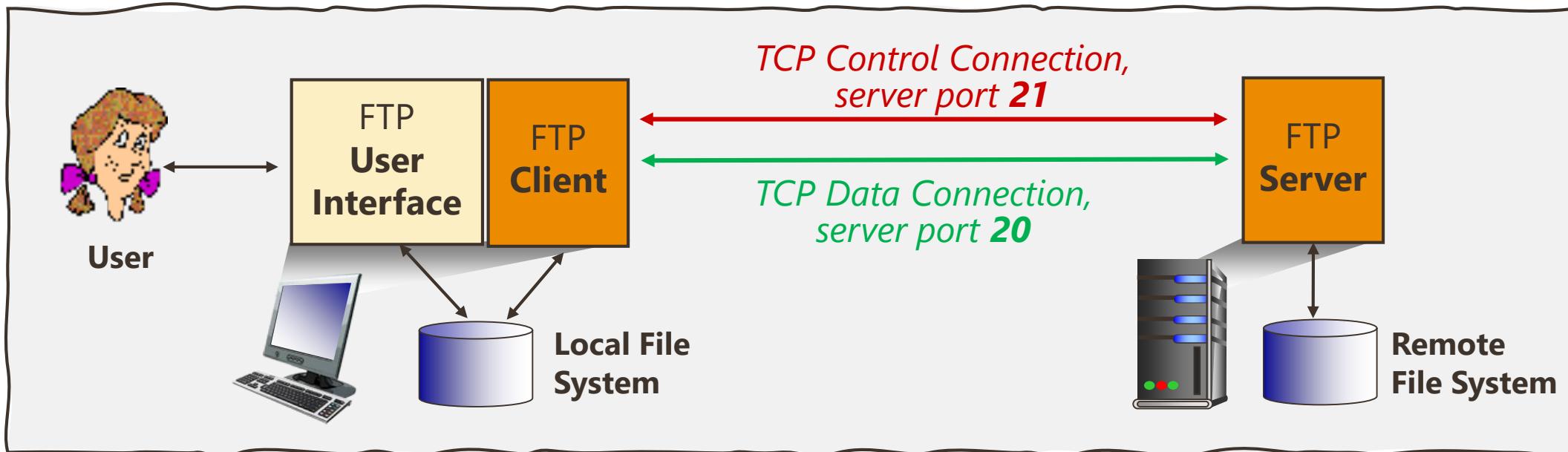


Application Layer Protocols - **FTP**

- Fundamentals
- Connections
- Commands and Responses

File Transfer Protocol (FTP)

- Client **initiates** connection
- Client **authorized over control connection**
- Client browses remote directory, sends commands over control connection
- When server receives file transfer command, server opens **2nd TCP data connection** (for file transfer) to client



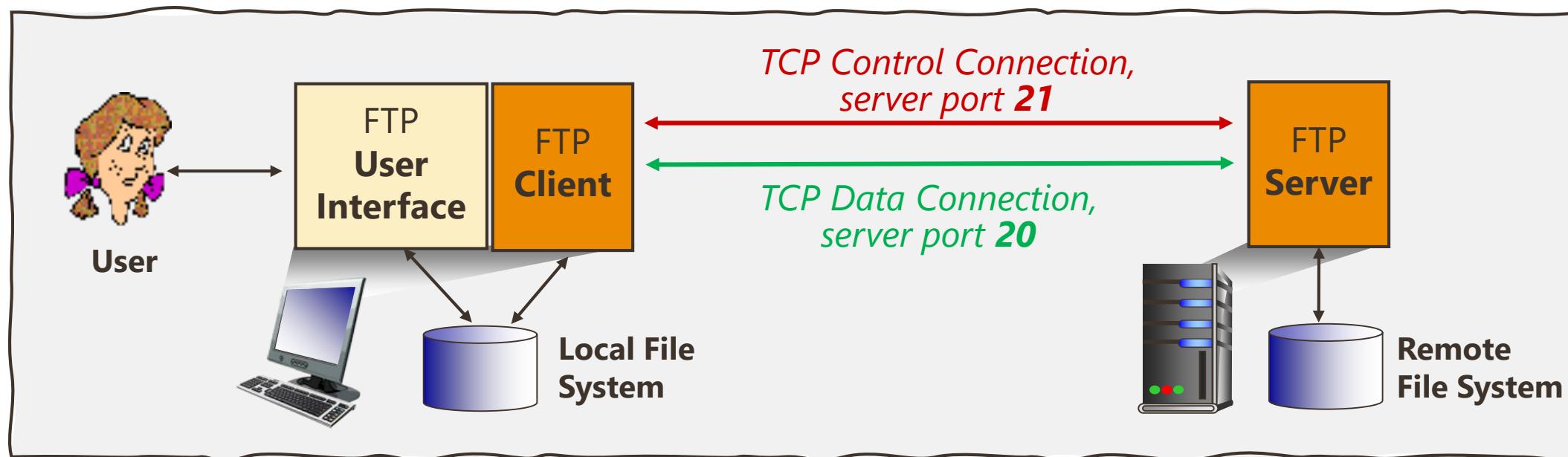
Separate control, data connections

- **Control Connection: “out of band”**

1 control connection, many data connections (i.e. each file transfer)

- **FTP server maintains “state”**

- current directory, earlier authentication etc.



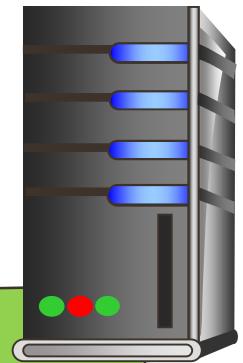
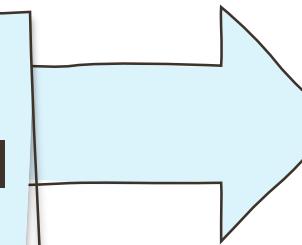
FTP Commands, Responses

Sample Commands

- Sent as **ASCII text** over control channel

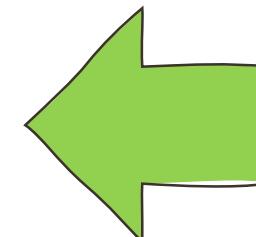
- ✓ **USER** username
- ✓ **PASS** password
- ✓ **LIST** return list of file in current directory
- ✓ **RETR** filename retrieves (gets) file

STOR filename stores (puts) file onto remote host



Sample Return Codes

- **status code** and phrase (as in HTTP)
 - ✓ **331** Username OK, password required
 - ✓ **125** data connection already open; transfer starting
 - ✓ **425** Can't open data connection
 - ✓ **452** Error writing file





Application Layer Protocols - **HTTP**

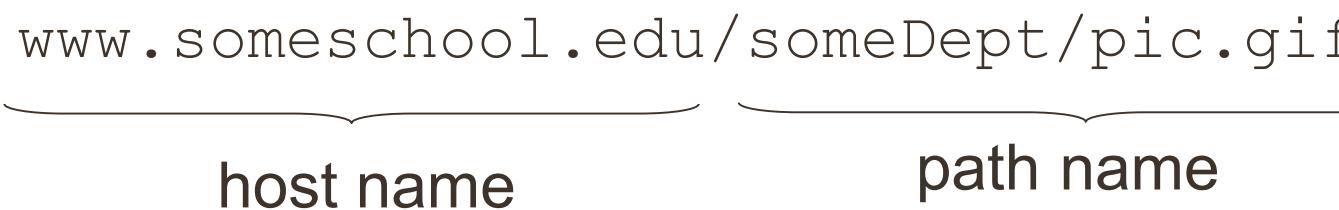
- Review on web
 - Web resource
 - WWW
- HTTP
 - Connections (persistent/non-persistent)
 - Messages (http request / http response)
 - HTTP 1.1 / HTTP 2.0
 - Maintaining State (Cookies)
 - Web Caching (Browser Cache, Proxy Server)
 - Conditional GET
 - Web Sockets

Web and HTTP

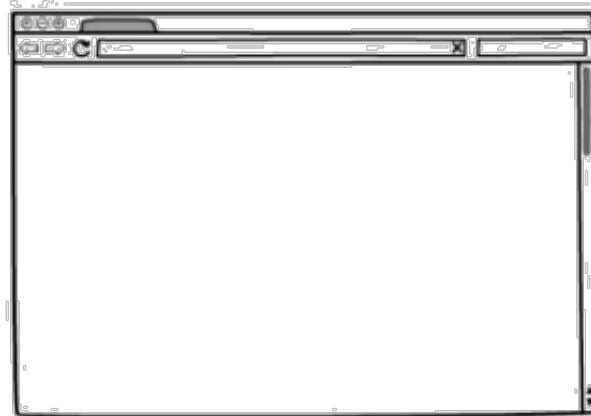
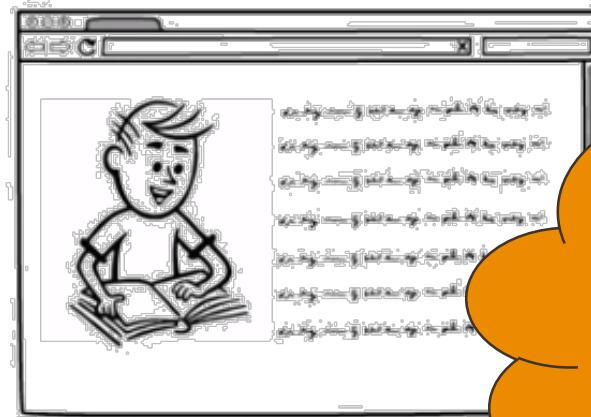
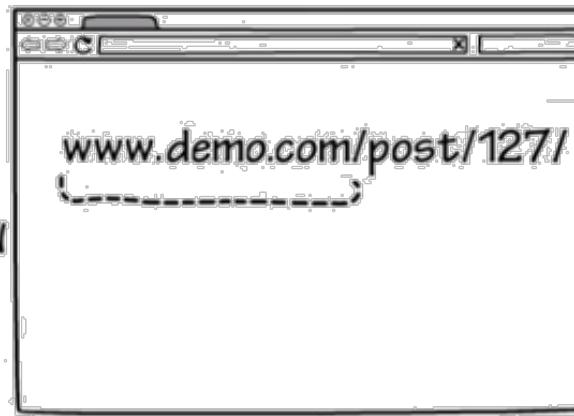
▪ First, a review...

- Web page consists of objects
- Object can be HTML file, JPEG image, Java applet, audio file,...
- Web page consists of base HTML-file which includes several referenced objects
- Each object is addressable by a URL, e.g.

www.someschool.edu/someDept/pic.gif



The URL is shown above two horizontal curly braces. The first brace spans from the start of the URL to the first slash, labeled "host name". The second brace spans from the first slash to the end of the URL, labeled "path name".

WWW**Website 1**www.test.com/post/1241**Website 2**www.demo.com/post/1261

All publicly
accessible hyper-
linked webpages

**WORLD
WIDE
WEB****Information
Space**

Web and HTTP – cont.

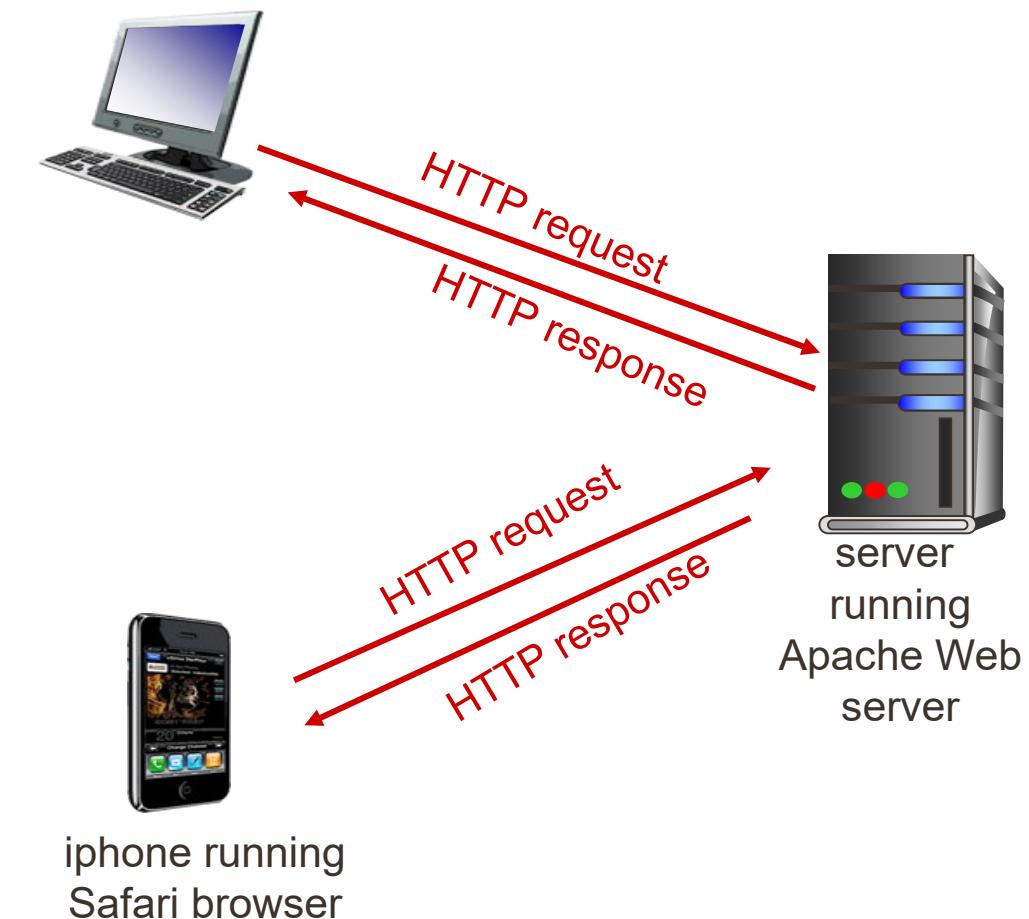
Is the Internet and the World Wide Web (WWW) the same



- Internet is a network of networks
- To store information and share it | i.e. website
- WWW is a ***distributed system*** that runs on top of the Internet
 - Not a network!

HTTP Overview

- **HTTP:** hypertext transfer protocol
- Web's application layer protocol
- **Client/Server model**
 - client: browser that requests, receives, (using HTTP protocol) and “displays” Web objects
 - server: Web server sends (using HTTP protocol) objects in response to requests



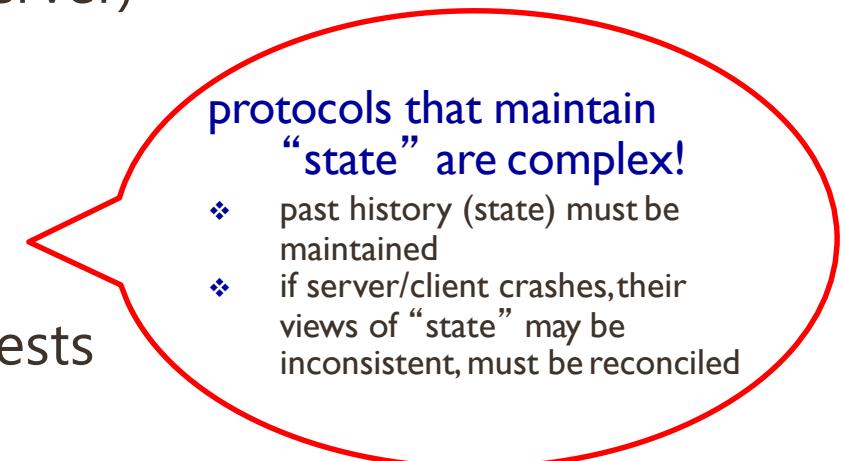
HTTP Overview – cont.

- **Uses TCP:**

- Client initiates TCP connection (creates socket) to server, port 80
- Server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

- **HTTP is “stateless”**

- server maintains no information about past client requests



protocols that maintain
“state” are complex!

- ❖ past history (state) must be maintained
- ❖ if server/client crashes, their views of “state” may be inconsistent, must be reconciled

HTTP Connections

▪ Non-persistent HTTP

- at most one object sent over TCP connection
- connection then closed
- downloading multiple objects required multiple connections

▪ Persistent HTTP

- multiple objects can be sent over single TCP connection between client and server



Non-persistent HTTP

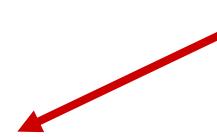
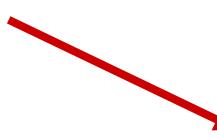
Suppose user enters URL:

www.someSchool.edu/someDepartment/home.index

contains text,
references to 10
jpeg images

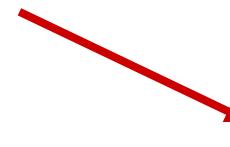
1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. “accepts” connection, notifying client



Non-persistent HTTP – cont.

2. HTTP client sends HTTP *request* message (containing URL) into TCP connection socket.
Message indicates that client wants object
`someDepartment/home.index`



3. HTTP server receives request message, forms *response* message containing requested object, and sends message into its socket
4. HTTP server closes TCP connection.



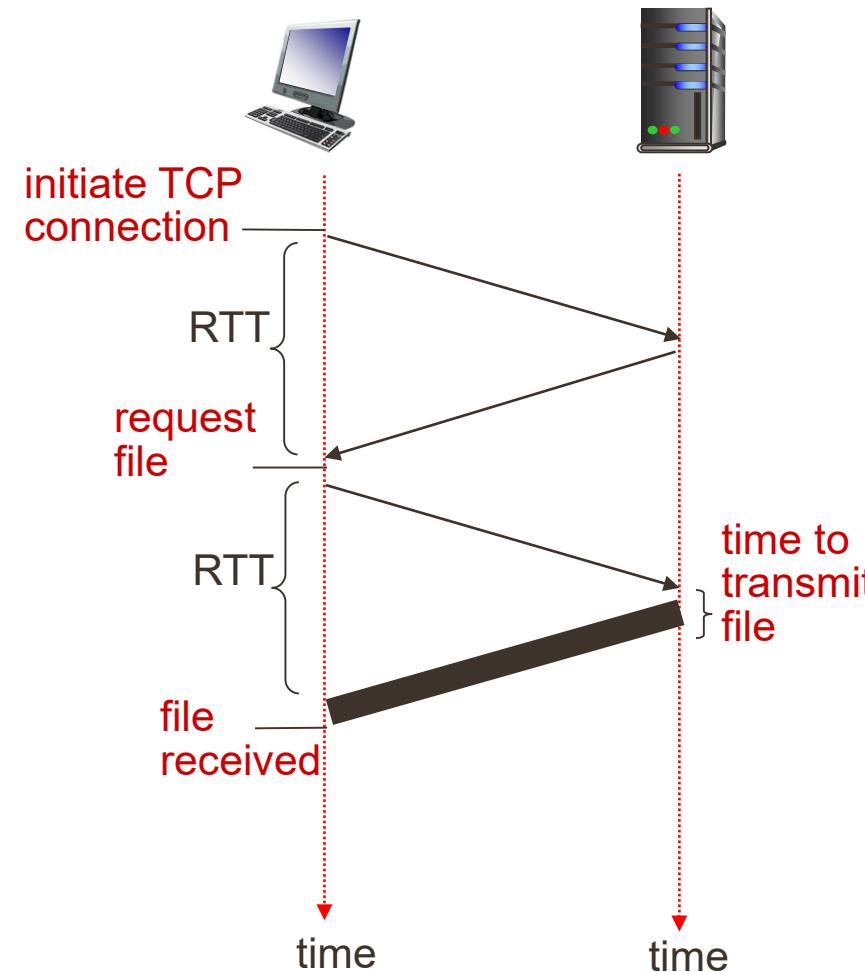
Non-persistent HTTP – cont.

- 
5. HTTP client receives *response* message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

 6. Repeat Steps 1-5 for each of the 10 jpeg objects

Non-persistent HTTP: response time

- HTTP response time:
 - one RTT to initiate TCP connection
 - one RTT for HTTP request and first few bytes of HTTP response to return
 - file transmission time
 - non-persistent HTTP response time = $2\text{RTT} + \text{file transmission time}$



Persistent HTTP

- **Non-persistent HTTP issues:**

- requires 2 RTTs per object
- OS overhead for each TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

- **Persistent HTTP:**

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

HTTP Messages

HTTP Request:

- ASCII (human-readable format)

✓ Methods

- GET, POST
- HEAD,
- PUT, DELETE

✓ URL

- Requested resource path (on server)

✓ Version

- HTTP/1.0
- HTTP/1.1
- HTTP/2.0



HTTP Response:

✓ Status code (*similar in FTP*)

200 OK

404 Not Found

400 Bad Request

301 Moved Permanently





HTTP Request

Req,
Method

URL

Version

GET

/index.html

HTTP/1.1

\r

\n

**Request
Line**

Name: Value
Pairs

Host:	www-net.cs.umass.edu	\r	\n
User-Agent:	Firefox/3.6.10	\r	\n
Accept:	text/html , application/xhtml+xml	\r	\n
Accept-Language:	en-us , en ; q=0.5	\r	\n
Accept-Encoding:	gzip , deflate	\r	\n
Accept-Charset:	ISO-8859-1,utf-8;q=0.7	\r	\n
Keep-Alive:	115	\r	\n
Connection:	Keep-alive	\r	\n
\r\n			

**Header
Lines**

END of Header Lines -
Carriage return & line
feed at start of line

Entity Body

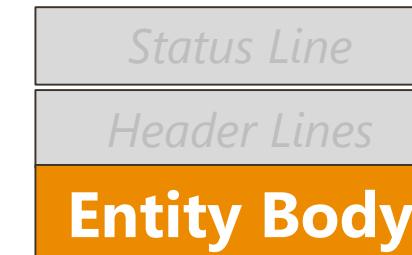
Body

Uploading Form Input

▪ GET Method

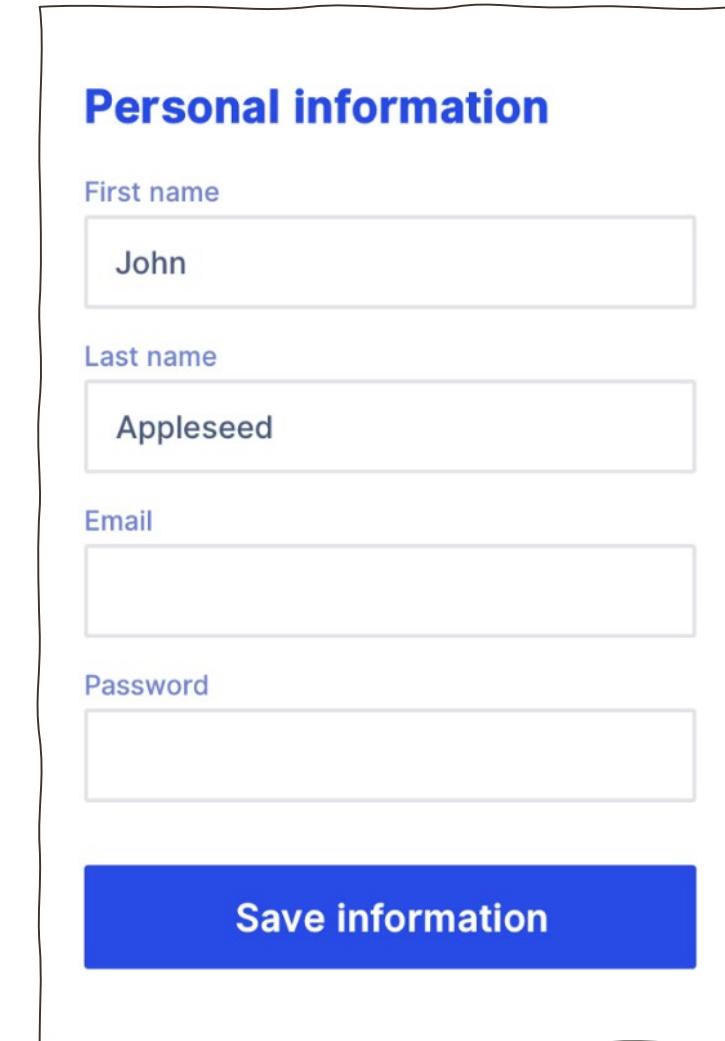
Input is uploaded in **URL field** of request line:

`www.w3schools.com/action_page.php?fname=John&lname=Appleseed`



▪ POST Method

Input is uploaded to server in the body of the request message



Personal information

First name

Last name

Email

Password

Save information

Request Methods – cont.

Status Line

Header Lines

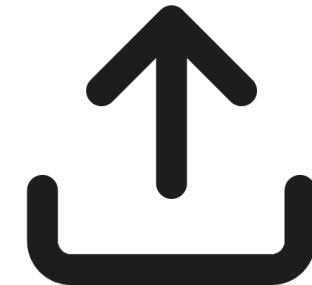
✗ Entity Body

HEAD

Asks server to leave **requested object out of response**

PUT

uploads file in entity body to path specified in **URL field**



DELETE

deletes file specified in the **URL field**



HTTP Response

Protocol

Status Code

Status Phrase

HTTP/1.1

200

OK

\r

\n

Status Line

Name: Value Pairs

Date:	Sun, 26 Sep 2010 20:09:20 GMT	\r	\n
Server:	Apache/2.0.52 (CentOS)\	\r	\n
Last-Modified:	Tue, 30 Oct 2007 17:00:02 GMT	\r	\n
ETag:	"17dc6-a5c-bf716880"\	\r	\n
Accept-Ranges:	bytes	\r	\n
Accept-Length:	2652	\r	\n
Keep-Alive:	timeout=10, max=100	\r	\n
Connection:	Keep-Alive	\r	\n
Connection-Type:	text/html; charset=ISO-8859-1		
\r\n			

Header Lines

END of Header Lines -
Carriage return & line
feed at start of line

Entity Body

Data i.e.
Requested
HTML file

HTTP Response - Status Codes

- **Some Sample Codes:**

- **200 OK**

- request succeeded, requested object later in this msg

- **301 Moved Permanently**

- requested object moved, new location specified later in this msg (Location:)

- **400 Bad Request**

- request msg not understood by server

- **404 Not Found**

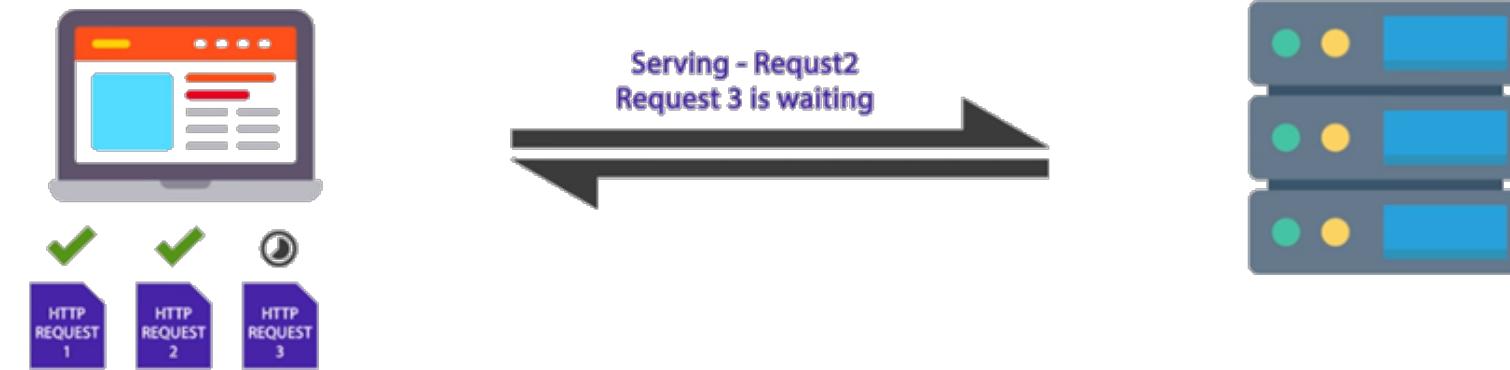
- requested document not found on this server

- **505 HTTP Version Not Supported**

HTTP 1.1 - Problems

1. Head-of-Line Blocking

The TCP connection/channel is blocked by the preceding request



2. Redundancy in request header

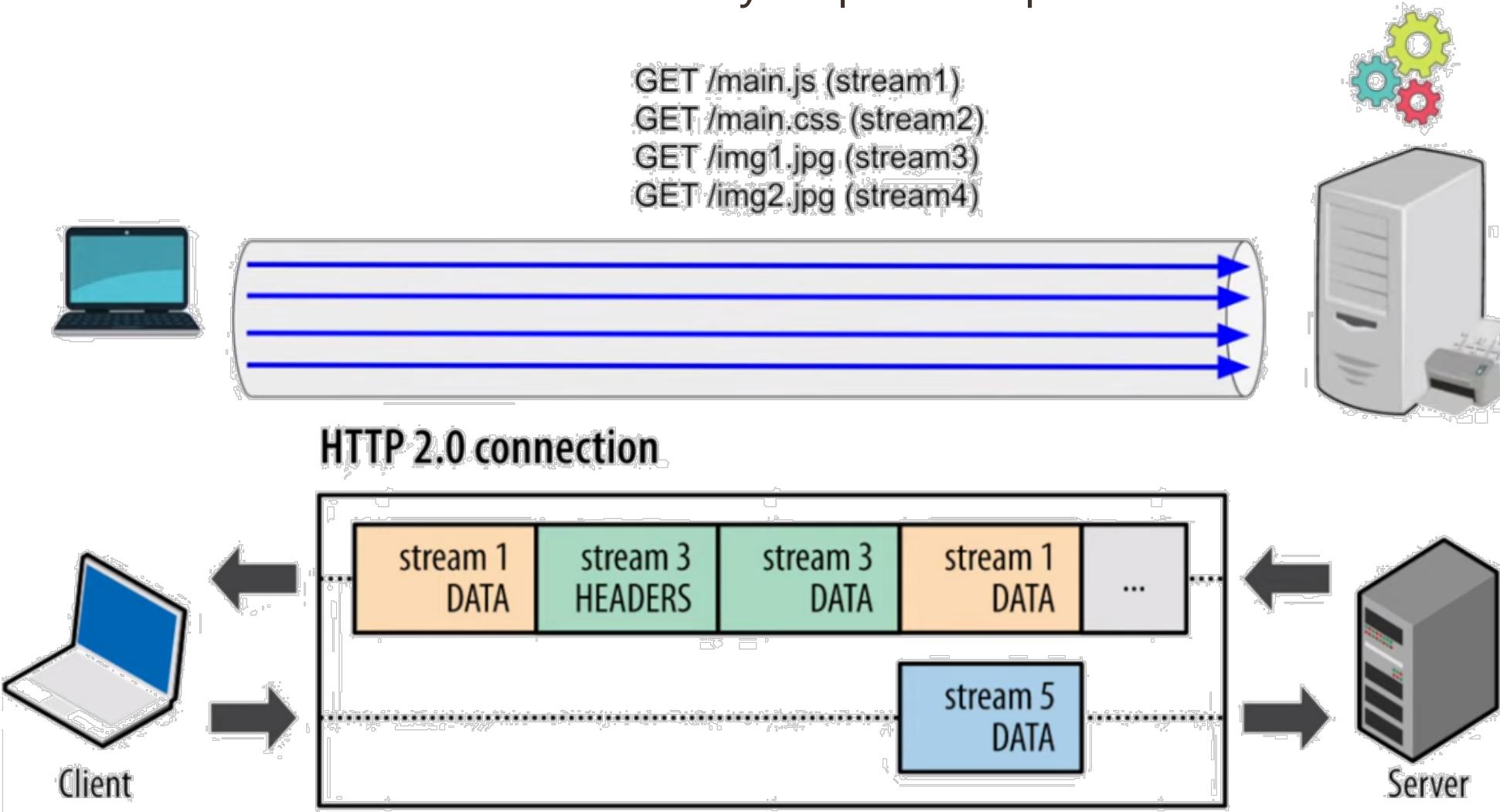
- Sending same static header parameters again and again

3. No header compression

HTTP 2.0

1. No More Head-of-Line Blocking

- Single TCP connection serve multiple requests by multiplexing
- StreamID is used to identify request-response streams

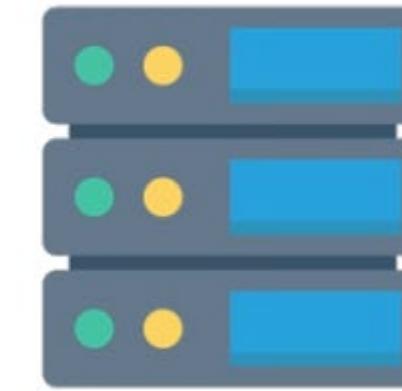


HTTP 2.0

2. Allows to compress HTTP Headers, Data



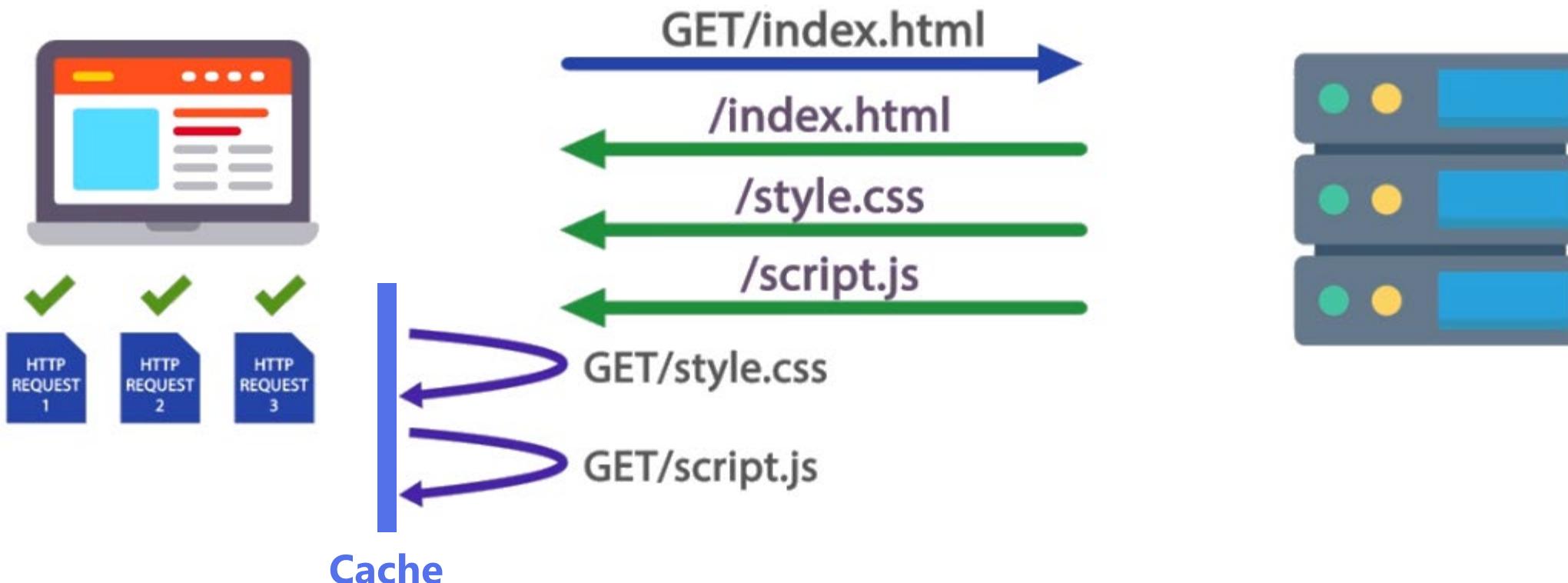
- Own compression format (HPACK)
- Compression works at connection level,
- so that headers can be shared among requests.



HTTP 2.0

3. Push (not push notifications)

- Allows respond to the request that hasn't even being sent.
- But you are sure the client would request it.
- During the actual request, it will be fetched from the **cache**.



User-server state: cookies

- Many web sites **use cookies**
- Four components:
 1. **cookie header line** of **HTTP response** message (*from server*)
 2. **cookie header line** in next **HTTP request** message (*to the server*)
 3. **cookie file** kept **on user's host**, managed by user's browser (no extra burden on the server)
 4. **back-end database** at web server

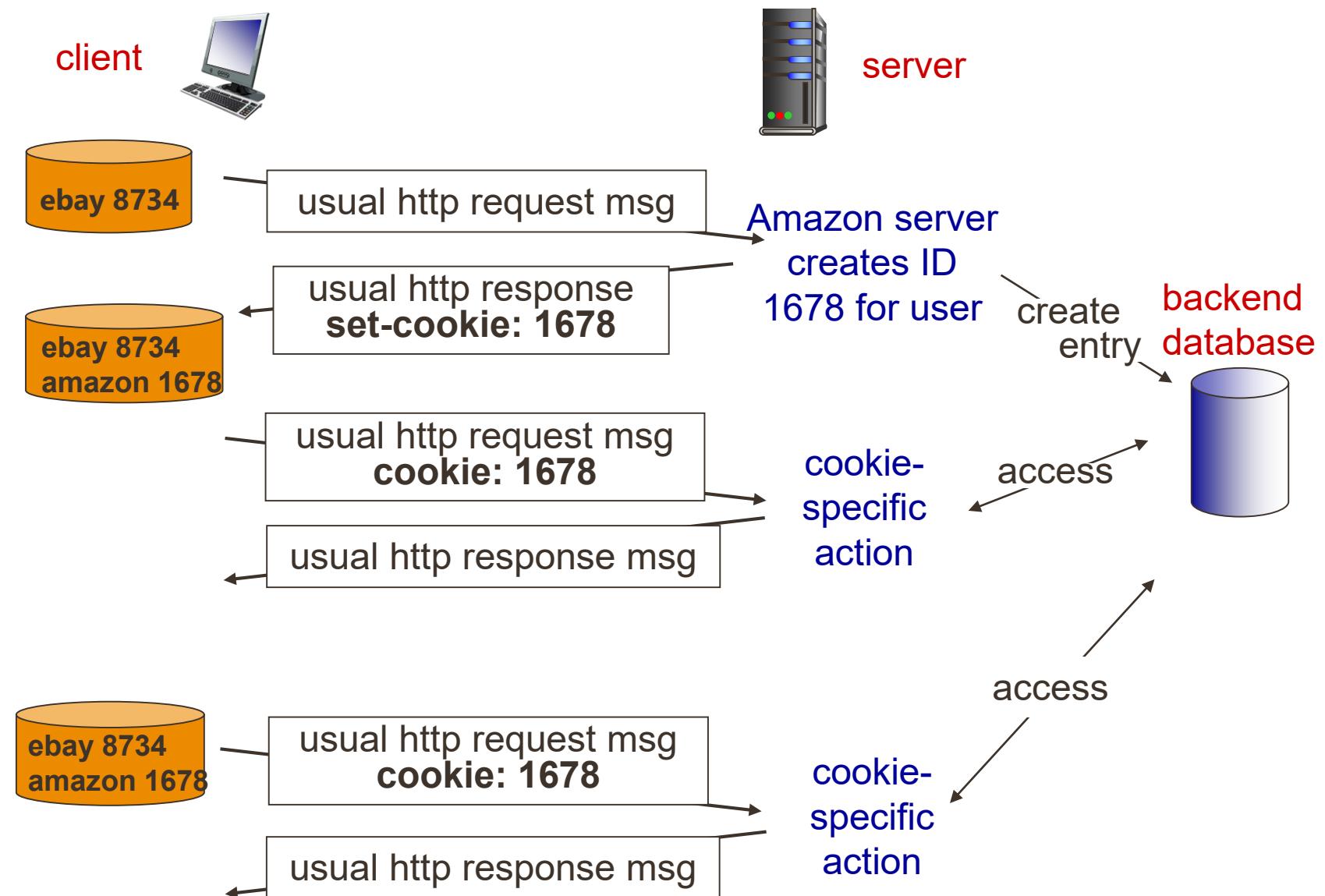


Cookies

Keeping “state”

Cookie File

One week
later



Cookies: keeping “state”

- **How to keep “state”:**

- ✓ **protocol endpoints:**

- maintain state at sender/receiver over multiple transactions

- ✓ **cookies:**

- http messages carry state

- **Cookies can be used for:**

- ✓ authorization
 - ✓ shopping carts
 - ✓ recommendations
 - ✓ user session state (Web e-mail)



Cookies:

- permits sites to learn a lot about you
- stored in clear text

Web Caching

- Typically cache is installed by ISP
(University, Company, Residential ISP)

- **Why Web caching?**

- ✓ Reduce response time for client request
- ✓ Reduce traffic on an institution's access link
- ✓ Internet dense with caches enables "poor" content providers to effectively deliver content (so too does P2P file sharing)

Web Caches: Proxy Server



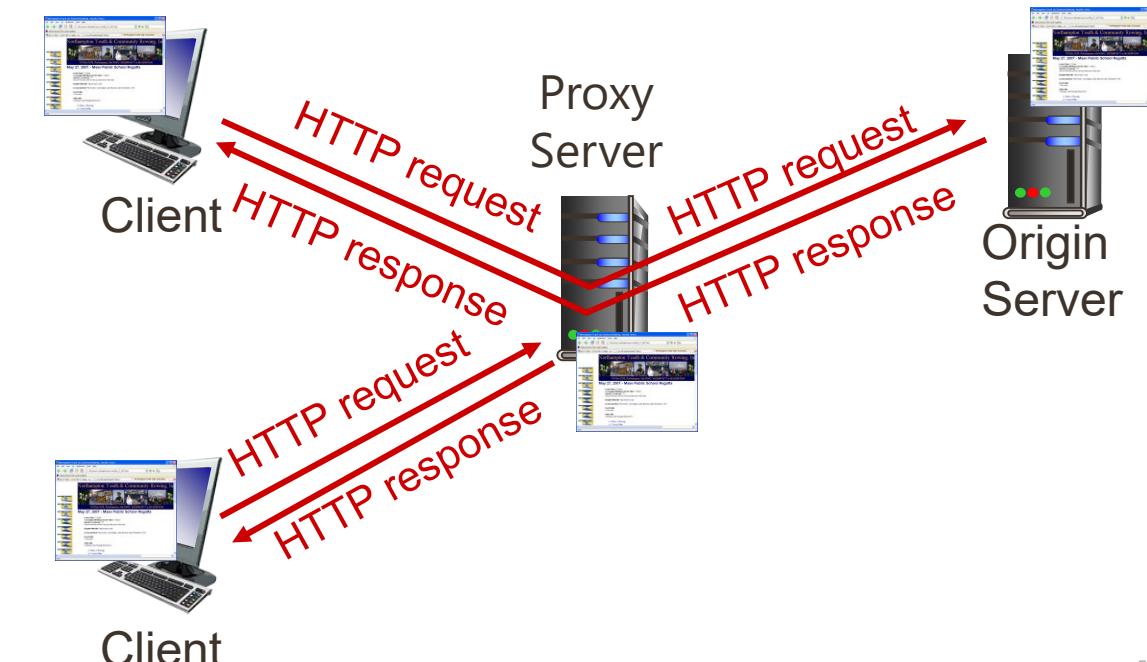
Goal: satisfy client request without involving origin server

- Browser (client) -> **Proxy Server (server)**

- ✓ Object in cache, return object
- ✓ Else, request from origin server

- **Proxy Server (client)**

- > Origin Server (server)

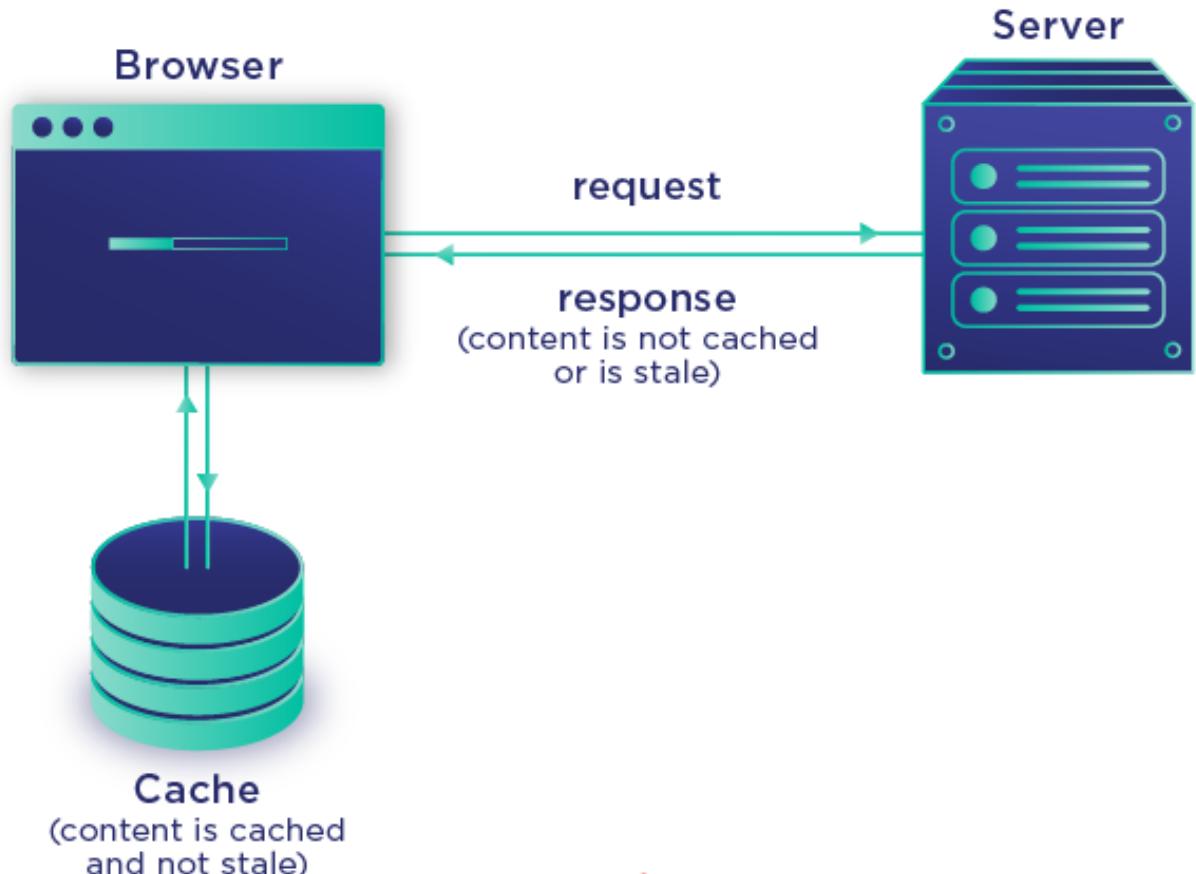


Web Caches: Browser Cache



Goal: satisfy client request without involving origin server

- Typically cache **static assets**
 - ✓ Parts of a website that do not change from visit to visit
 - ✓ i.e. HTML, CSS, JavaScripts, images, etc.

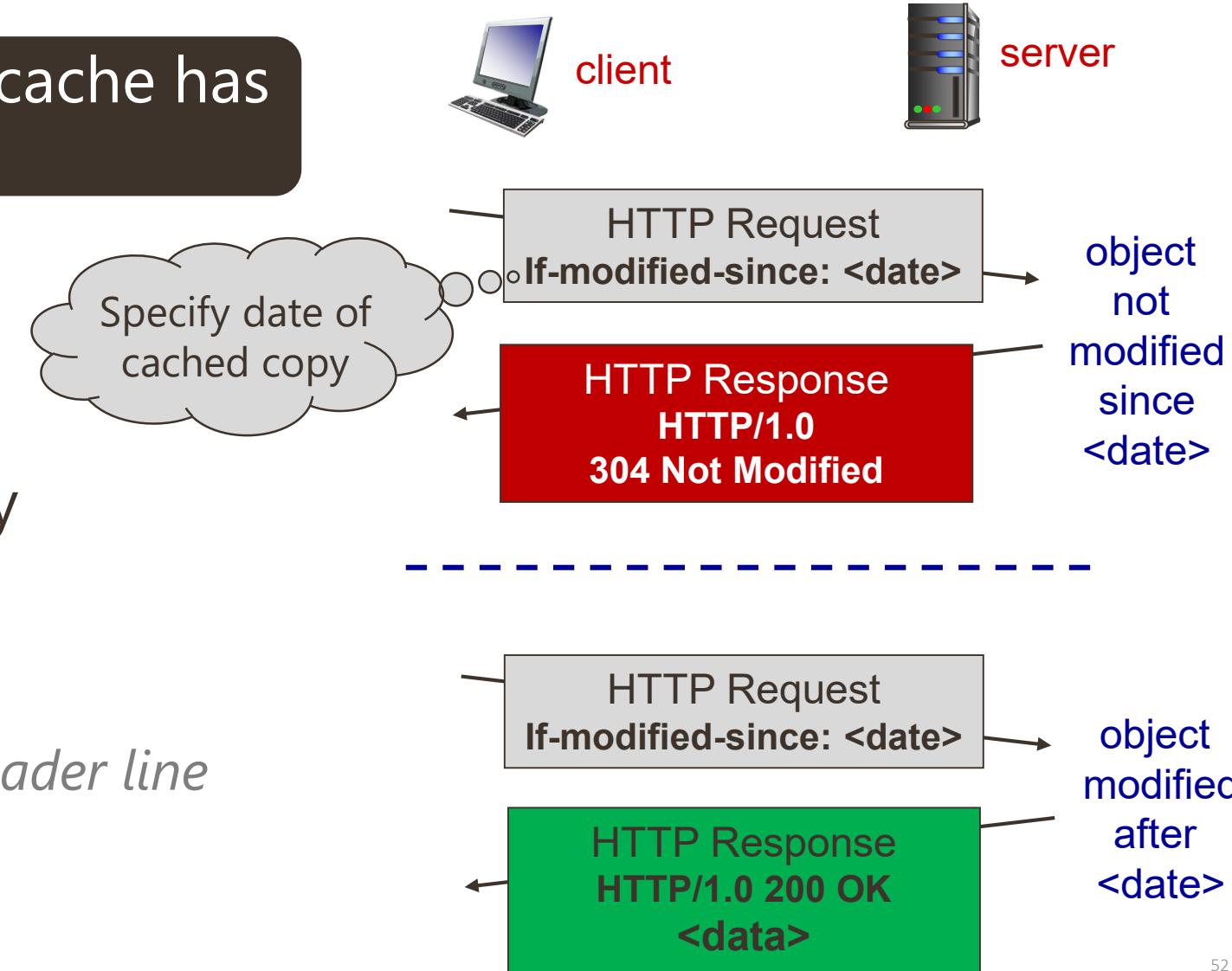


- What to cache? How long?
determined by the webserver

Conditional GET



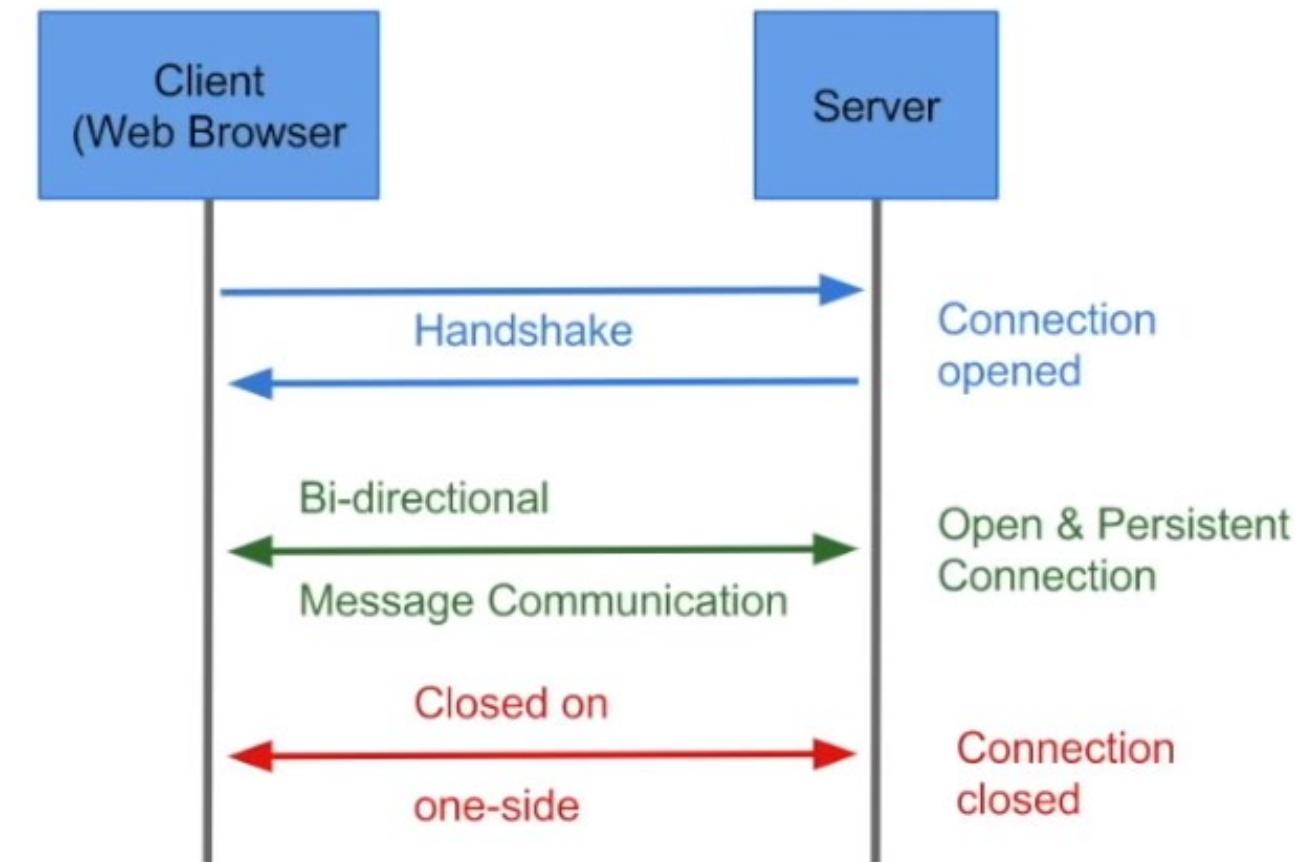
Goal: don't send object if cache has up-to-date cached version



- No object transmission delay
- Lower link utilization
- **If-modified-since: <date>** // header line

HTTP and Web Sockets

- **Bi-directional** (*unlike http uni-directional request-response*)
- **Persistent Connection**, Faster
- **Message Oriented Protocol**
- **For Real-Time** applications
 - ✓ No need to refresh UI/browser





Application Layer Protocols - **SMTP, POP3, IMAP**

- Mail-sending Protocol
 - SMTP
- Mail-Access Protocol
 - HTTP (Web-mail)
 - POP3
 - IMAP

Electronic Mail

1. User Agents

- ✓ composing, editing, reading mail messages
- ✓ **outgoing, incoming** messages stored on server

2. Mail Servers

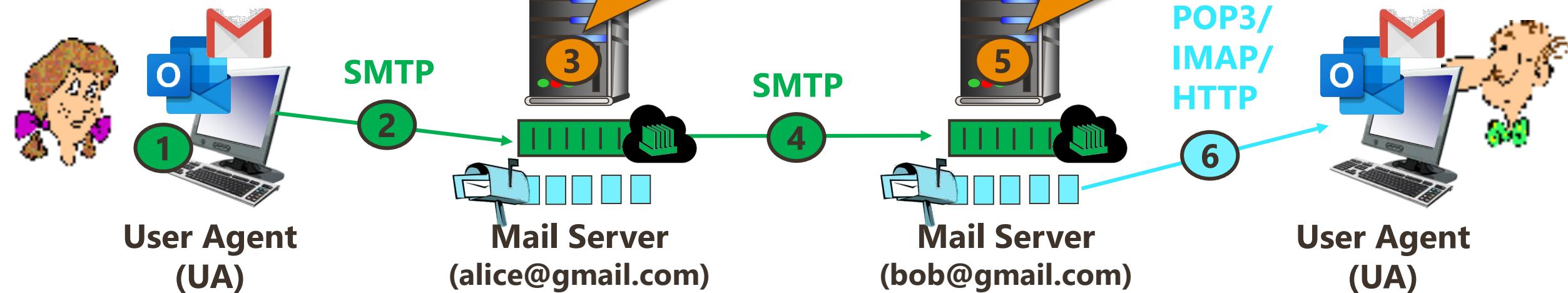
- “Mailbox”:** incoming messages for user
- Message Queue:** outgoing (to be sent) mail messages

3. SMTP: Simple Mail Transfer Protocol

- ✓ **“client”:** sending mail server
- ✓ **“server”:** receiving mail server

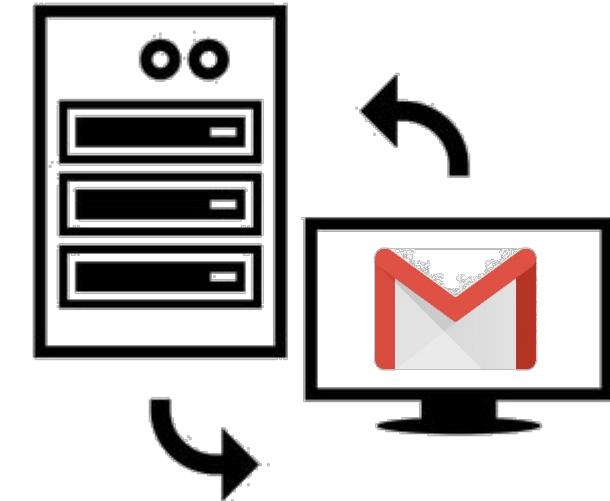


Electronic Mail



Sample SMTP interaction

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```



SMTP [RFC 2821]

- Uses TCP to reliably transfer email message from client to server, **port 25**
- **Direct Transfer:** Sending server to receiving server
- **Messages** must be in **7-bit ASCII**

- 
- **Three phases of transfer**
 1. handshaking (greeting)
 2. transfer of messages
 3. closure



▪ **Command/Response Interaction**
(like HTTP, FTP)

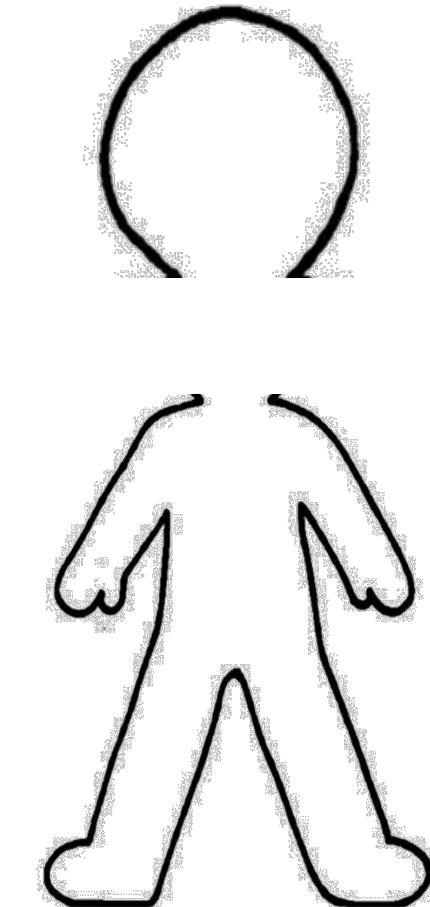
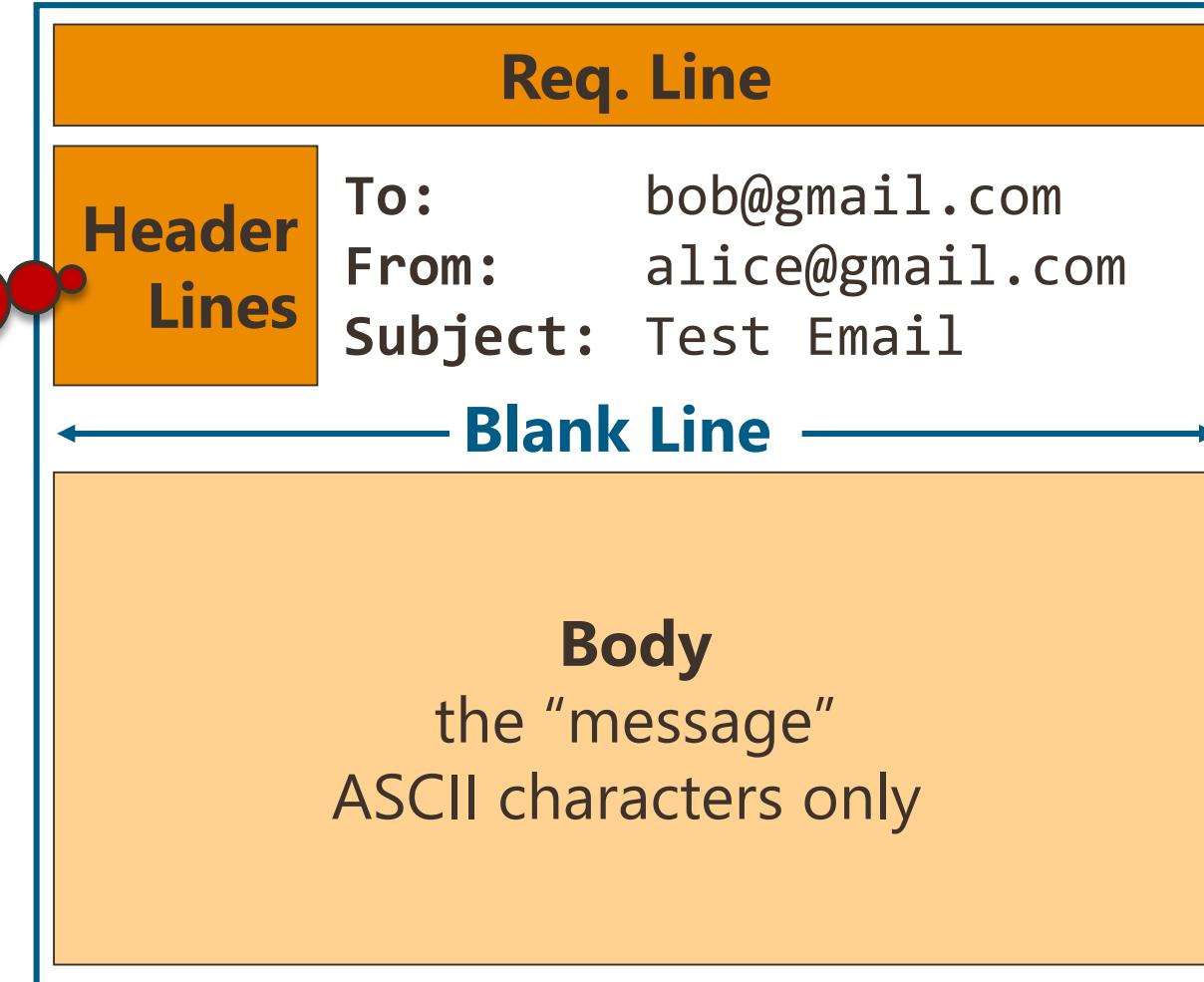
- ✓ **Commands:** ASCII text
- ✓ **Response:** Status code and phrase

SMTP – cont.

- Uses **persistent connections**
- Requires message (header & body) to be in **7-bit ASCII**
- Server uses **CRLF.CRLF** to determine **end of message**
- **Comparison with HTTP:**
 - **HTTP:** pull
 - **HTTP:** each object encapsulated in its own response message
 - **SMTP:** push
 - **SMTP:** multiple objects sent in multipart message
 - both have ASCII command/response interaction, status codes

SMTP Mail Message Format

*different from
SMTP MAIL
FROM, RCPT TO:
commands !*

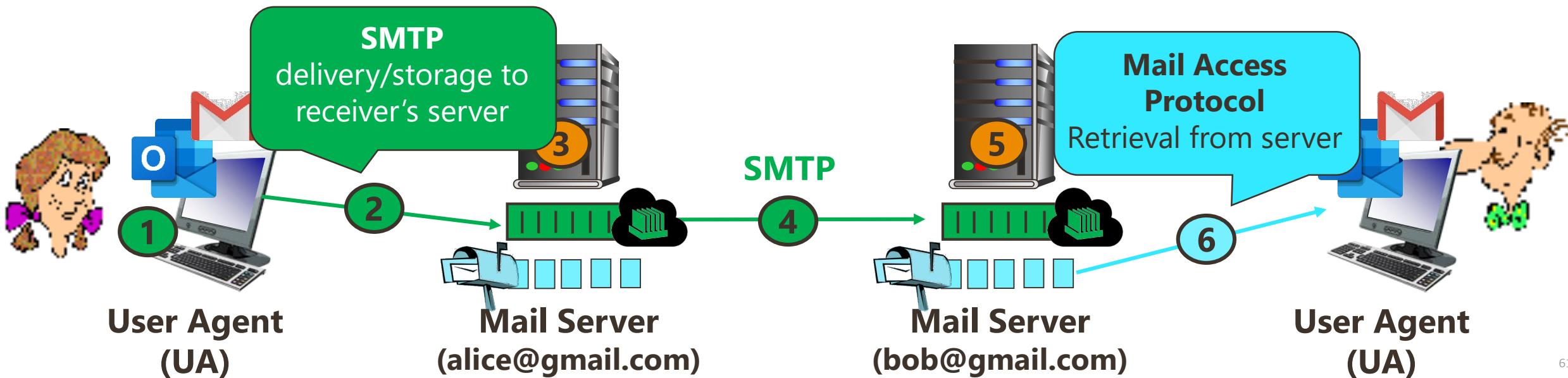


Mail Access Protocols

1. POP: Post Office Protocol: **Authorization, Download**

2. IMAP: Internet Mail Access Protocol: **More Features**
including manipulation of stored messages on server

3. HTTP: Gmail, Hotmail, Yahoo! Mail, etc.





POP3 protocol

Authorization phase

- client commands:
 - **user**: declare username
 - **pass**: password
- server responses
 - +OK
 - -ERR

Transaction phase, client:

- **list**: list message numbers
- **retr**: retrieve message by number
- **dele**: delete
- **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

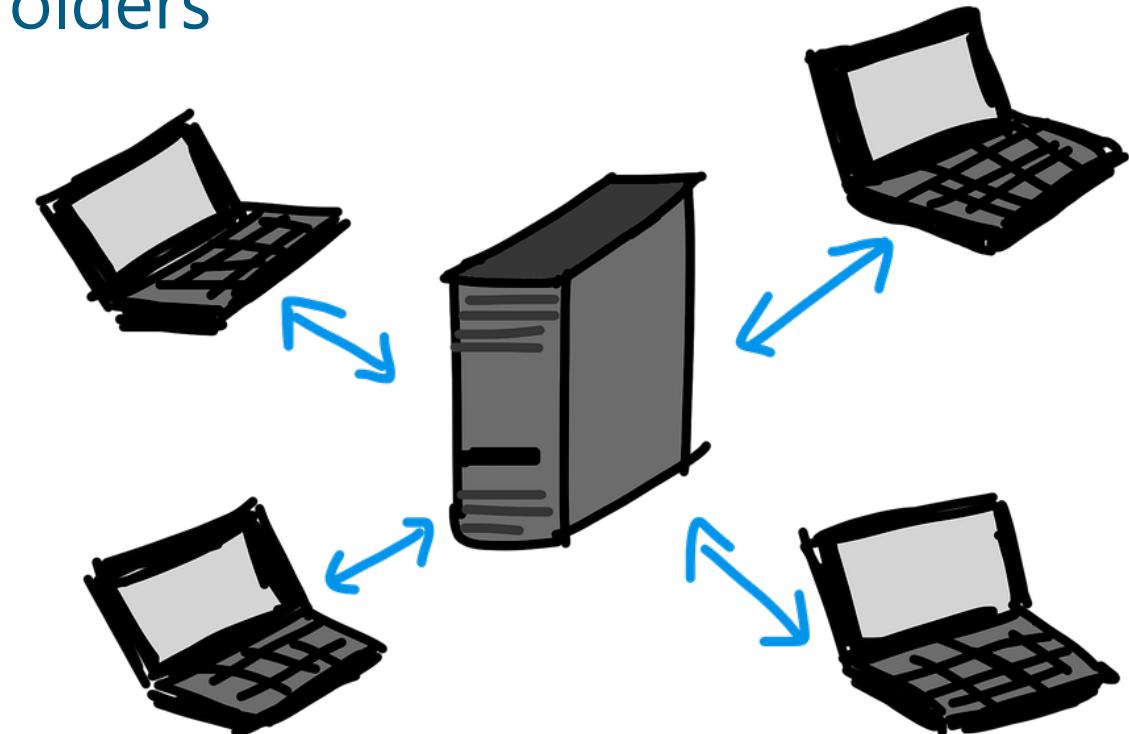
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 2 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

POP3 – cont.

- **POP3 is stateless** across sessions
- **Two Modes:**
 1. **Download and Delete:** Previous example uses this mode
Bob cannot re-read e-mail if he changes client
 2. **Download and Keep:** Copies of messages on different clients

IMAP

- Keeps all messages in one place: at server
- Allows user to organize messages in folders
- **IMAP is stateful** across sessions
 - ✓ Names of folders and mappings between message IDs and folder name





■ Application Layer

- Fundamentals
- App-protocol Contract
- Required Transport Services
- Apps underlying TCP Protocols
- Secure Communication
- Architectures
 - Client-server
 - P2P

■ App Protocols – Telnet

- NVT, VTP
- Connections
- Highlights

■ App Protocols – FTP

- Connections
- Commands and Responses

■ App Protocols – HTTP

- Web Basics
 - WWW
- HTTP
 - Connections (persistent/non-persistent)
 - Messages (http request / http response)
 - HTTP Request Methods
 - HTTP 1.1 Problems
 - HTTP 2.0
 - Maintaining State (Cookies)
 - Web Caching (Browser Cache, Proxy Server)
 - Conditional GET
 - Web Sockets

■ App Protocols – SMTP, POP3, IMAP

- Mail-sending Protocol
 - SMTP
- Mail-Access Protocol
 - HTTP (Web-mail)
 - POP3
 - IMAP

THANK YOU

Make tomorrow better.

Application Layer II

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 10

Semester 1, 2021



Application Layer Protocol - **DHCP**

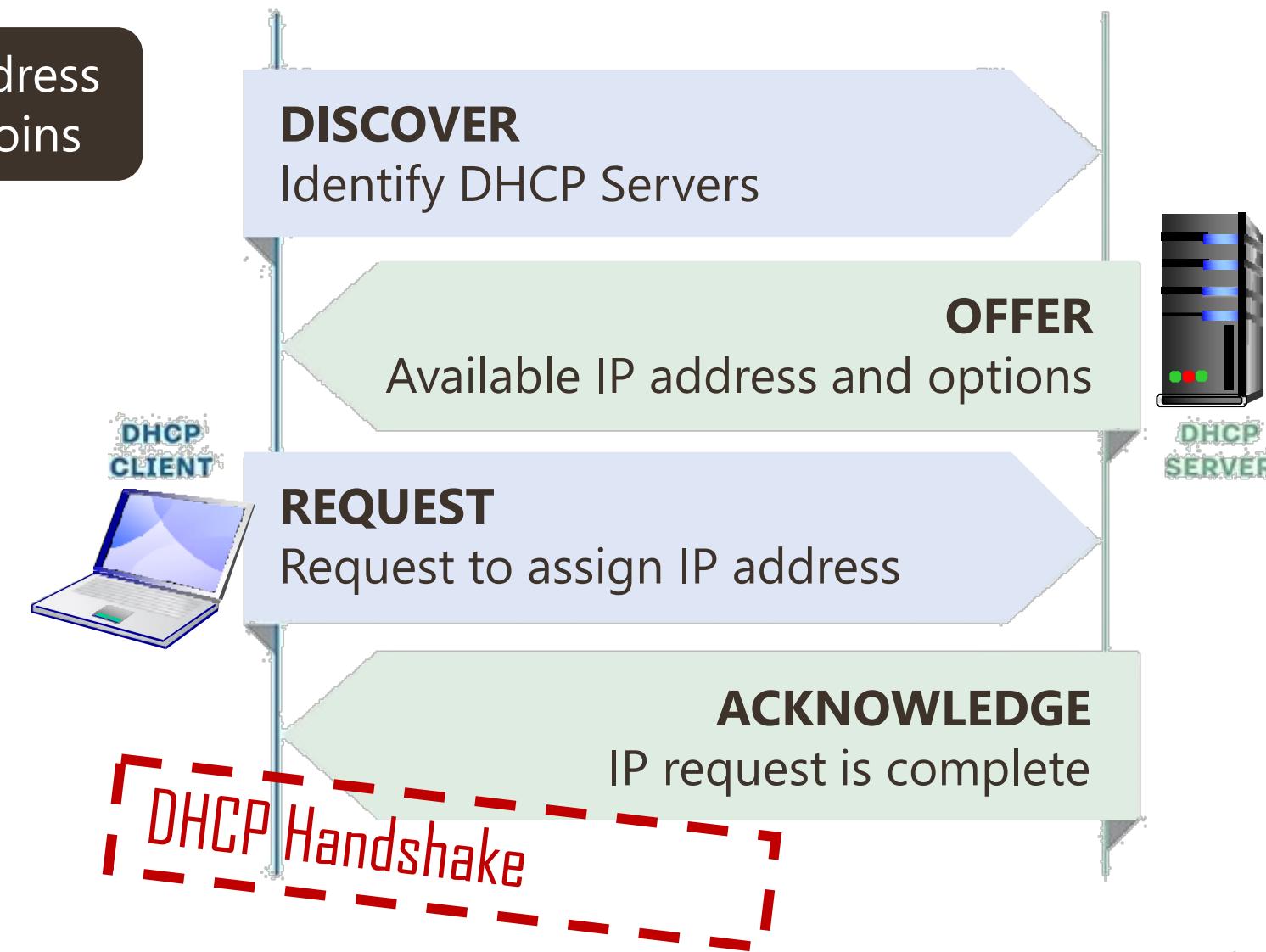
- Fundamentals
- 4-way Handshake
 - DHCP Discover
 - DHCP Offer
 - DHCP Request
 - DHCP Ack

DHCP: Dynamic Host Configuration Protocol



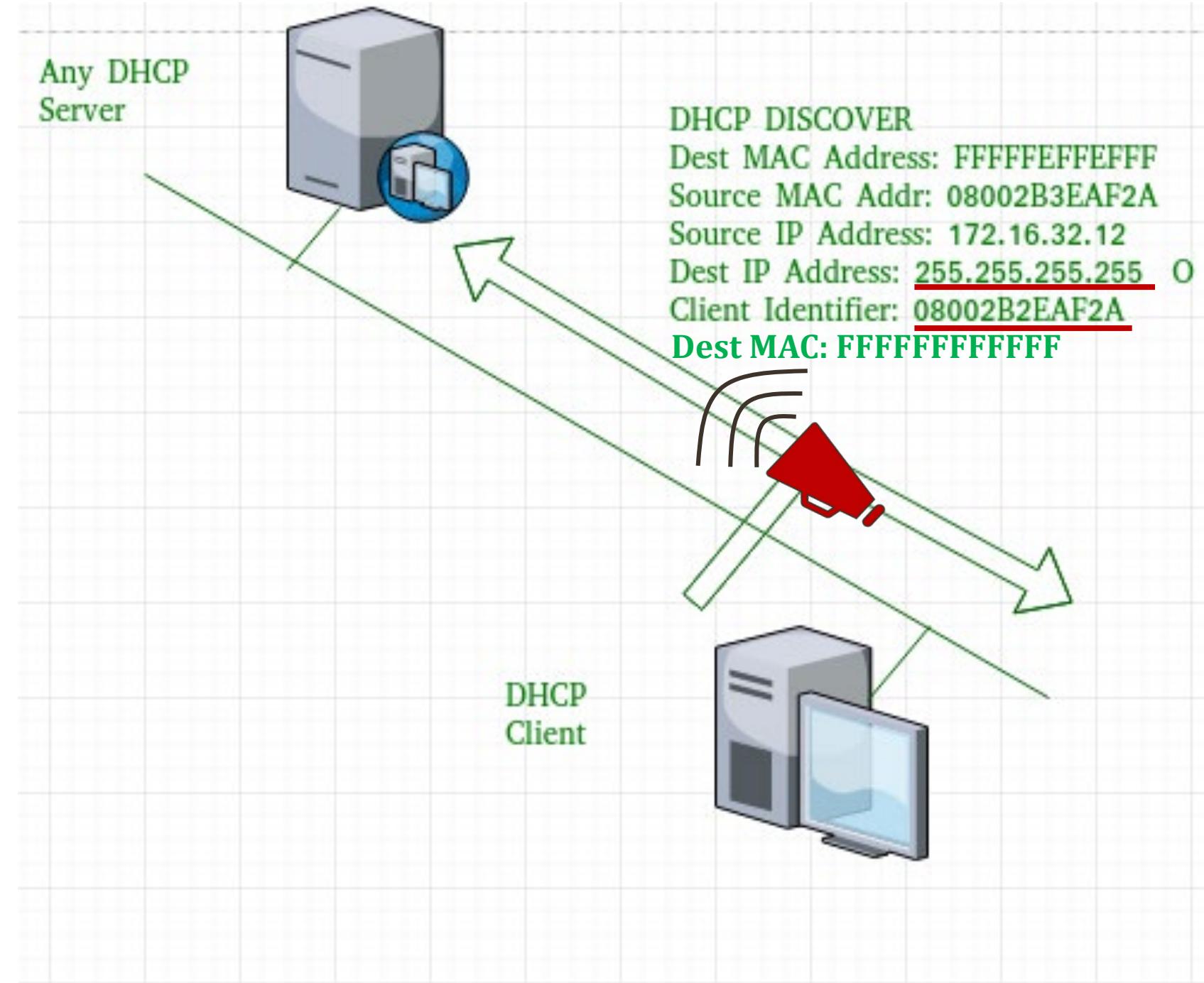
Goal: Dynamically obtain an IP address from network server when a host joins

- **Can renew** its lease on IP address in use
- **Allows reuse** of addresses (only hold address while connected /"on")
- Support for mobile users who want to join network (more shortly)
- Uses **UDP Broadcast**



1 DHCP Discover

Generated by Client host
to discover any DHCP server/servers in a network



2 DHCP Offer

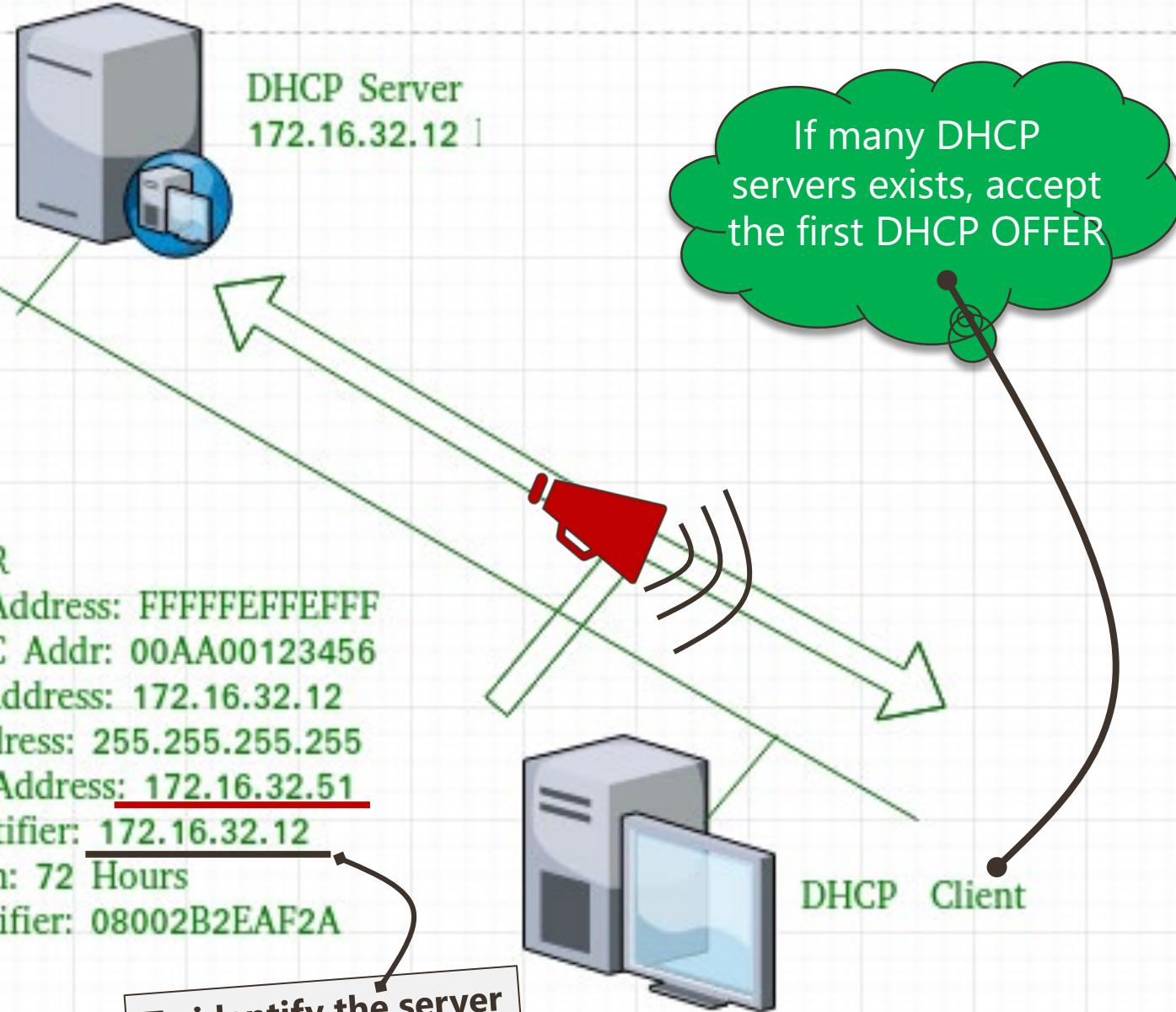
Server's Offer (UDP Broadcast)

- **Unleased IP Address**
- **Server Identifier:** To identify the server if there are many
- **Other TCP configuration**
(Default GW, DNS server) information.

Renew IP
after
3 days

DHCPOFFER
 Dest MAC Address: FFFFFFFFFFFF
 Source MAC Addr: 00AA00123456
 Source IP Address: 172.16.32.12
 Dest IP Address: 255.255.255.255
 Offered IP Address: 172.16.32.51
 Server Identifier: 172.16.32.12
 lease Length: 72 Hours
 Client Identifier: 08002B2EAF2A

To identify the server
if there are many





Other TCP Configuration

ipconfig /all

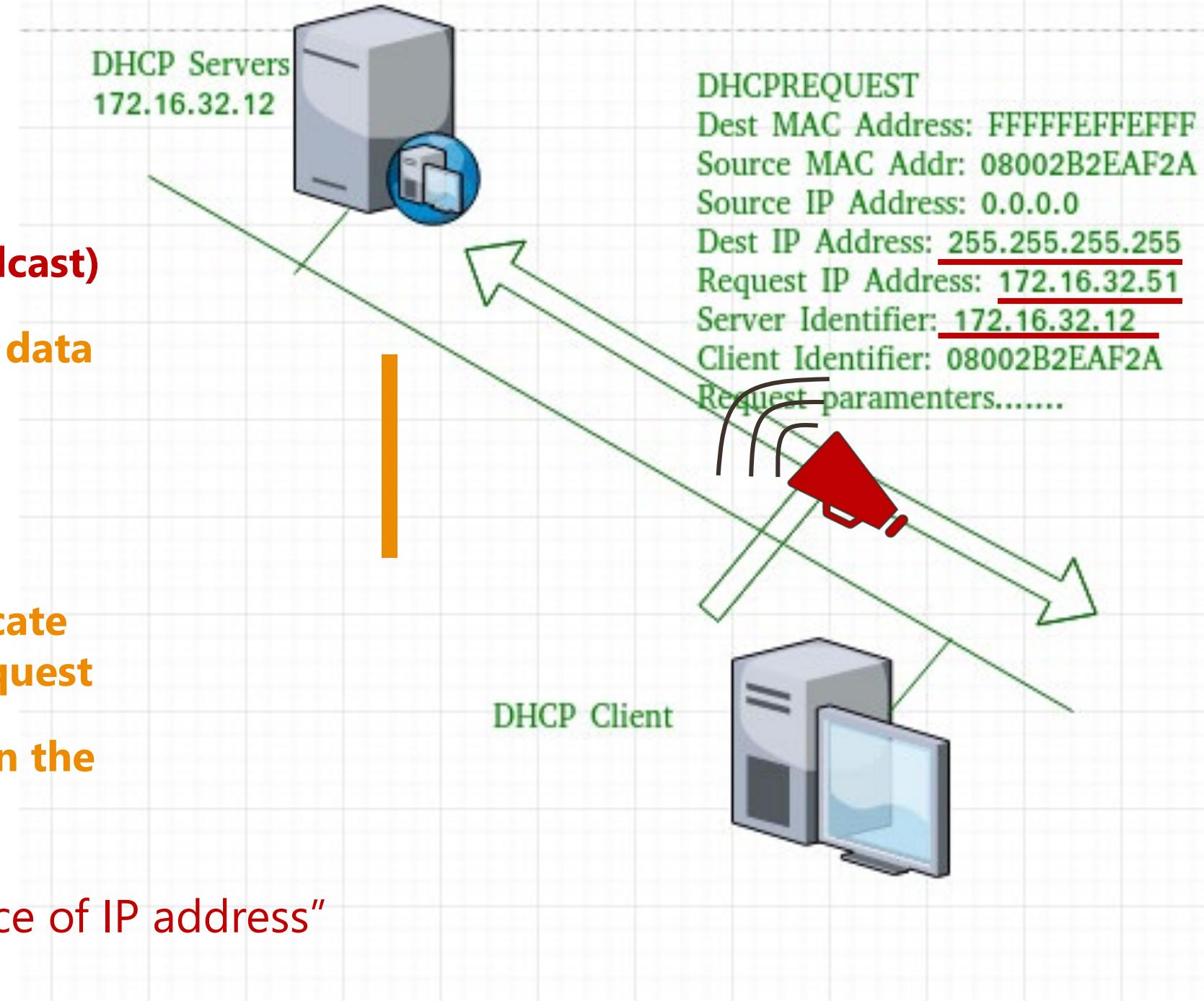
Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix . : stmary.local
Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
Physical Address. . . . . : 88-B1-11-37-25-67
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::55d0:f8cb:699e:fa1b%18(PREFERRED)
IPv4 Address. . . . . : 10.172.112.109(PREFERRED)
Subnet Mask . . . . . : 255.255.248.0
Lease Obtained. . . . . : Monday, March 26, 2018 8:35:05 PM
Lease Expires . . . . . : Wednesday, March 28, 2018 7:19:47 PM
Default Gateway . . . . . : 10.172.112.1
DHCP Server . . . . . : 172.16.13.12
DHCPv6 IAID . . . . . : 143175953
DHCPv6 Client DUID. . . . . : 00-01-00-01-21-2D-EC-36-54-E1-AD-5D-00-2F
DNS Servers . . . . . : 172.16.13.12
                                                172.16.13.10
NetBIOS over Tcpip. . . . . : Enabled
```

3 DHCP Request

Client request (UDP Broadcast)

- Network configuration data including an IP address
- If more than one offers were received:
 - Select one and indicate the server in the request
 - Inform all servers on the selection



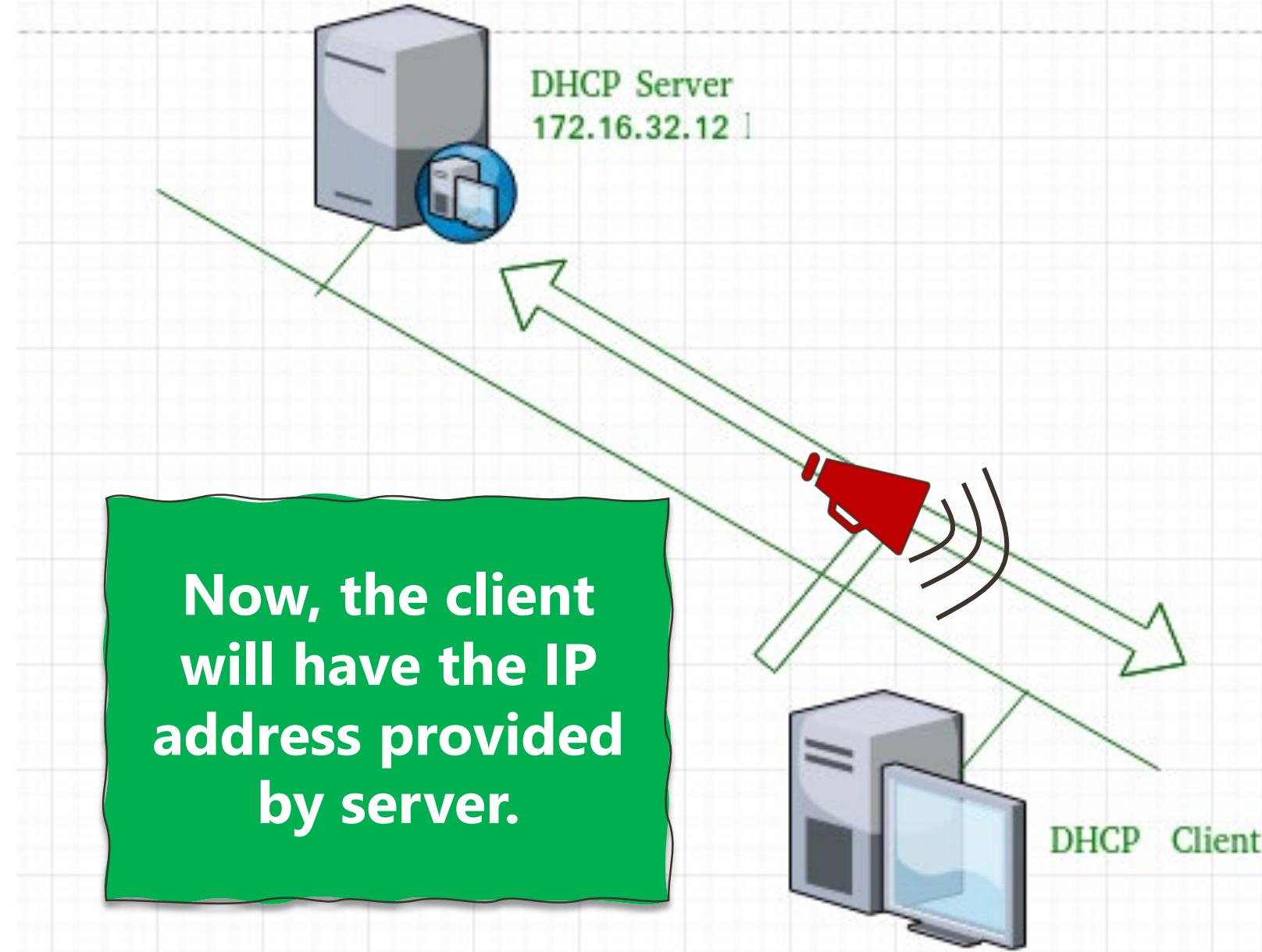
"showing the acceptance of IP address"

4 DHCP Ack

Server will make:

- **Entry with specified client ID**
- **Bind the IP address offered with lease time**

Now, the client will have the IP address provided by server.





Domain Name System

- Domain Name System (DNS)
 - hosts File
- Main Elements
 - Domain Name Space
 - Name Server
 - Resolvers
- DNS Database
- DNS Registrar
- Web Hosting

DNS

- Application program refers to **host by ASCII** string names:

name: **ark.cs.curtin.edu.au**

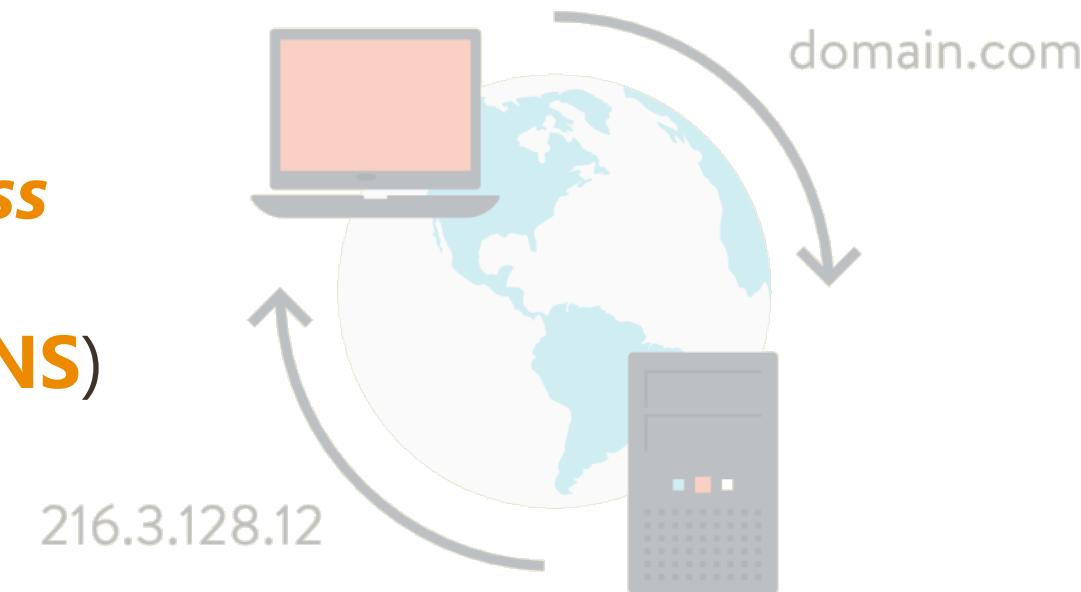
IP address: **134.7.1.10**

- Network does not understand ASCII names

- Given a **machine name** - *need some mechanism to convert to an IP address*

- Today:** uses Domain Name System (**DNS**)

- ✓ Defined in RFC 1034 and RFC 1035
- ✓ A directory lookup service



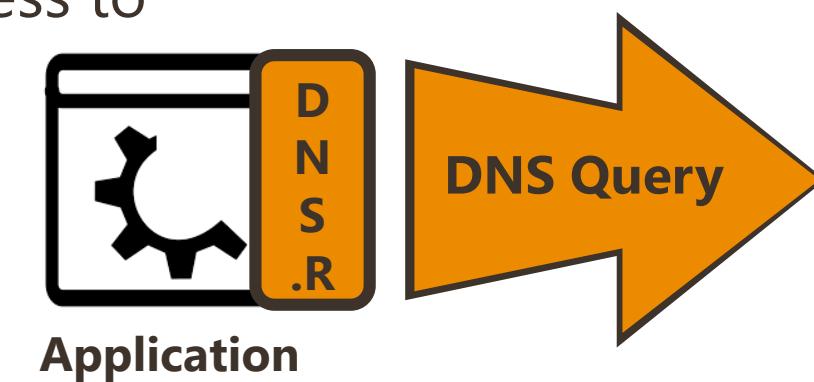
Hosts File

- Path
“C:\Windows\System32\drivers\etc\hosts”
- Plain text file
- Maps hostnames to IP addresses
- Originally a file named HOSTS.TXT was manually maintained for ARPANET



DNS – Cont.

- **DNS plays a support role to other applications**
 - Application program calls a library procedure – the **resolver** to resolve a remote host name.
 - **Resolver** then queries the local **DNS server**
 - DNS server answers the query with an IP address
 - Application program can then use the IP address to communicate with the remote host
 - All DNS queries **use UDP**



DNS Main Elements

1. Domain Name Space

2. Name Servers

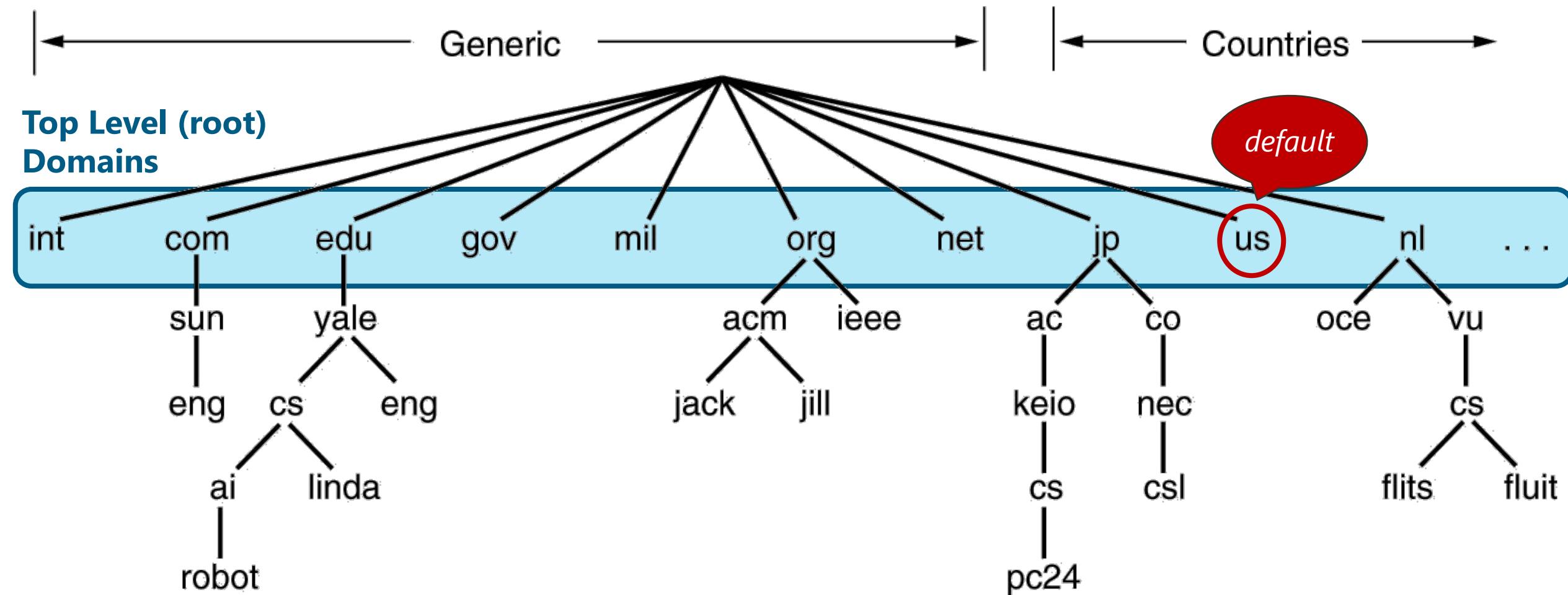
3. Resolvers

1. Domain Name Space

- DNS uses a **hierarchical, domain-based naming scheme** to identify resources on the Internet.
- At the top are a small number of domains that **encompass the entire internet**
 - ✓ Controlled by **ICANN**
(International corporation for assigned names and number)



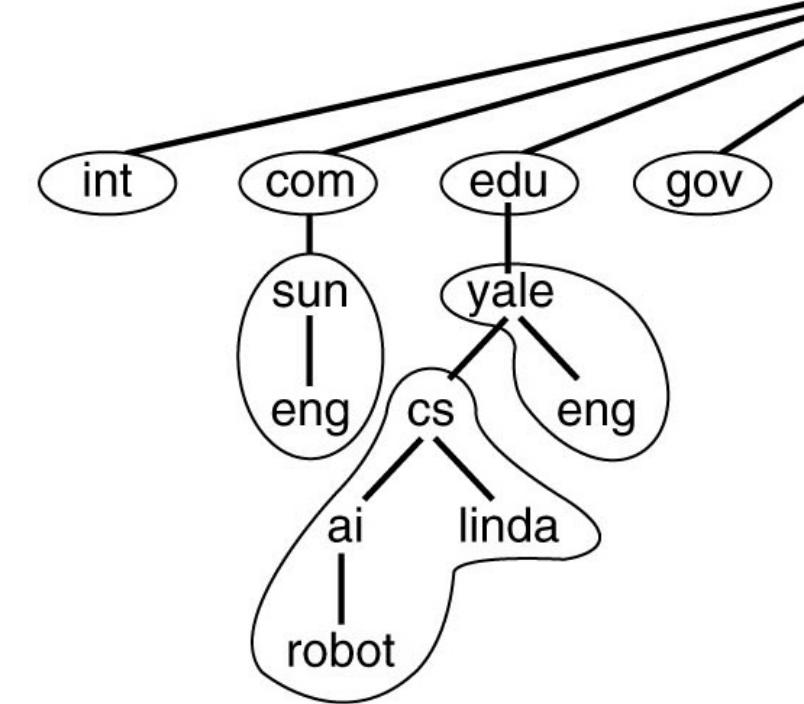
Internet Domain Name Space



robot.ai.cs.yale.edu.us

1. Domain Name Space – cont.

- Domain names are **case insensitive**,
 - *Edu, edu, EDU*
- Component names can be up to 63 characters long and the full path cannot exceed 255 characters
- Naming follows **organizational boundary, not physical network!**
 - One domain may consist of multiple subnets & vice versa



2. Name Servers

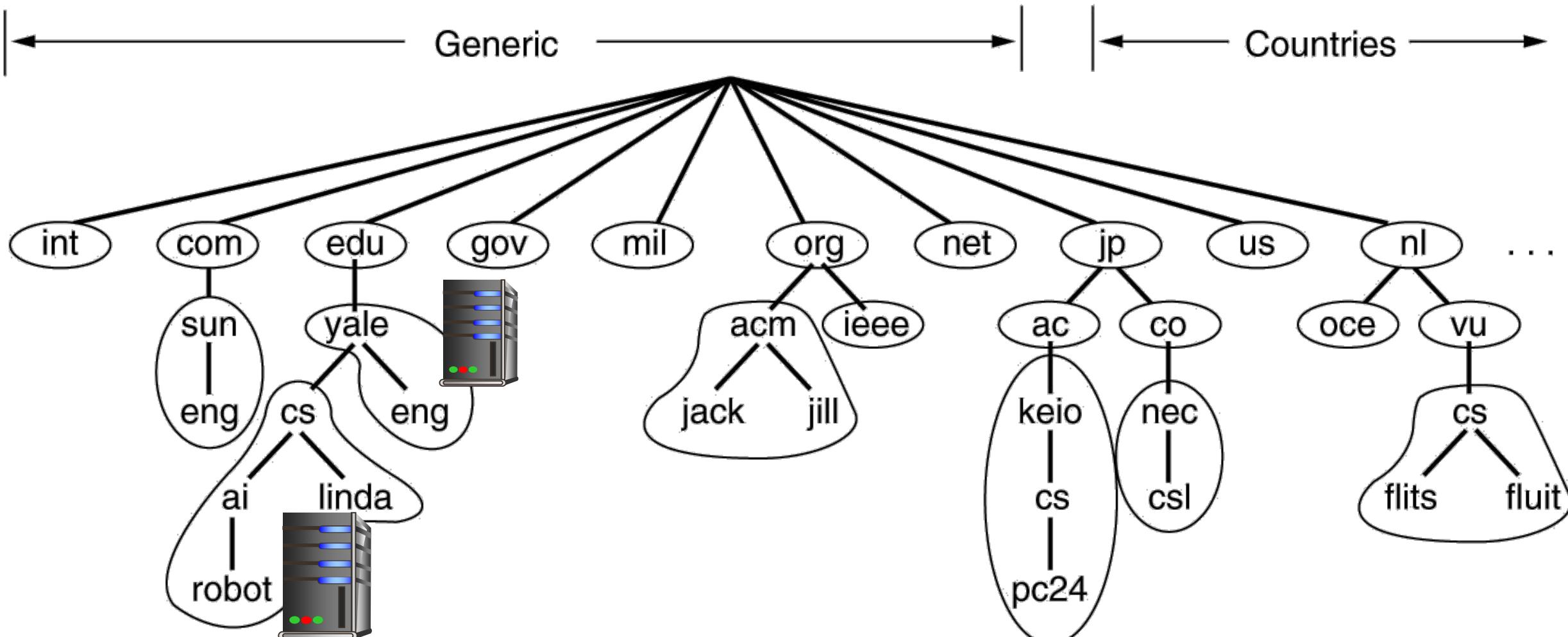
- **Distributed Approach:** dividing into **non-overlapping "zones"**
- **Each Zone:**
 - **Contains some part of the DNS tree**
 - Managed by **one or a set of DNS servers** – normally one primary server or/and one or more secondary servers
 - **Managing server** does not have to belong to the zone it manages
 - **Zone boundaries** are the decision of the **zone administrator**

For Example:

*Curtin zone, managed by Curtin DNS server,
while Computing zone, managed by Computing DNS server*



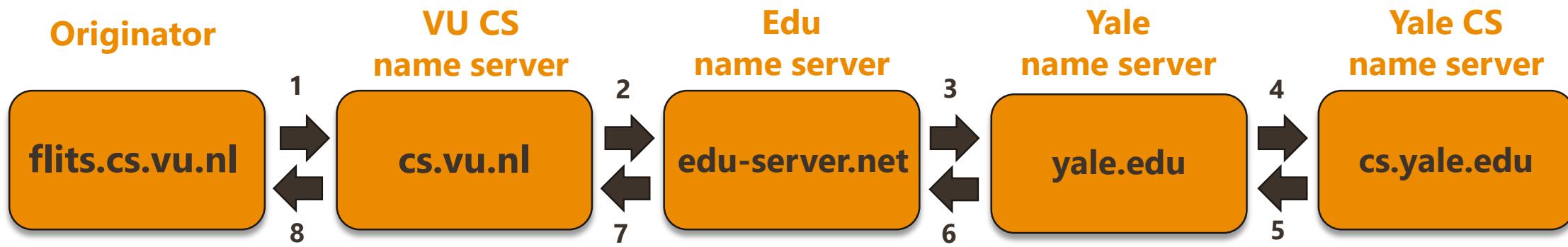
DNS Zones



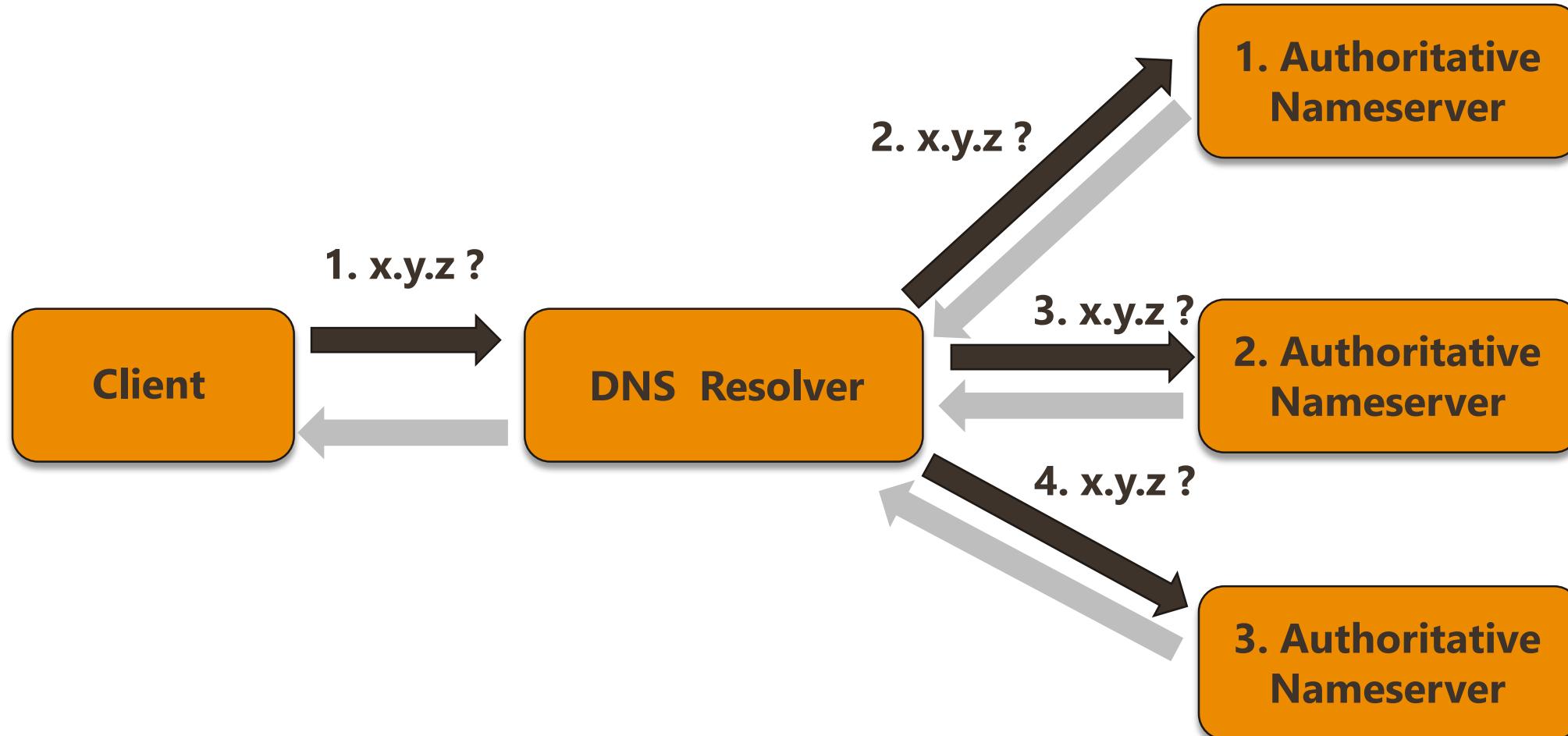
3. Resolvers

- DNS server receives a **type A query** from a resolver (client) for "**www.mit.edu**"
- **If** domain name being queried is managed by the server, returns the matched RR (**authoritative record**)
- **If** a matched RR exists on the cache, returns the cached RR (**non-authoritative answer**)
- **Else,**
 - (recursive query method) **sends a query to** the DNS server for the **top-level domain "edu"**, waits for answers, and forward answers back to client.

Resolver – Implementation 01



Resolver – Implementation 02



DNS Database

- DNS is based on a hierarchical database containing **Resource Records (RRs)**
- Every domain **can have a set of RRs**
- A RR contains – the host name, the IP address and other information about the host
- **Structure** of RR format:

Domain Name	Time To Live	Class	Type	Value
uranus.cs.curtin.edu.au	86400	IN	A	137.7.2.130



RR Types

Domain Name	Time To Live	Class	Type	Value
uranus.cs.curtin.edu.au	86400	IN	TXT	"Lab219 Computing Curtin"
uranus.cs.curtin.edu.au	86400	IN	A	137.7.2.130
uranus.cs.curtin.edu.au	86400	IN	A	137.7.1.112
uranus.cs.curtin.edu.au	86400	IN	MX	1 bike.cs.curtin.edu.au
uranus.cs.curtin.edu.au	86400	IN	MX	2 dns.cs.curtin.edu.au
uranus.cs.curtin.edu.au	86400	IN	HINFO	Redhat Linux 7.1
www.cs.curtin.edu.au	86400	IN	CNAME	kickit.cs.curtin.edu.au

A:	Domain to IP
MX:	Mail Server
HINFO:	Host Info
CNAME:	Canonical Name

DNS SOA (Stat of Authority) RR

- Indicates which Domain Name Server (DNS) is the best source of information for the specified domain.
 - Server Name:** Primary name server for the domain.
 - Mailbox:** Email for the domain.
 - Refresh time:** The number of seconds before the zone refreshes.
 - Retry time:** The number of seconds before a failed refresh is retried.
 - Expiration time:** The time, in seconds, before the data is considered unreliable.
 - Minimum TTL:** The default that applies to all of the resource records in the zone.

Every domain must have an SOA record

Domain Name	Type	Value
oasis.com	SOA	Server Name: dns.oasis.com Mailbox: mail.oasis.com MinTTL: 100 Retry Time: 100 Refresh Time: 100 Expiry Time: 100

Inserting records into **DNS**

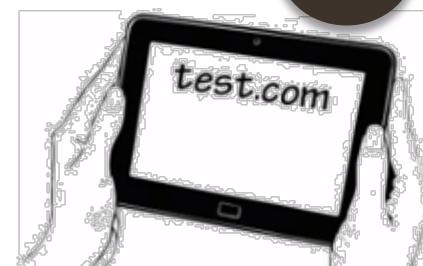
- **Example: new startup “Network Utopia”**
- Register name networkuptopia.com at **DNS registrar** (e.g., Network Solutions)
 - provide names, IP addresses of authoritative name server (primary and secondary)
 - **DNS registrar inserts two RRs into .com TLD server:**
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
- **create** authoritative server **type A record** for www.networkuptopia.com; type **MX record** for networkutopia.com, etc.

Website Hosting on Internet

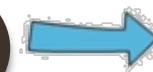
2

Domain Name Registrar (\$)

1. Purchase an available domain name (www.test.com)
2. Set DNS entries for NS (Name Server)

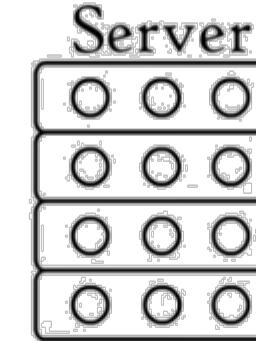


1



Web Hosts (\$)

GoDaddy
HostGator
Bluehost

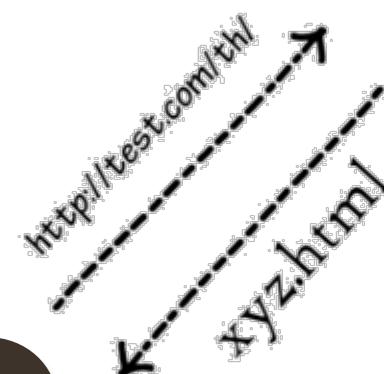


10 GB



**Upload to
remote server**

3



4



domain name test.com

HTML
CSS
images
videos
sounds



Web Search Engines

- Fundamentals
- Ranking Algorithm
- Search Results

Search Engines (SE)

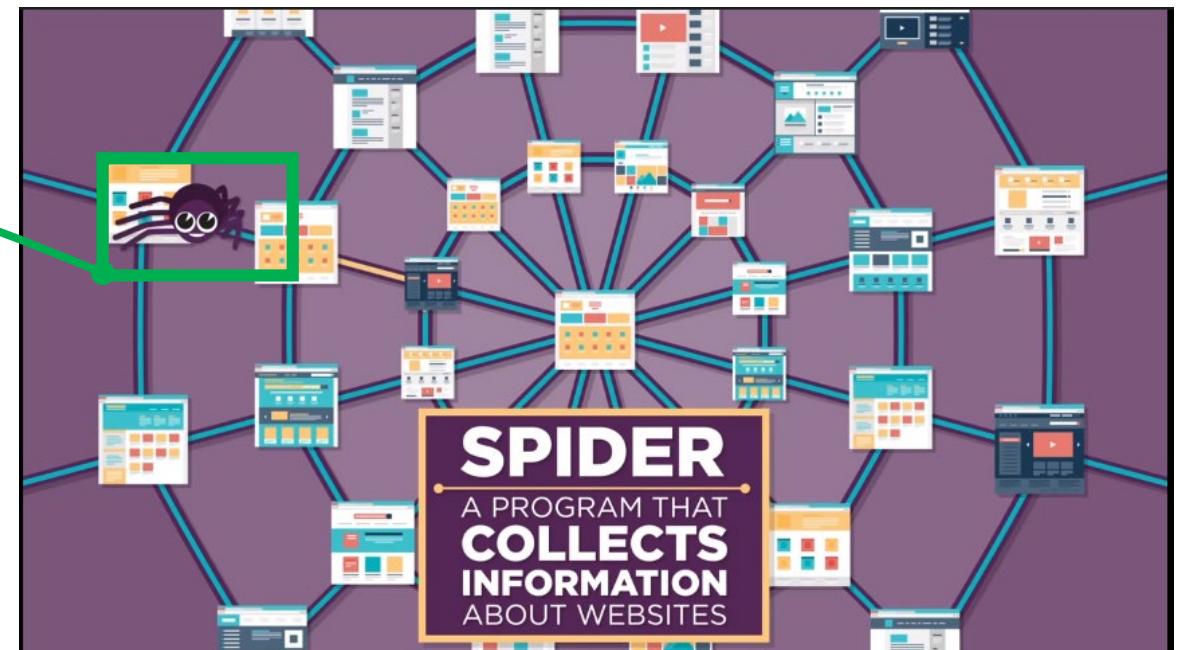
Request ->
Results ?
How ?

- **No real-time search in WWW**
(There is > billion websites in WWW)
- SE constantly **scans** the web in advance

WWW is a web of pages connected
(hyperlinks)

A **web spider (program)** travels
through these links collecting
information

Builds the **Search Index Database**



Search Engines: Ranking Algorithm

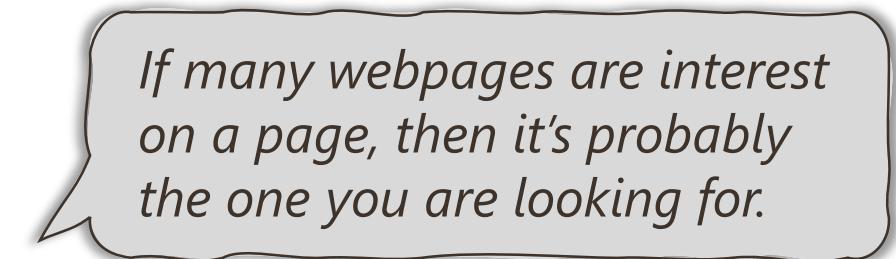
- **Search Query -> Results ? How ?**

- Checks whether search terms shows up in page title
- Sequence of words
- Meta data of the website

- **Page Rank** (Invented by Google - Larry Page)

- Considers how many other webpages linked to a given page
- SE regularly updates the Algorithm to avoid spamming (fake and untrustworthy site)

- **SEO: “Search Engine Optimization” techniques** would help to rank a page higher

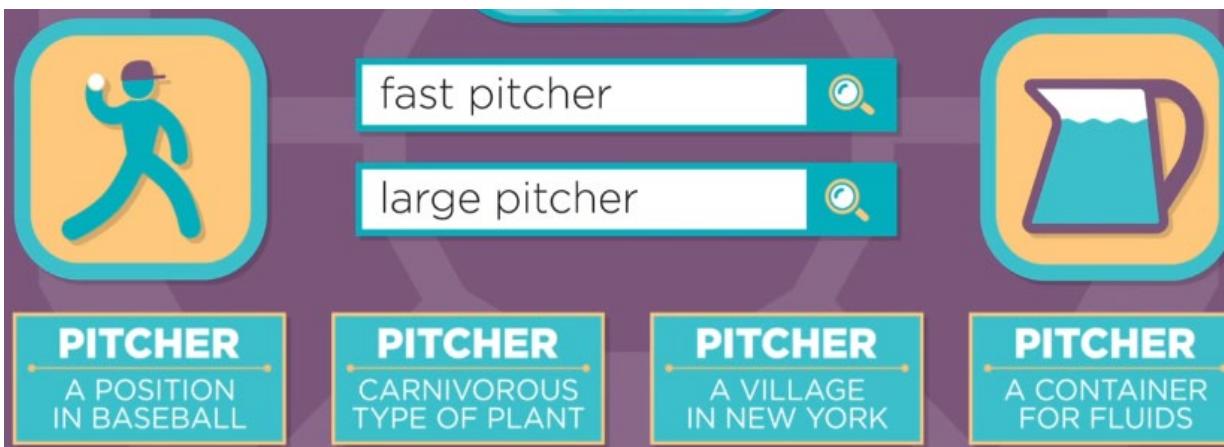


If many webpages are interest on a page, then it's probably the one you are looking for.

Search Engine: Results

To provide better/faster results

- Constant updates to the algorithms
 - Use user's implicit information (GEO location, Past search queries etc.)



Modern Search Engines are Intelligent

- Understand underlying meaning of the words
- Understand Images (Machine Learning)
- Understand Video Sequences (AI)



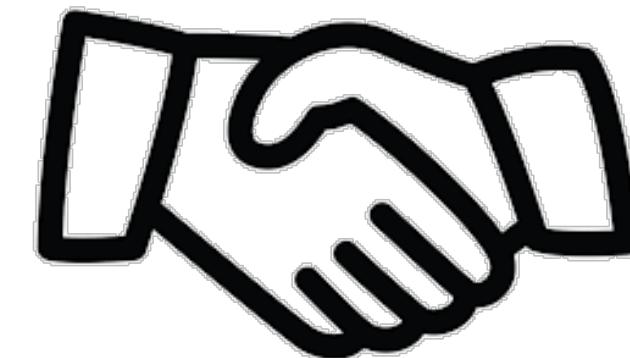
Web Services

- REST API
- URI / URL / URN
- REST API Highlights

Web Services

A web service is a software system designed to support interoperable machine-to-machine interaction over a network

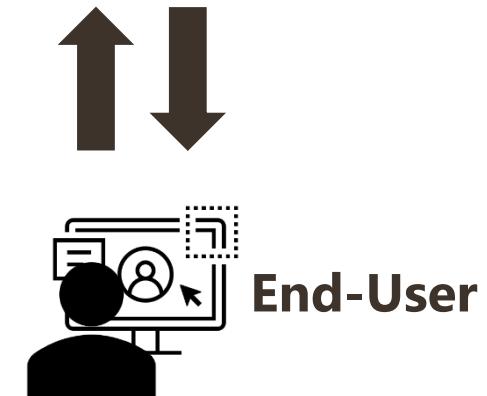
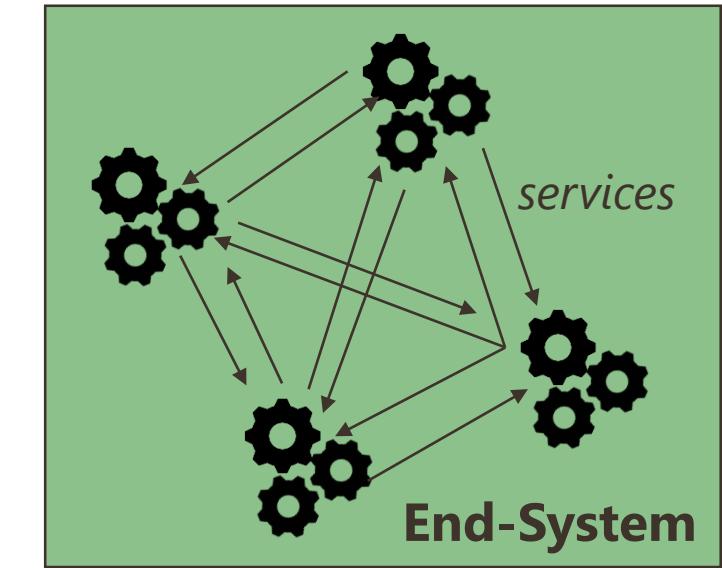
- **A Web Service is defined by rules:**
 - How software component will talk?
 - What kind of messages will be passed?
 - How requests and Responses are handled?
- **Harness** the power of **HTTP**



Web Services: Why we need them?

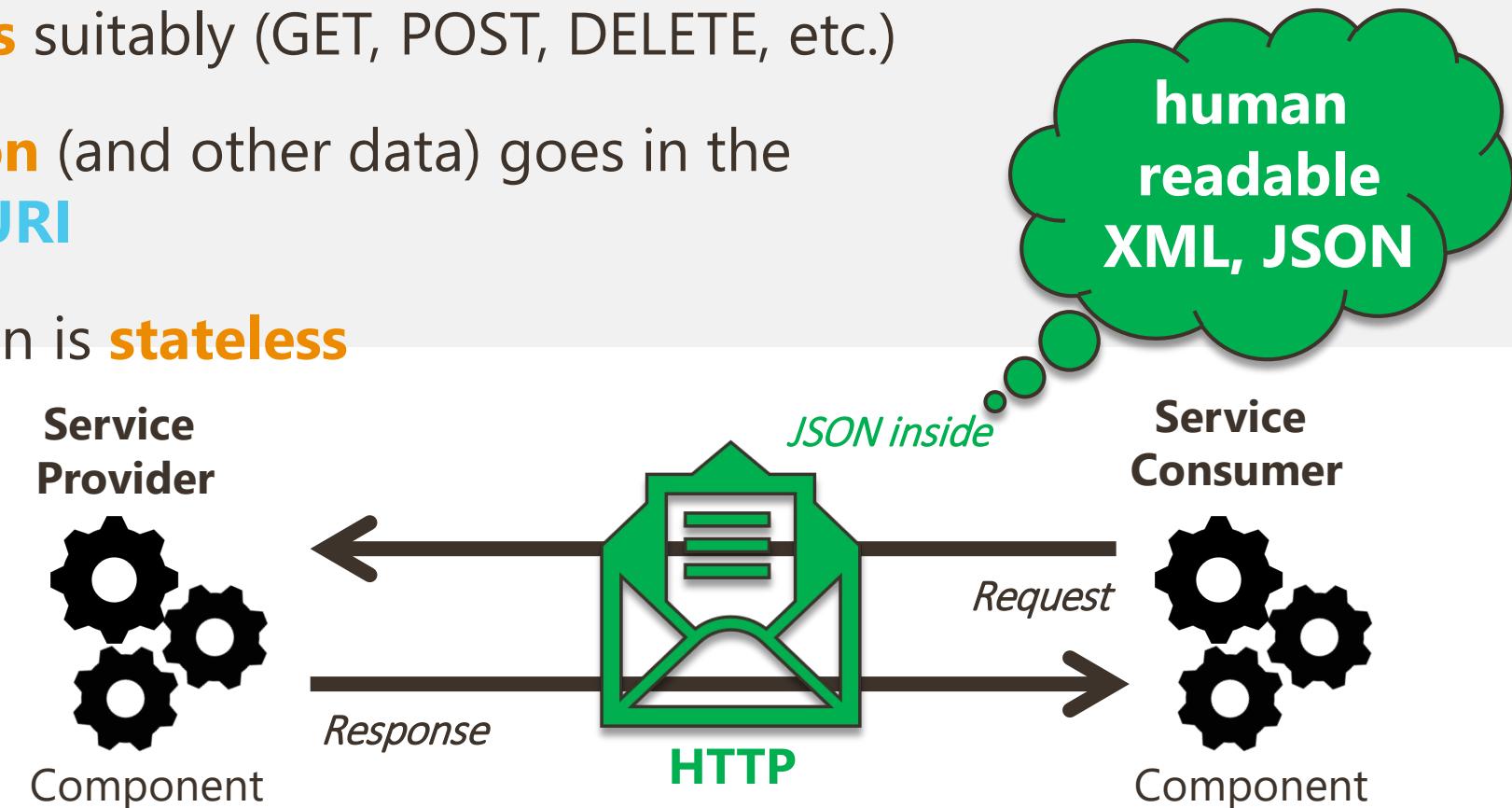
▪ **Distributed Programming**

- allow various applications to talk to each other and share data and services among themselves
- The web is now a very large interlinked set of systems
- Allow component on the web to talk to each other
- Provide better user experience
- Associated with **Service-oriented Architectures (SOA)**
- SOAP (Simple Object Access Protocol) are commonly used (which relies on XML)



REST Web Service / REST API

- Latest is **RE**presentation **S**tate **T**ransfer (**REST**) Protocol
 - an architectural style that makes use of existing and widely adopted technologies, specifically HTTP
- Uses **HTTP methods** suitably (GET, POST, DELETE, etc.)
- **Scoping information** (and other data) goes in the **parameters of the URI**
- **REST** Communication is **stateless**

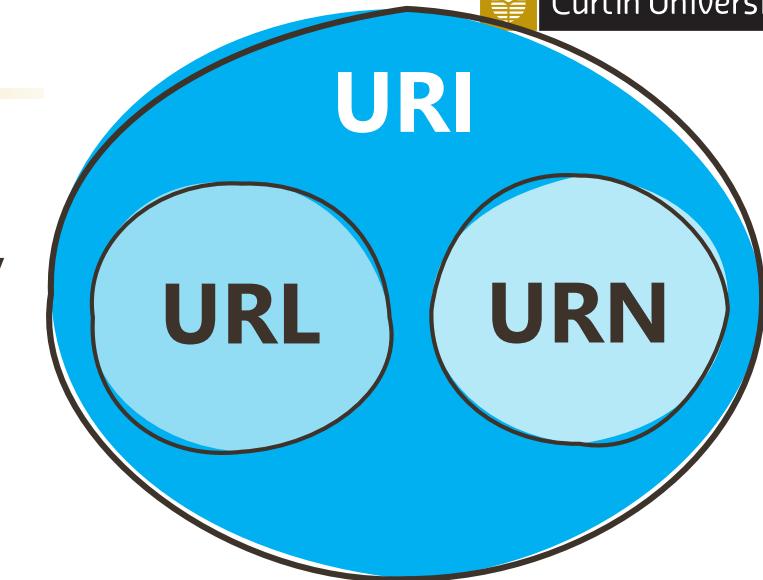


URI / URL / URN

Name:
Flink SideSmack
Address:
12, Beveridge St,
Bentley

Uniform Resource Identifier

- String of characters used to identify resource on the internet,
- by location or by name, or **both**



Uniform Resource Locator

Can contain **Query Strings**

[http://www.myweb.com/signup?
fn=Flick&ls=Sidemack](http://www.myweb.com/signup?fn=Flick&ls=Sidemack)

Can contain **Fragments**

[http://en.wikipedia.org/wiki/
web_developer#external_links](http://en.wikipedia.org/wiki/web_developer#external_links)

Uniform Resource Name

Subset of URLs that include a **name** **within a given space**, but no location

Name:
urn:isbn:0451450523



Peer To Peer P2P

- Fundamentals
- Napster
- Gnutella
- BitTorrent – in depth
- P2P Web

What is peer-to-peer (P2P)?

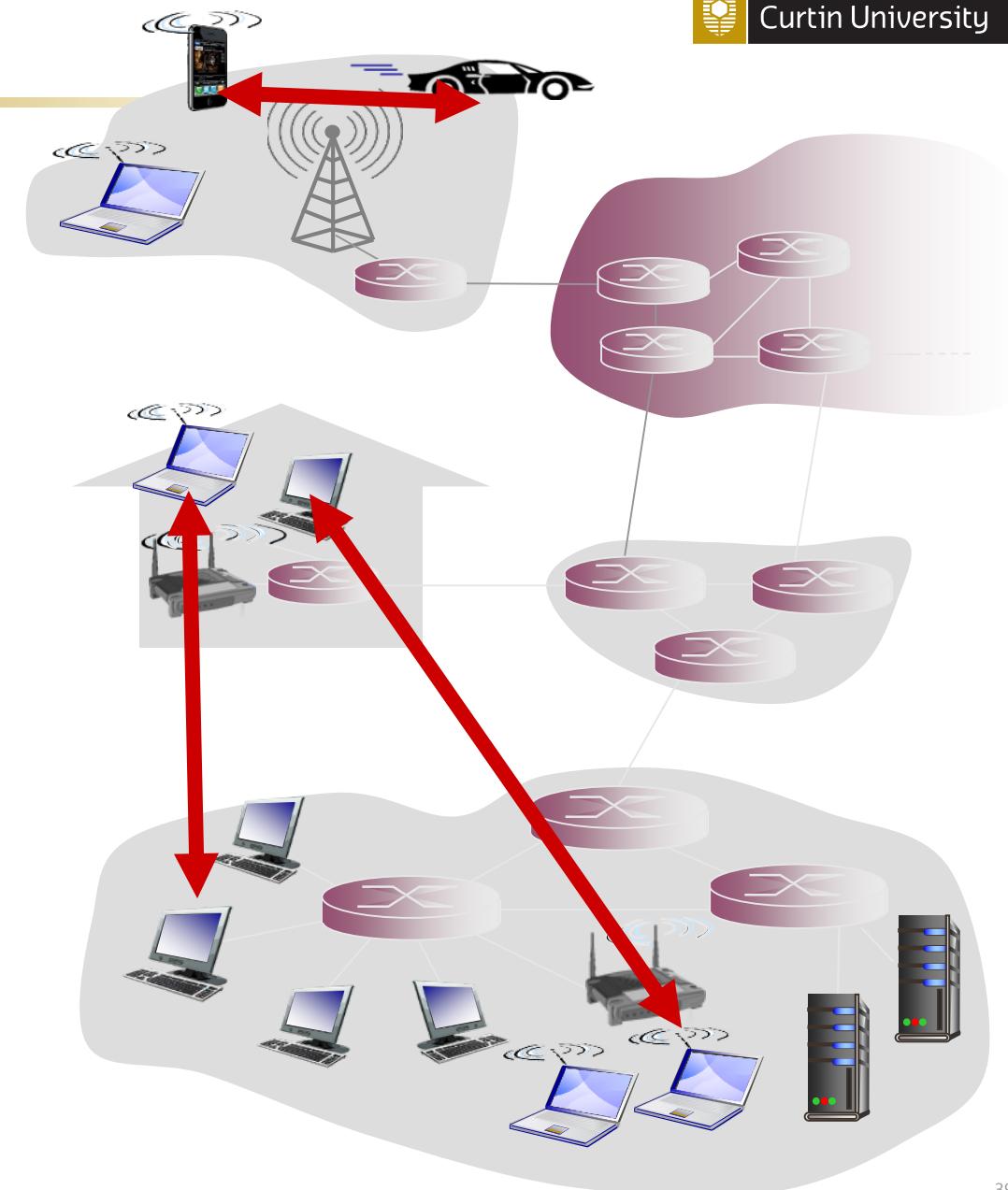
“Peer-to-peer is a way of structuring distributed applications such that the **individual nodes have symmetric roles**. Rather than being divided into clients and servers each with quite distinct roles, in P2P applications **a node may act as both a client and a server**.”



-- Charter of Peer-to-peer Research Group,
IETF/IRTF, June 24, 2004

Pure P2P Architecture

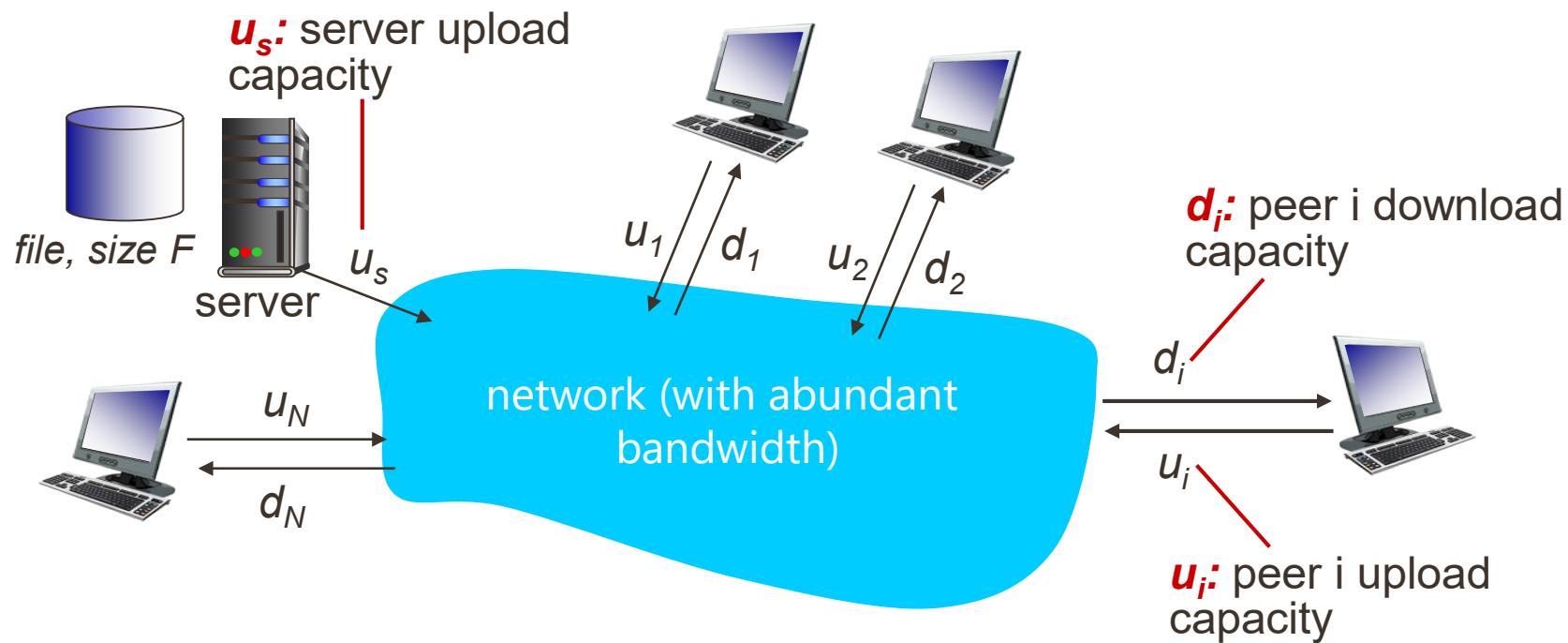
- No **always-on** server
- Arbitrary end systems **directly communicate**
- Peers are intermittently connected and **change IP addresses**



Client-server vs. P2P

Question: how much time to distribute file (size F) from one server to N peers?

- peer upload/download capacity is limited resource

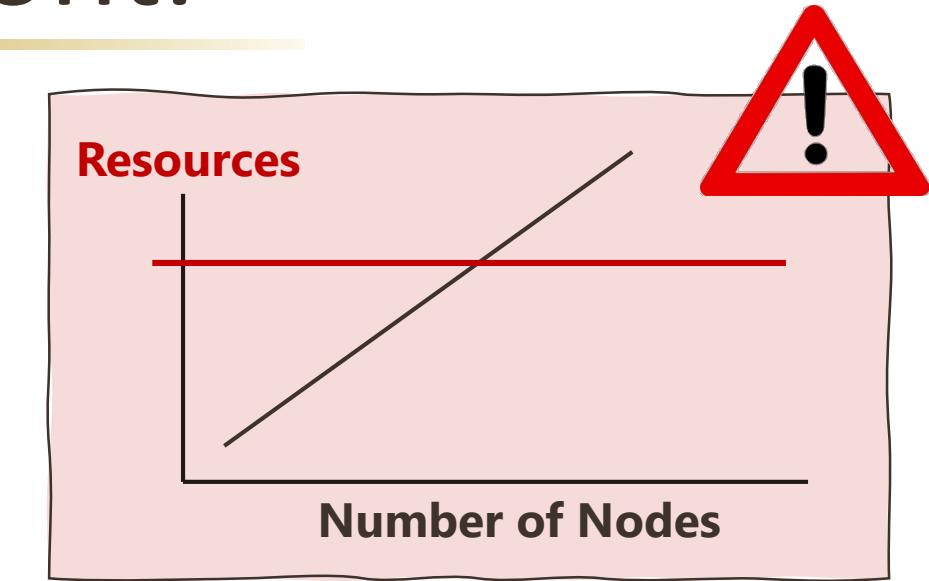


Client-server vs. P2P – cont.

▪ Client-Server architecture

Problems:

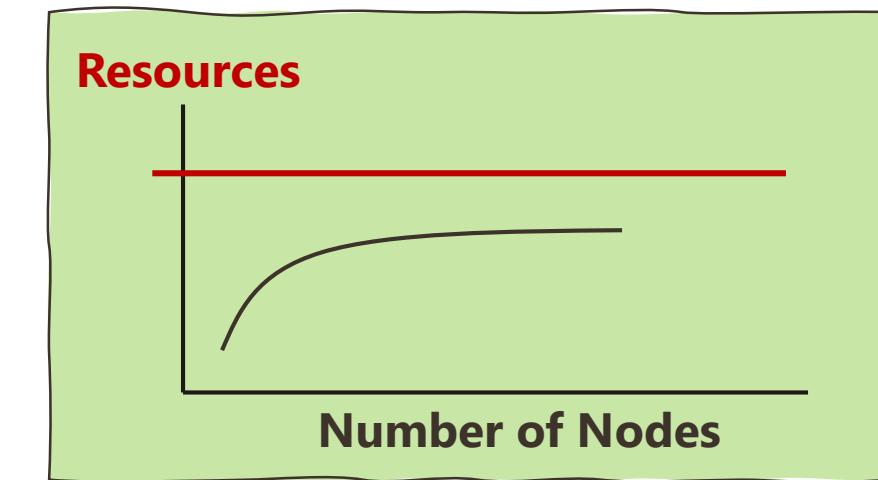
- Limited resources
- All loads are centered on the server
- Server-based architecture has low scalability
- The setup and maintenance cost is high



▪ Peer-to-Peer (P2P) architecture

Advantages:

- Distributing loads to all users
- Users consume and provide resources
- P2P architecture has high scalability
- The setup and maintenance cost is low



P2P Protocols



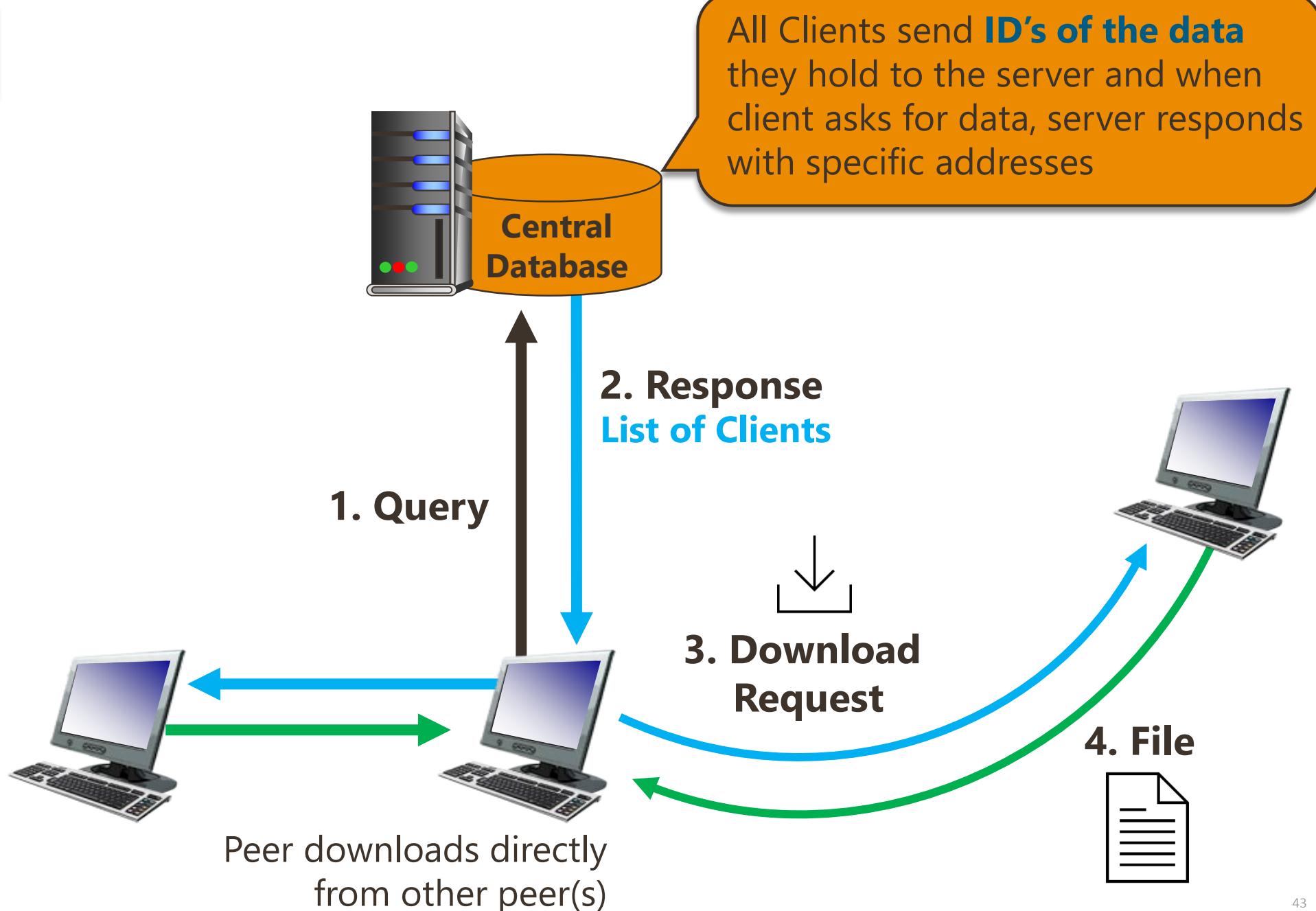
Hybrid
(not pure P2P)



Napster



hybrid system
NOT pure
P2P network



Napster

- **Services**

- Was used primarily for file sharing (audio)
- Chat program, instant messaging service, tracking program,...

- **Centralized system**

- Single point of failure => limited fault tolerance
- Limited scalability (server farms with load balancing)

- **Query is fast** and **upper bound for duration** can be given

Gnutella

- **Pure Peer-To-Peer**

- ✓ very simple protocol
- ✓ no routing "intelligence"

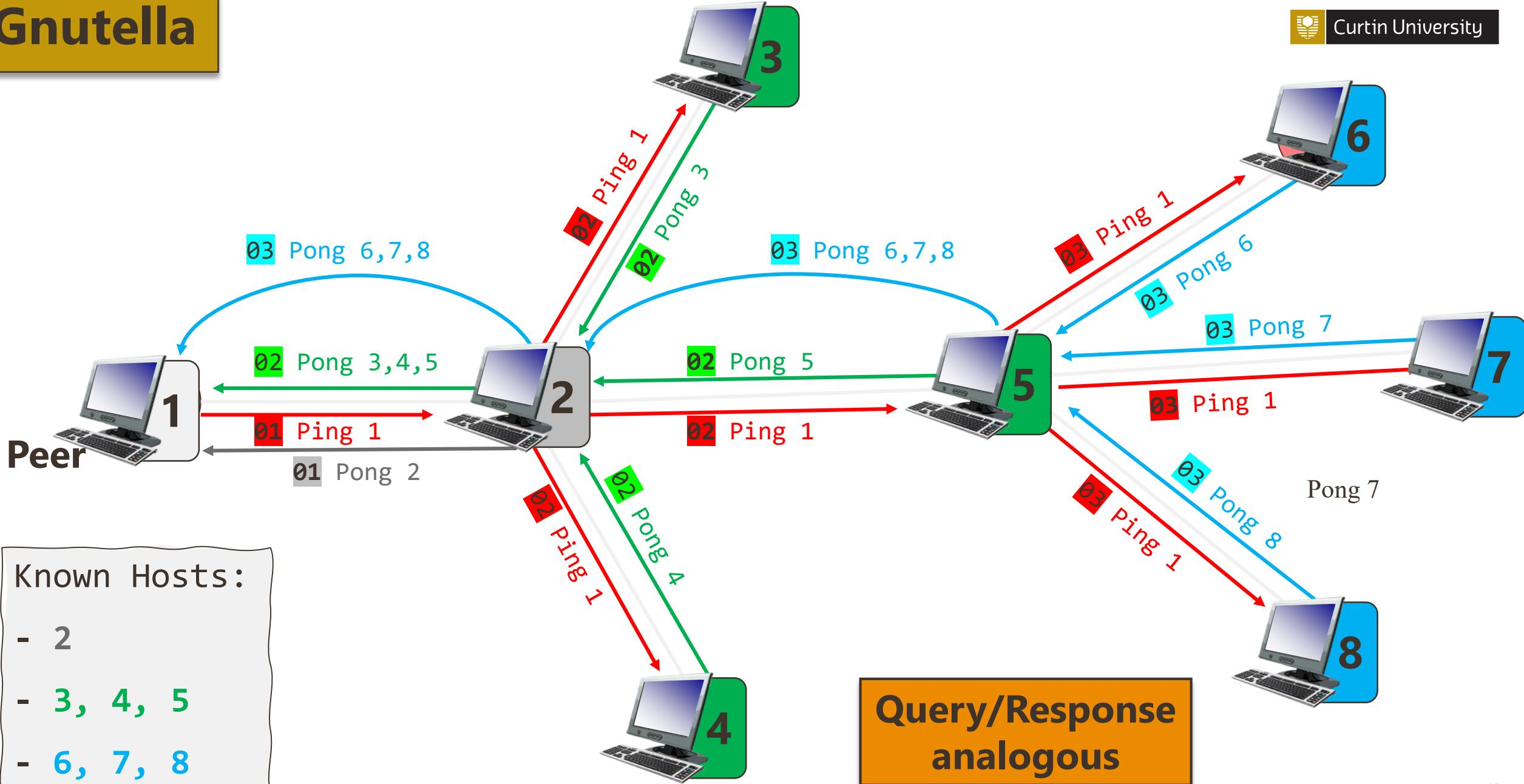
- **Constrained Broadcast**

- ✓ Life-time of packets limited by TTL (typically set to 7)
- ✓ Packets have unique ids to detect loops





Gnutella



Problem - Free riding

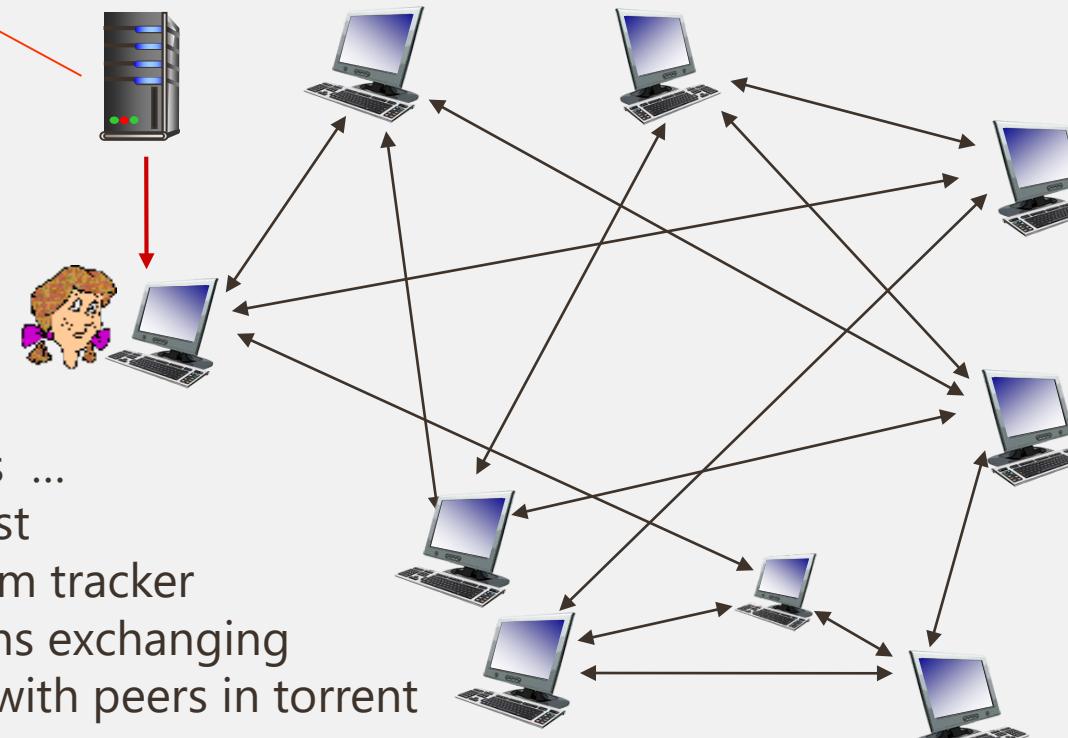
- File sharing networks rely on users sharing data
- Two types of free riding
 1. Downloading but not sharing any data
 2. Not sharing any interesting data
- On Gnutella
 - 15% of users contribute 94% of content
 - 63% of users never responded to a query



BitTorrent: P2P File Distribution

- **File** divided **into** 256Kb **chunks**
- Peers in torrent **send/receive file chunks**

tracker: tracks peers participating in torrent

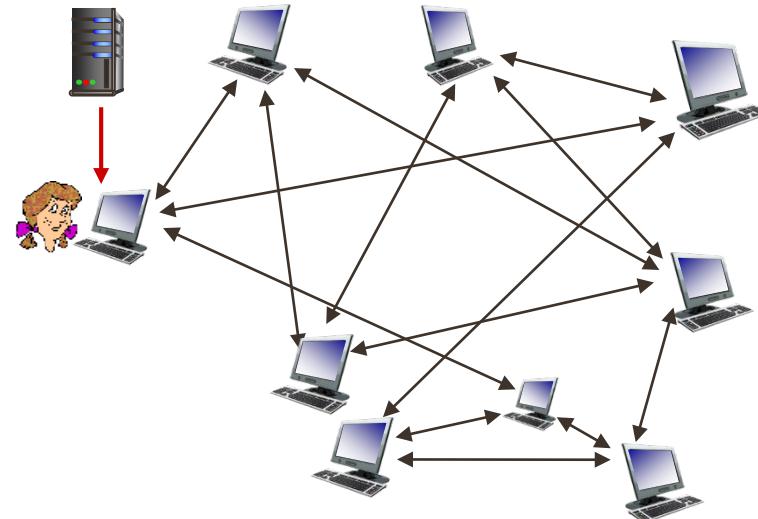


Alice arrives ...
... obtains list
of peers from tracker
... and begins exchanging
file chunks with peers in torrent



torrent: group of peers exchanging chunks of a file

BitTorrent – cont.



▪ Peer joining torrent:

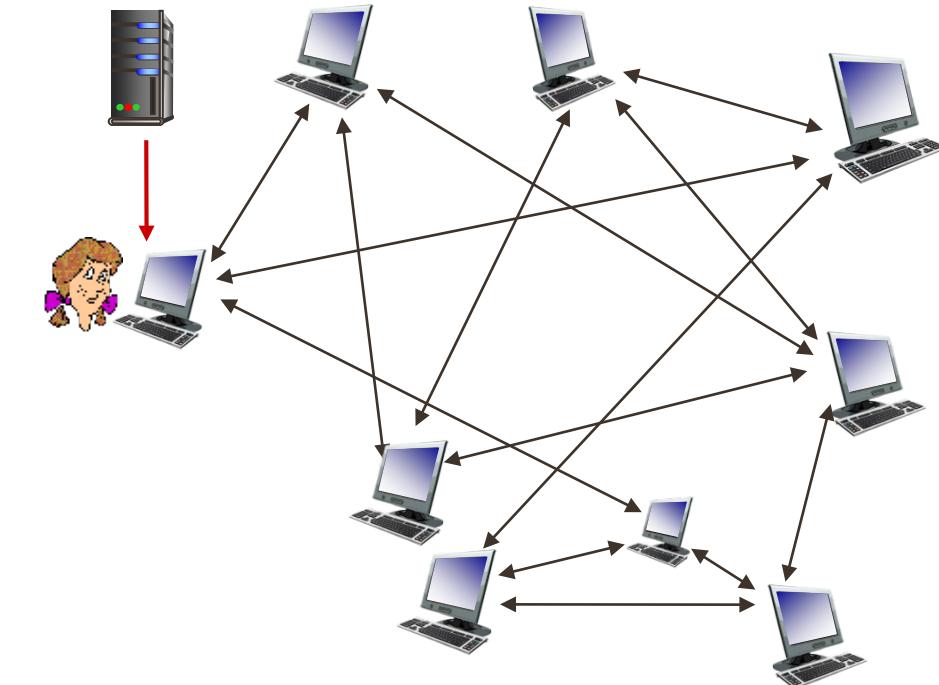
- has no chunks, but will accumulate them over time from other peers
- registers with tracker to get list of peers, connects to subset of peers ("neighbors")

- While downloading, peer uploads **chunks** to other peers
- Peer may change peers with whom it exchanges chunks
- **churn:** peers may come and go
- once peer has entire file, it may (selfishly) leave or (altruistically) remain in torrent



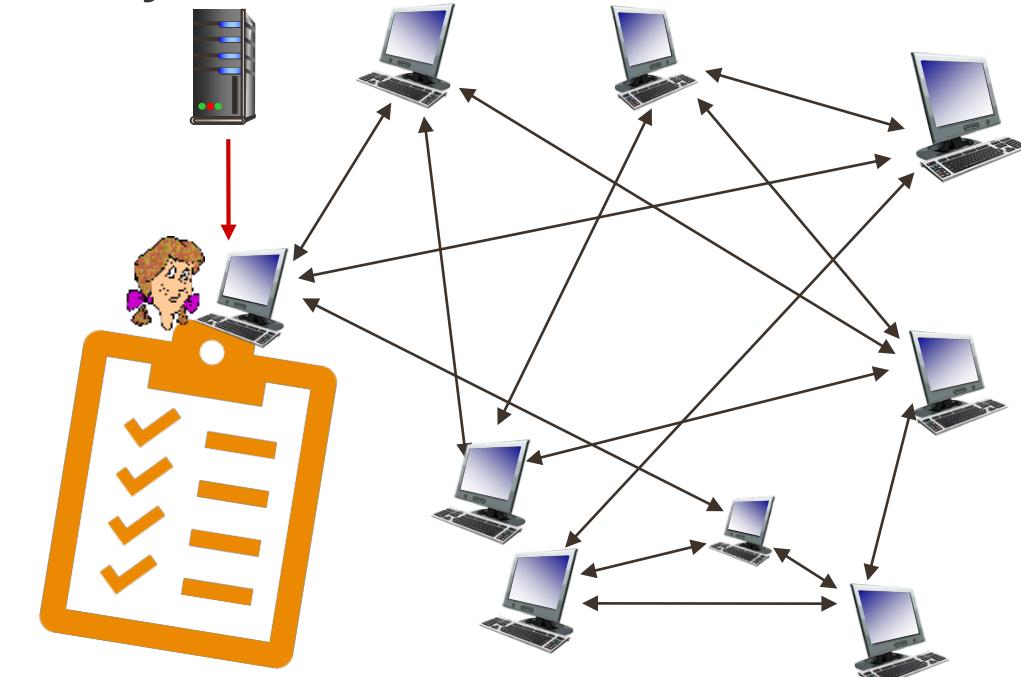
Requesting File Chunks

- at any given time, different peers have different subsets of file chunks
- periodically, Alice asks each peer for list of chunks that they have
- Alice requests missing chunks from peers, rarest first



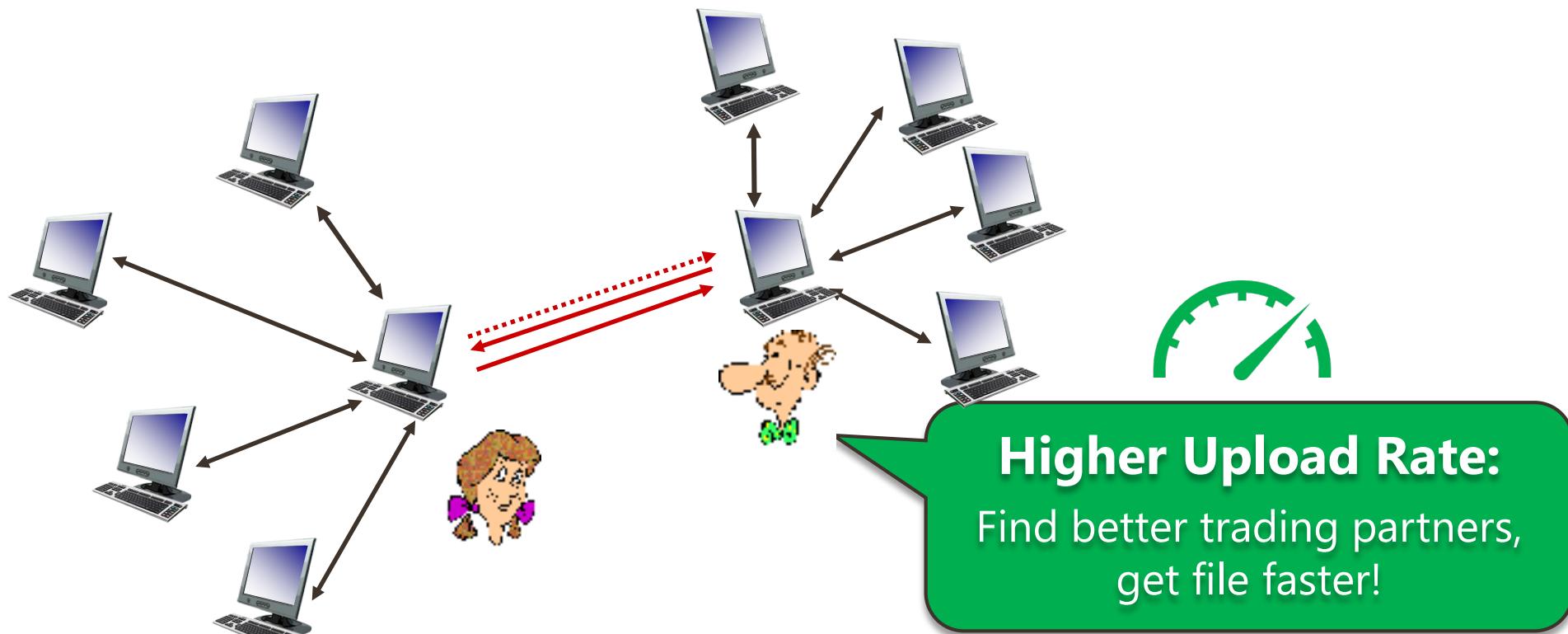
Sending File Chunks (tit-for-tat)

- Alice sends chunks to the **four peers** currently **sending her chunks at highest rate**
 - Other peers are choked by Alice (*do not receive chunks from her*)
 - Re-evaluate top 4 **every 10 secs**
- **Every 30 secs:** randomly select another peer, starts sending chunks
 - “**optimistically unchoke**” this peer
 - newly chosen peer may join top 4



BitTorrent: tit-for-tat

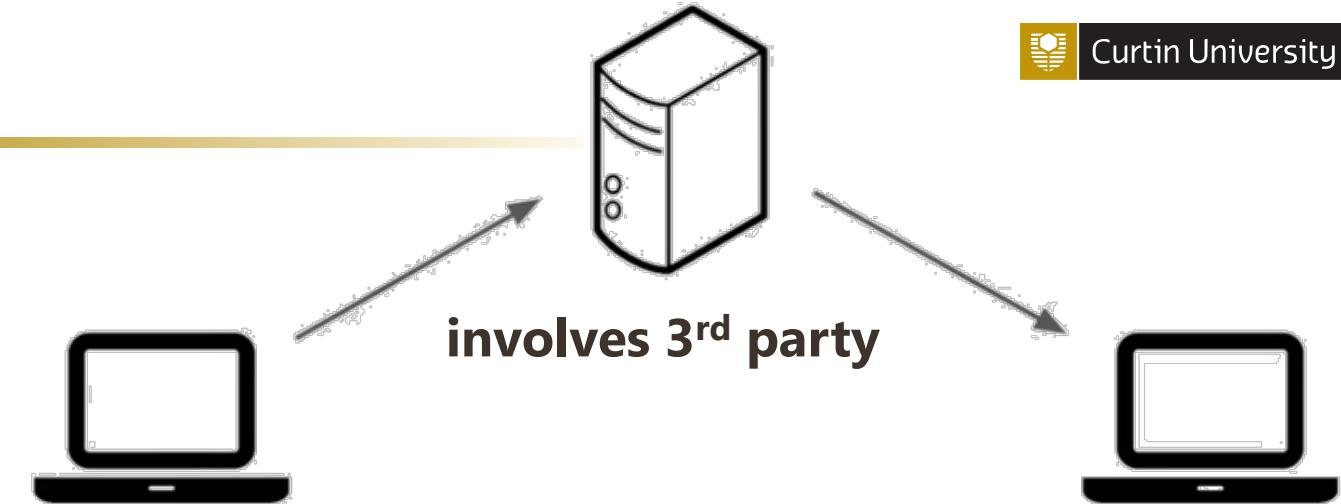
1. Alice “optimistically unchoke” Bob
2. Alice becomes one of **Bob’s top-four providers**; Bob reciprocates
3. Bob becomes one of **Alice’s top-four providers**



P2P Web

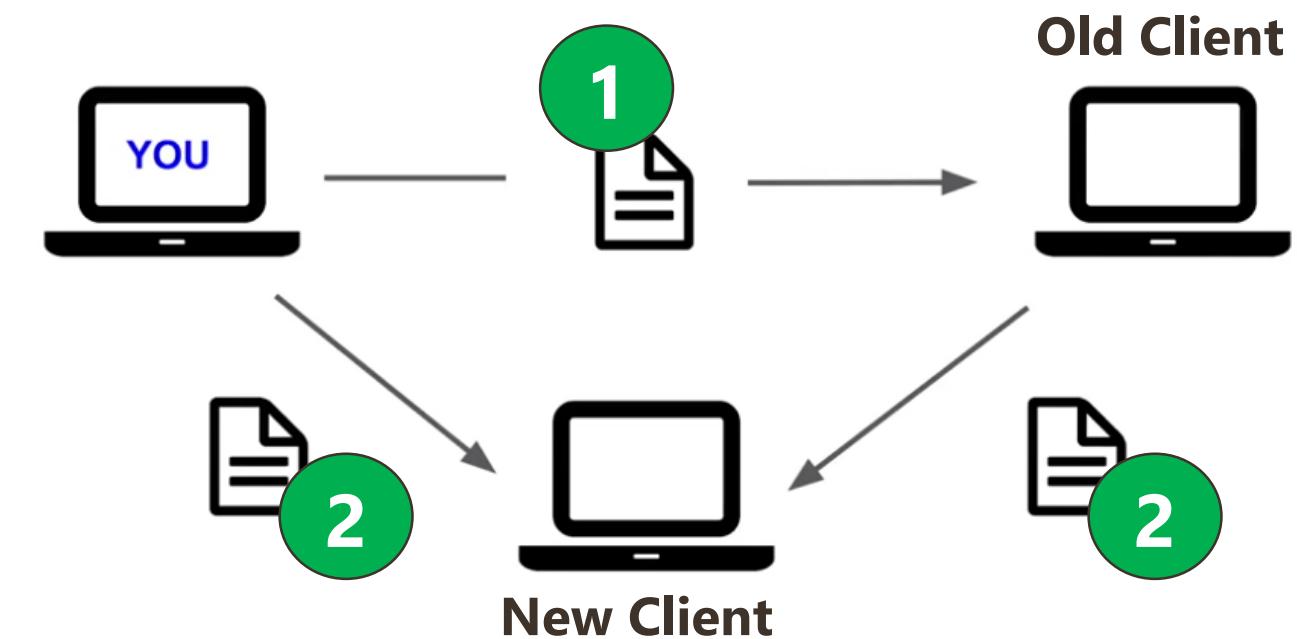
▪ Client-Server Model

- Lack of privacy, not secure !
 - *Communication is not really end-to-end*
- Lack of data ownership



▪ P2P Web

- Improve privacy
- Give users ownership
- Build on top of BitTorrent Protocol
- Files distributed P2P
- **Visiting users contribute bandwidth**
- Anyone can publish from their device
(can become a server ad-hoc fashion)





Digital Encryption

- Fundamentals
- Symmetric Key Encryption
- Asymmetric Key Encryption
- Digital Signature
- SSL Certificate

What is Encryption?

- Encryption is a process that **encodes a message** or file so that it can be only be read by certain people.
- Encryption uses an **algorithm to scramble, or encrypt, data** and then **uses a key for the receiving party to unscramble, or decrypt, the information**

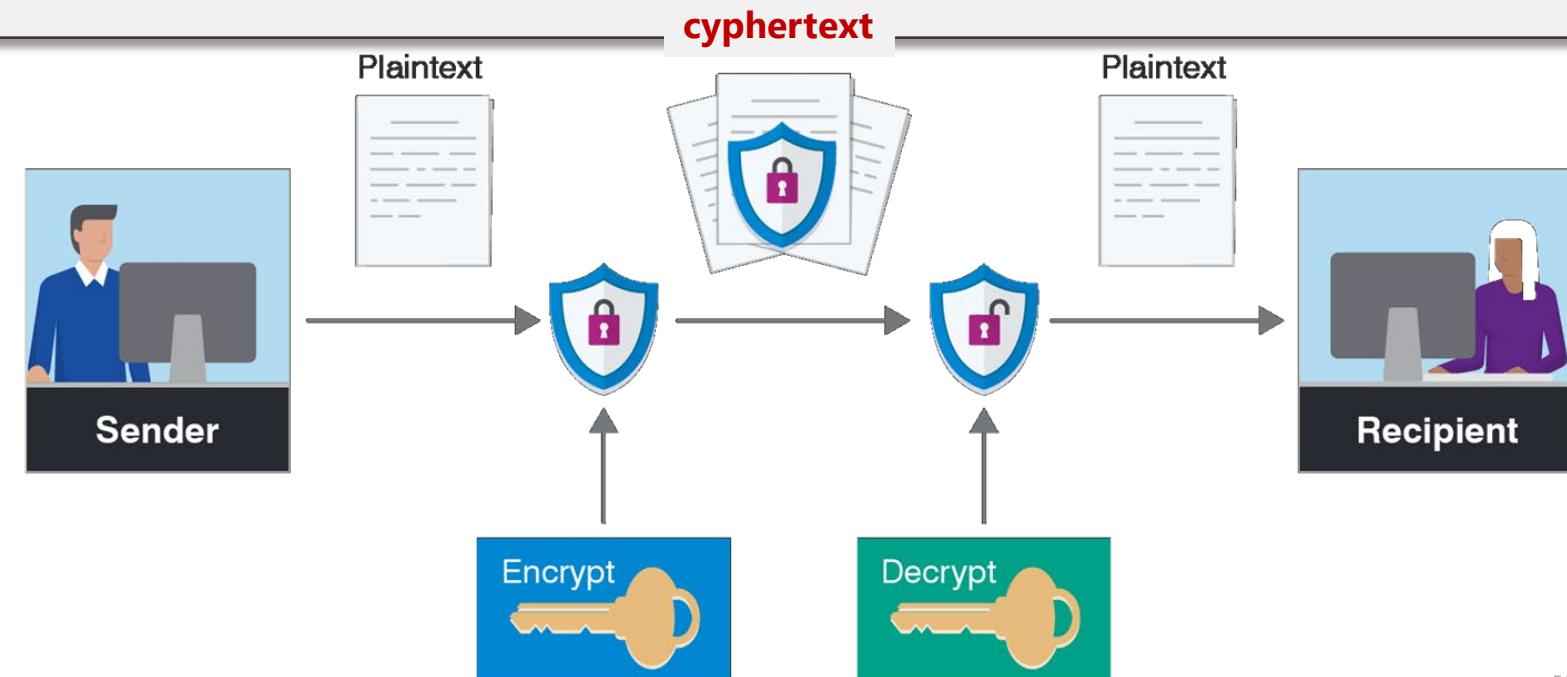
- Types:
 - **Symmetric Key Encryption** (Single Key)
 - **Asymmetric Key Encryption** (Public/Private Key)



Symmetric Key Encryption

- **Only one key** (a secret key) **is used** to both encrypt and decrypt information

- **Encrypt (Message, SymKey) = Cyphertext**
- **Decrypt (Cyphertext, SymKey) = Message**



Asymmetric Key Encryption

- Each person has a **key-pair (public key - pk, private key - sk)**

1. Secrecy:

- **pk** encrypts **sk** decrypt

2. Authenticity:

- **sk** can encrypt **pk** decrypt

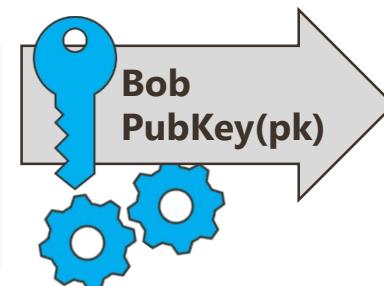
Kept at certification authorities
Some integrated to browsers

Kept at person (secret)

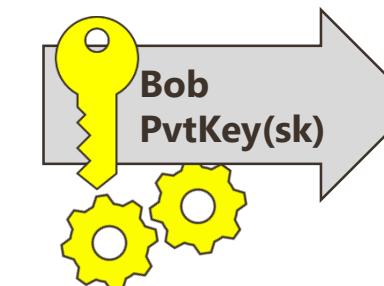


message

Easter
Bunny is
Real.



^43&(&#^@sd
a342356asd32ey
h21&8)#2139%9
0*(0348kshe48^

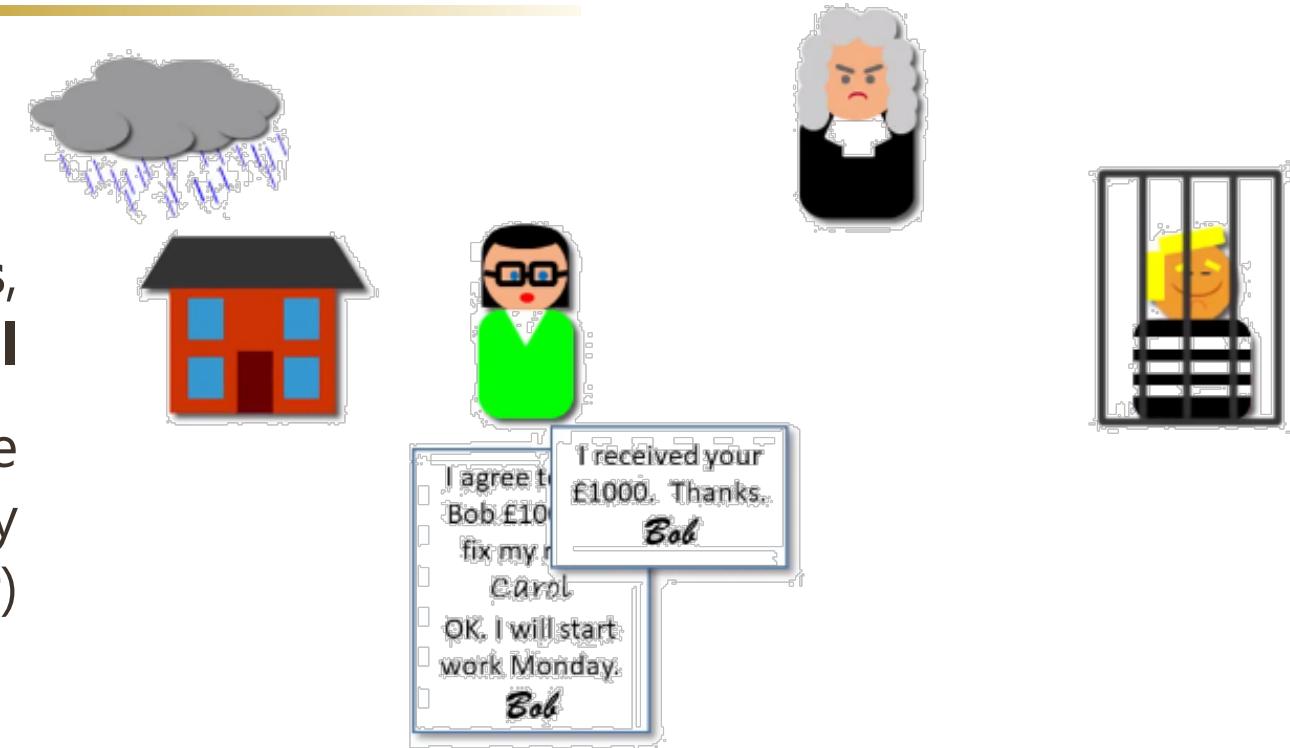


message

Easter
Bunny is
Real.

Digital Signature

- Same as Handwritten signatures, but digital
 - To ensure the authenticity of the sender (i.e. a document signed by the sender)



▪ Asymmetric Key Encryption

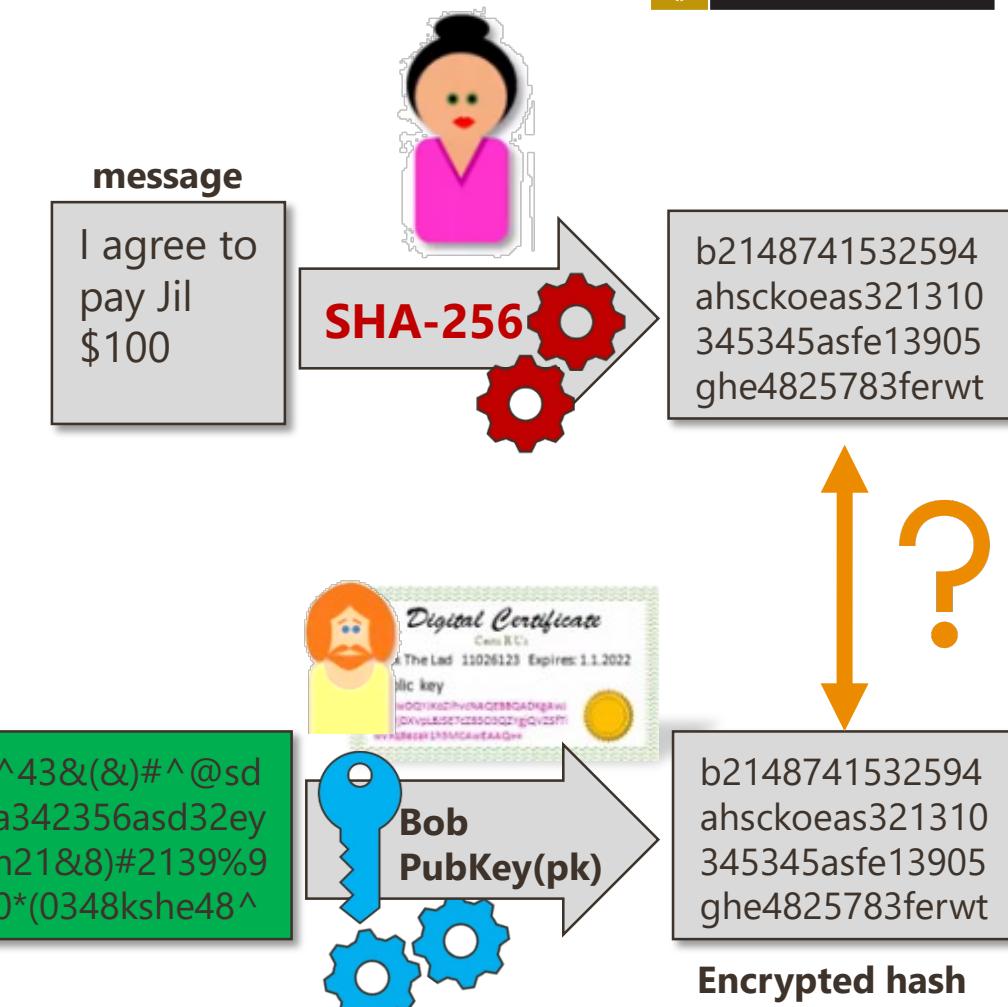
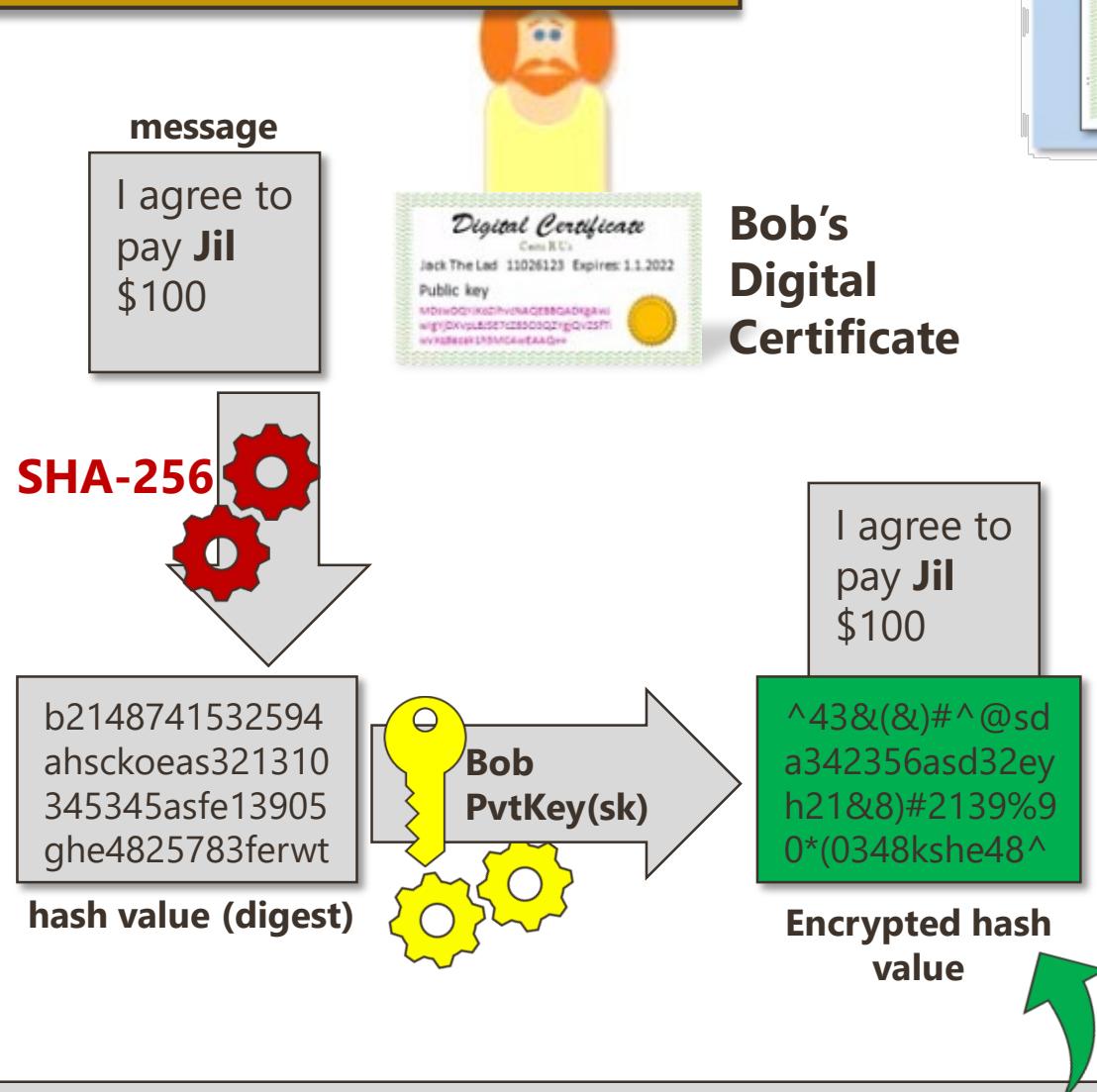
- ✓ Any key can be used to encrypt or decrypt
- ✓ This make Digital Signatures possible

▪ Hashing

- ✓ Generate a hash value
- ✓ 1-way process (not reversible)
- ✓ i.e. SHA-128, SHA-256



Digital Signature



Sign(Message, sk_{bob}) = Signature

Verify(Message, Signature, pk_{bob}) = T/F

SSL Certificate



Web Server's Digital Certificate

- Verified Identity with the Public Key



Client



Send the shared (symmetric) key

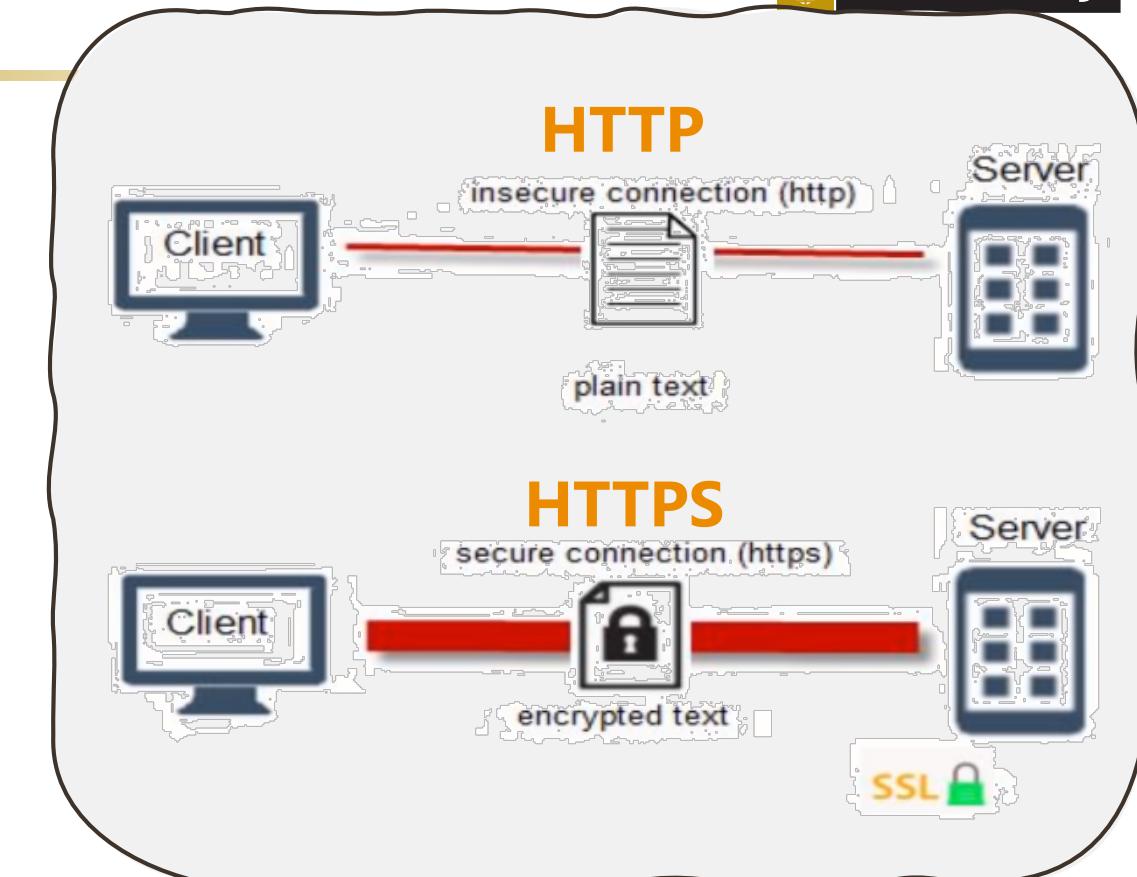
encrypted with Web server's pub key



Server



Future Communication is Encrypt with shared key





■ App Protocols – DHCP

- 4-way Handshake
 - DHCP Discover
 - DHCP Offer
 - DHCP Request
 - DHCP Ack

■ Domain Name System (DNS)

- hosts File
- Main Elements
 - Domain Name Space
 - Name Server
 - Resolvers
- DNS Database
- DNS Registrar

■ Web Search Engines

- Fundamentals
- Ranking Algorithm
- Search Results

■ Web Services

- REST API
- URI / URL / URN
- REST API Highlights

■ P2P

- Fundamentals
- Napster
- Gnutella
- BitTorrent – *in depth*
- P2P Web

■ Digital Encryption

- Fundamentals
- Symmetric Key Encryption
- Asymmetric Key Encryption
- Digital Signature
- SSL Certificate

THANK YOU

Make tomorrow better.

Emerging Networking Technologies

Prof. Ling Li | Dr. Nadith Pathirage | Lecture 11

Semester 1, 2021

Emerging Networking Technologies

- Serverless Computing
- Edge Computing
- Internet Of Things (IoT)
- Software Defined Networking
- Blockchain
 - Bitcoin
 - Other application examples

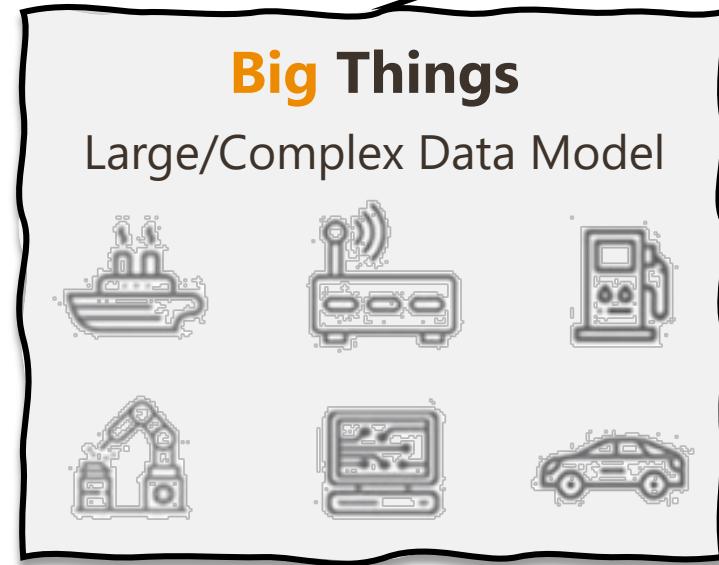


Internet of Things (IoT)

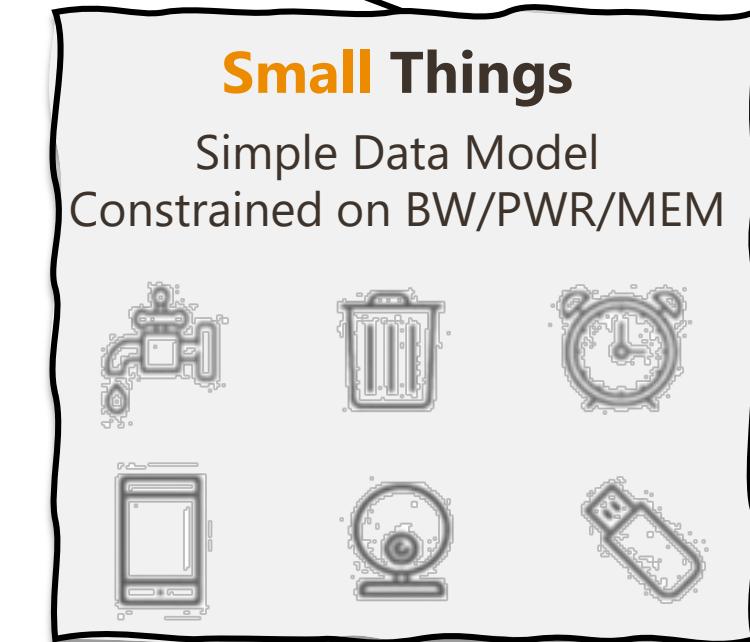
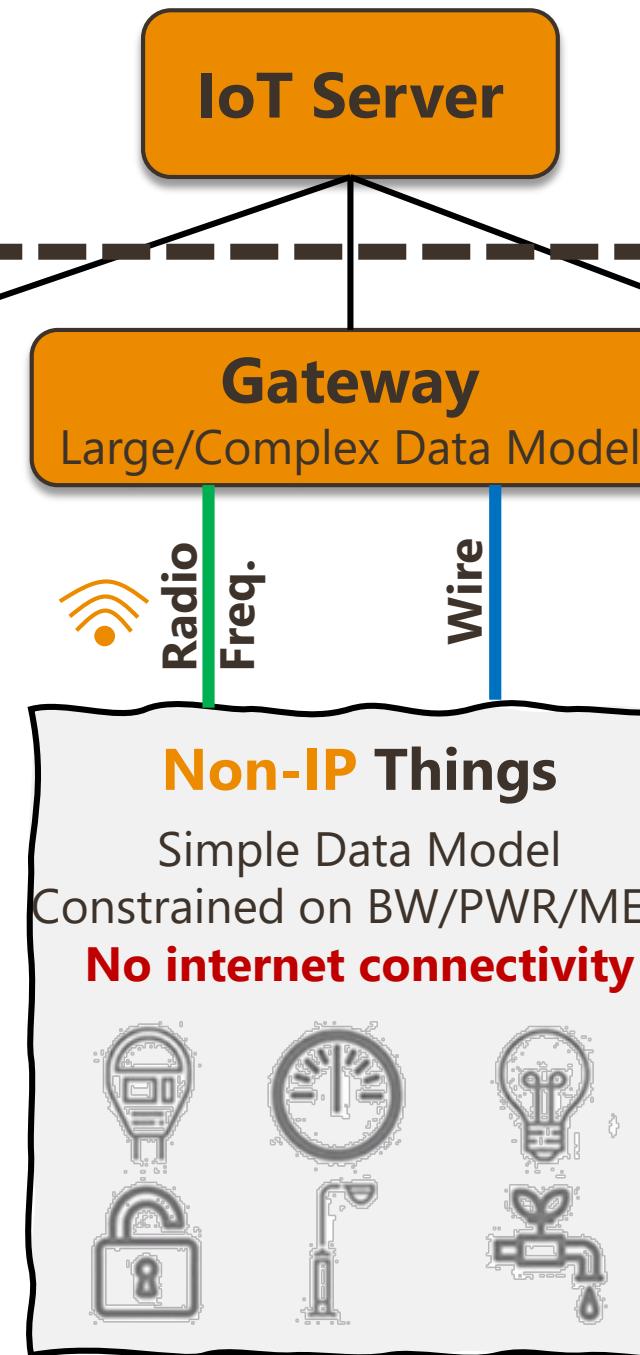
- The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.
- These devices range from ordinary household objects to sophisticated industrial tools.



Internet of what Things



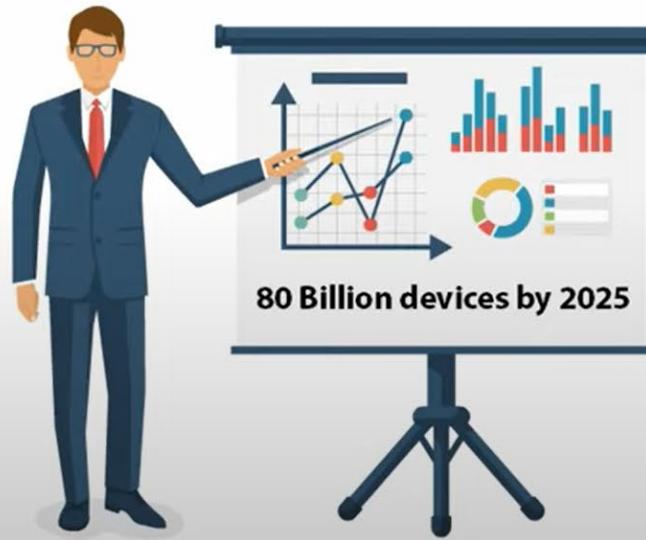
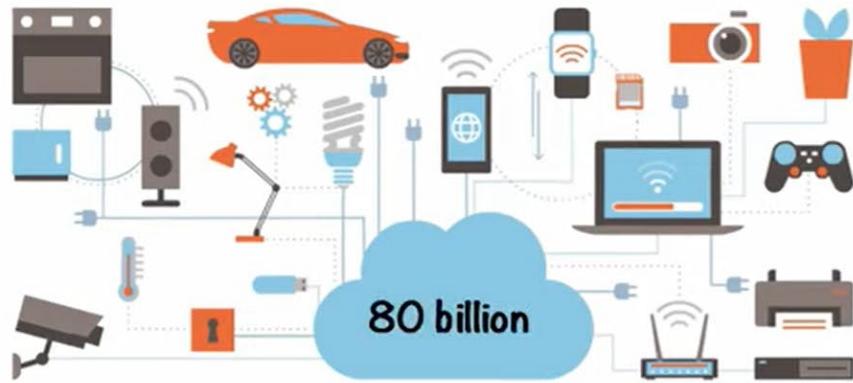
- Big in complexity
- No problem with power
- Always connected
- Thousands of parameters to control



- Most connected with sim card
- Operate on battery
- Less parameters to control

In 2025:

20.4 billion
by the end
of 2020



Area	IoT Applications
Consumer Applications	Smart home technology, health and fitness apps, smart appliances, wearable tech
Medicine	Emergency alert systems, smart devices like hearing aids, smart bed management, remote health monitoring
Agriculture	Environmental sensors for farmland information
Manufacturing	Smart control of manufacturing systems, plant optimization
Energy	Remote control of heating systems, smart grid for balance energy usage
Infrastructure Management	Monitoring traffic, wind farms, railway track and bridges
Environmental protection	Checking pollution levels, soil health, and earthquake early-warning sensors



Software Defined Networks

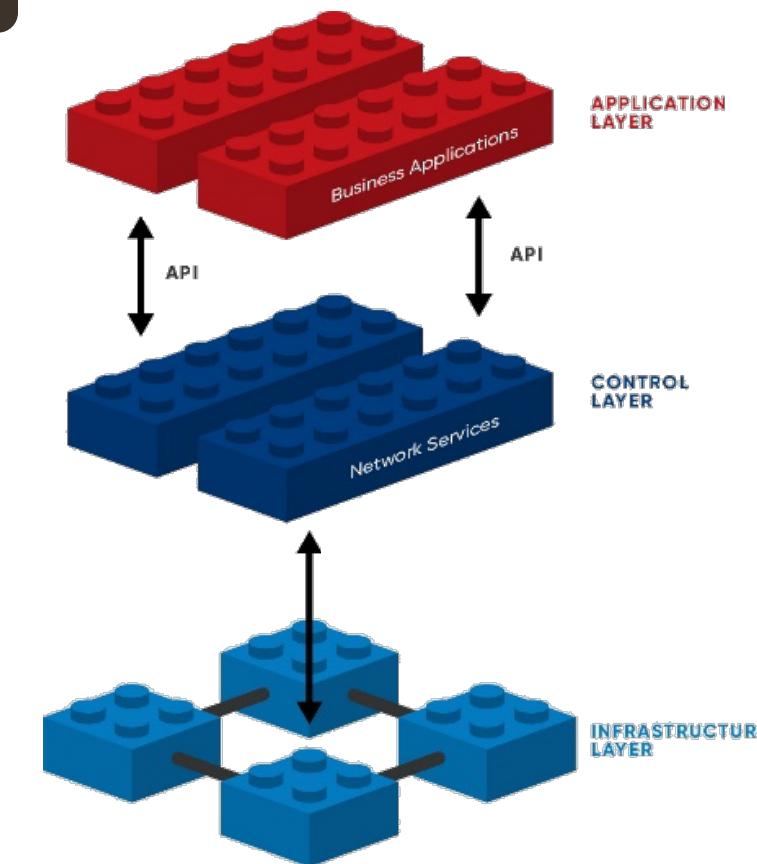
- Fundamentals
- SDN Model
 - SDN Controller
- Traditional Networks vs SDNs
- SDN Benefits

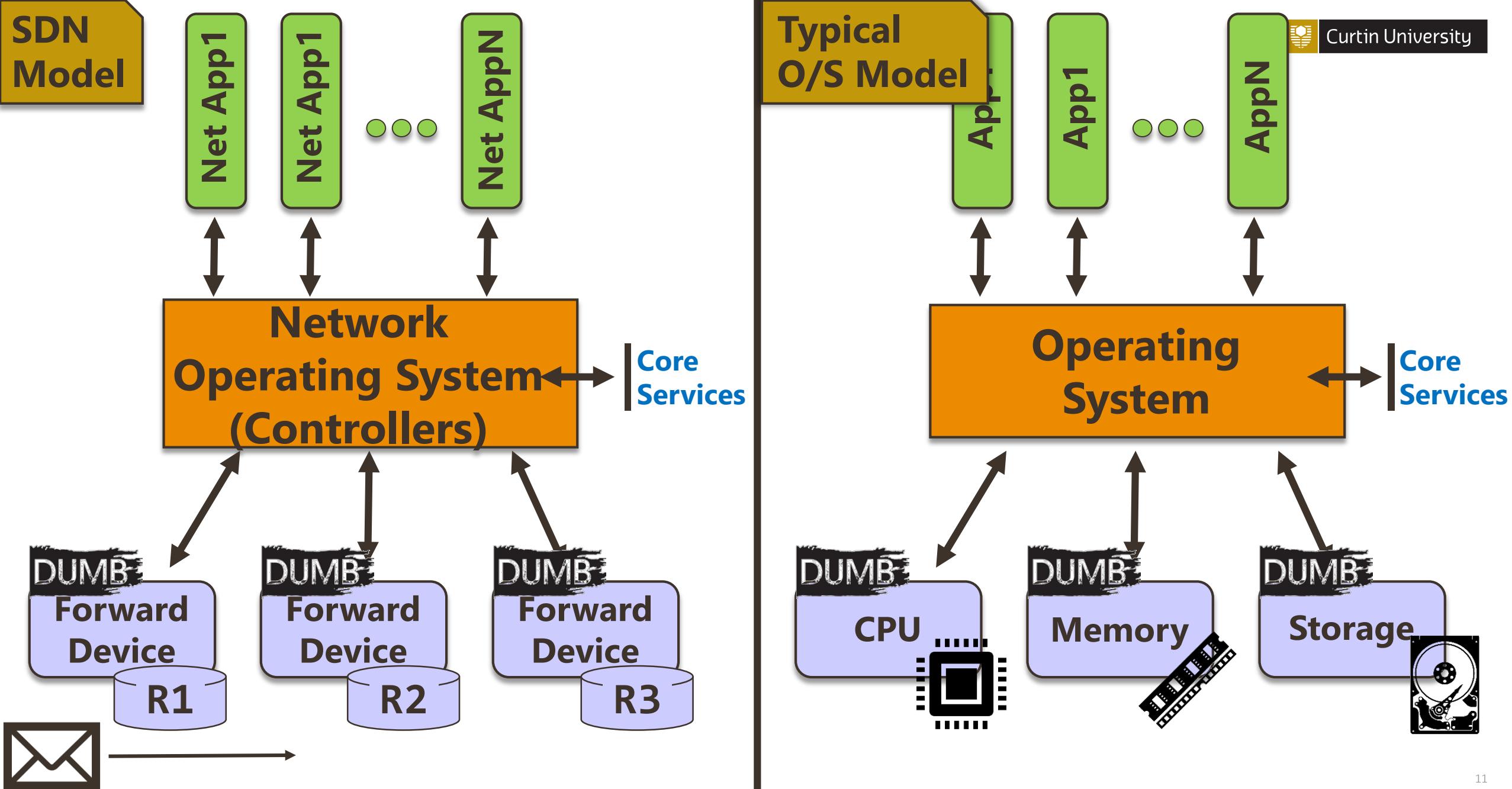
Software Defined Networks (SDN)



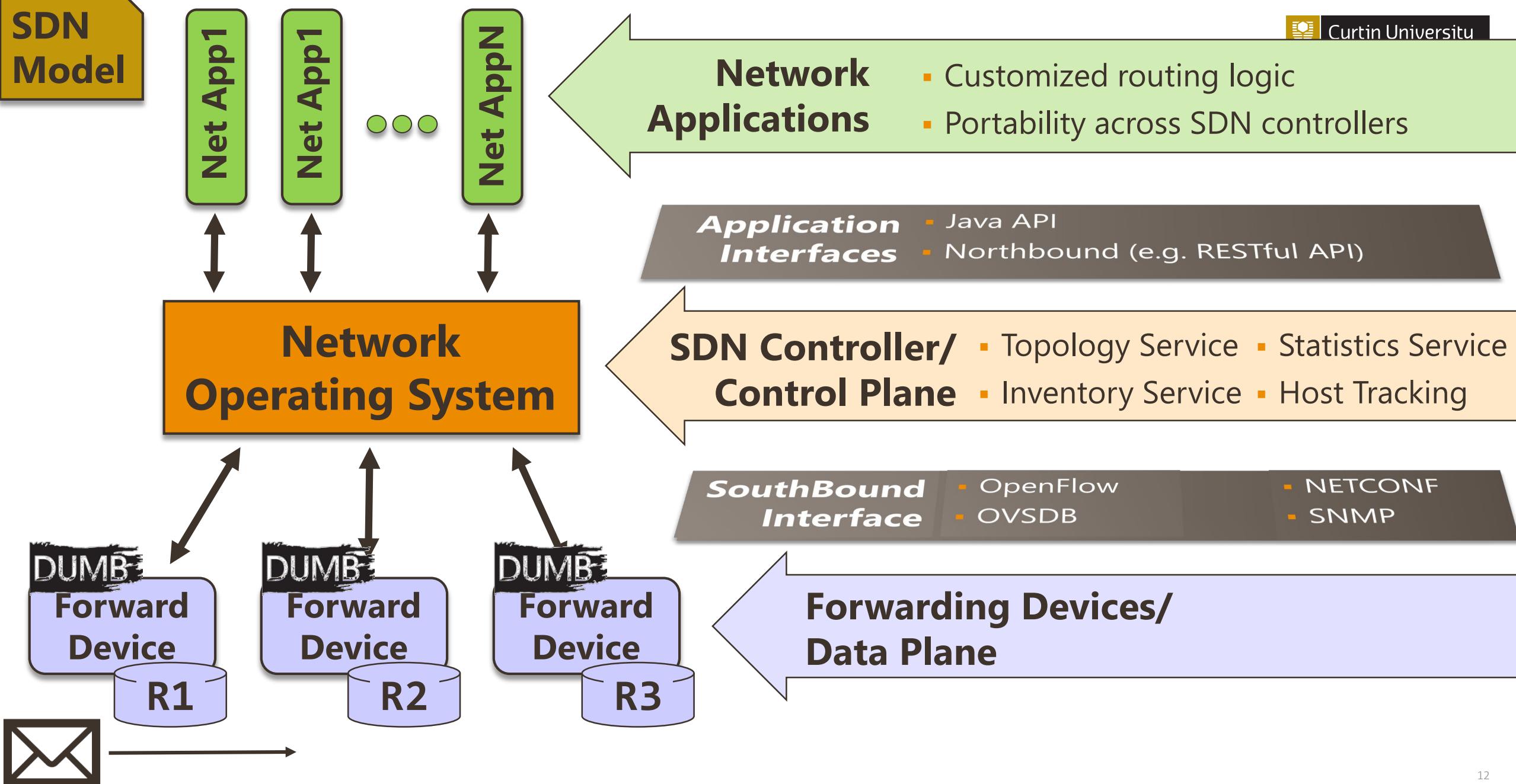
Goal: Network to be dynamic and programmable

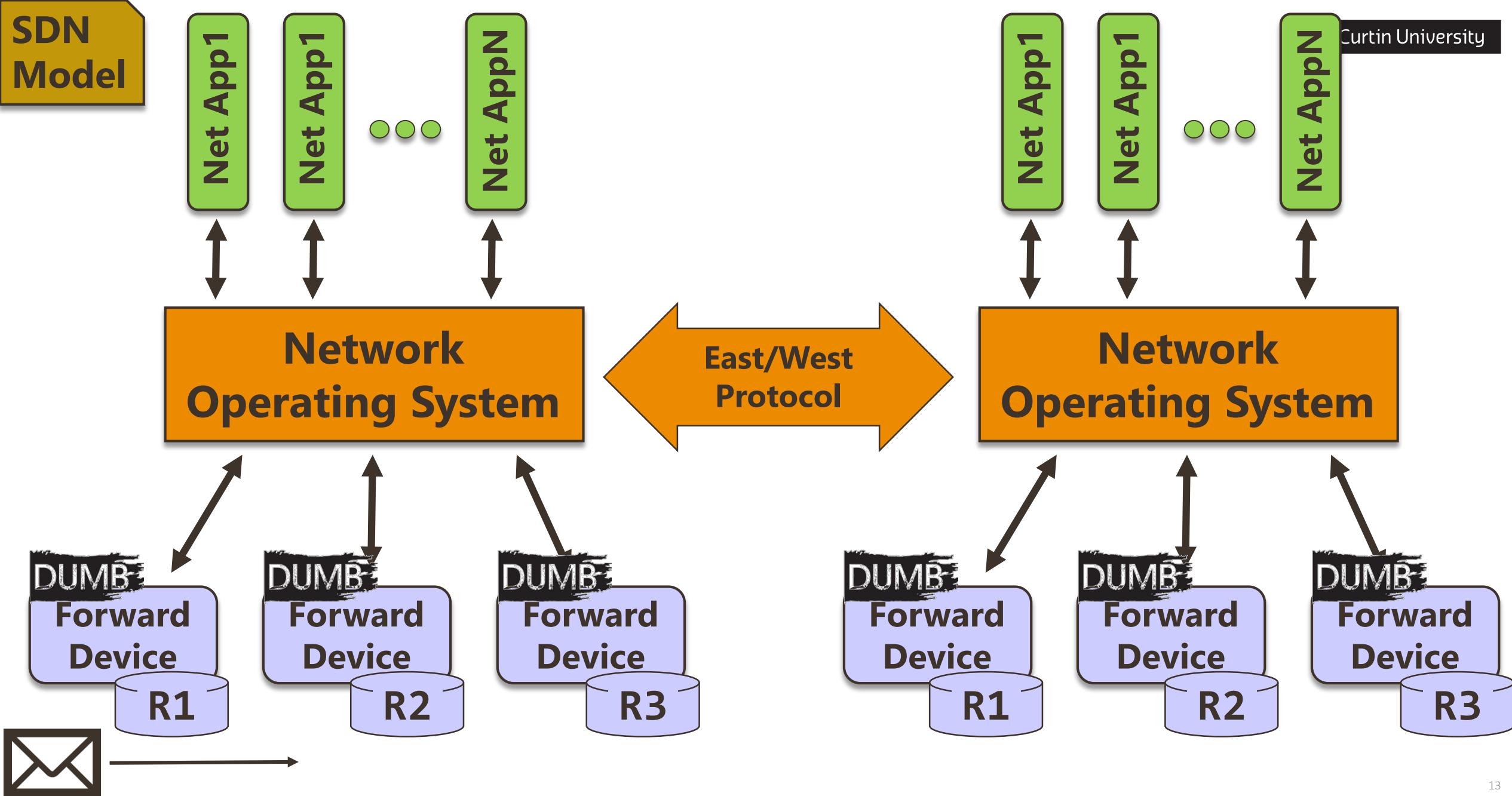
- Achieves agility, flexibility, and scalability that mobility, cloud, and IoT demand
- SDN attempts to centralize network intelligence in one network component by disassociating the forwarding process of network packets (**data plane**) from the routing process (**control plane**)
- Enabling the network control to become directly programmable and the underlying infrastructure to be abstracted from applications and network services





SDN Model



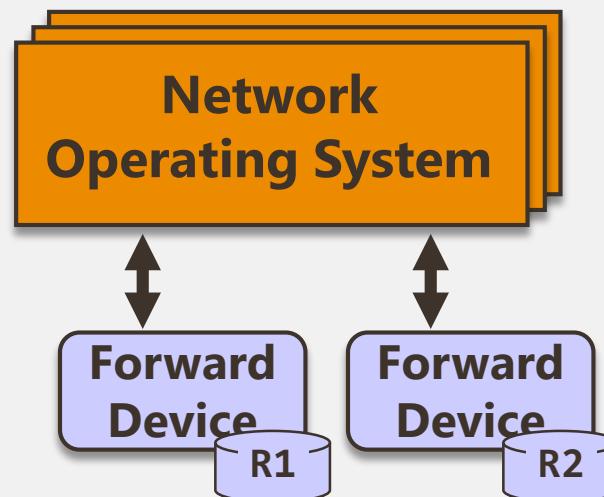


SDN Controller (a.k.a Network O/S)

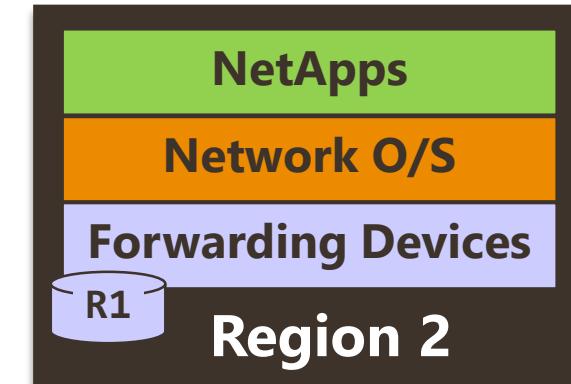
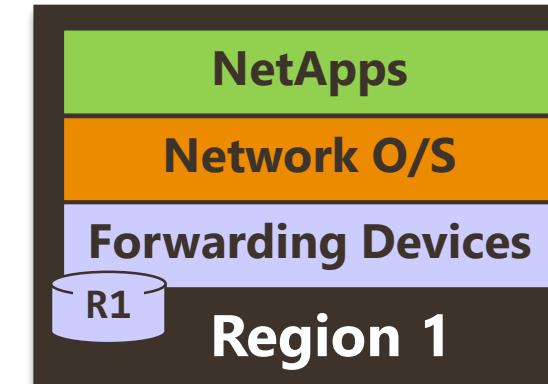
- Logically Centralized Network O/S has a global view of all forwarding devices below it.
- Provides a programming interface to the network applications

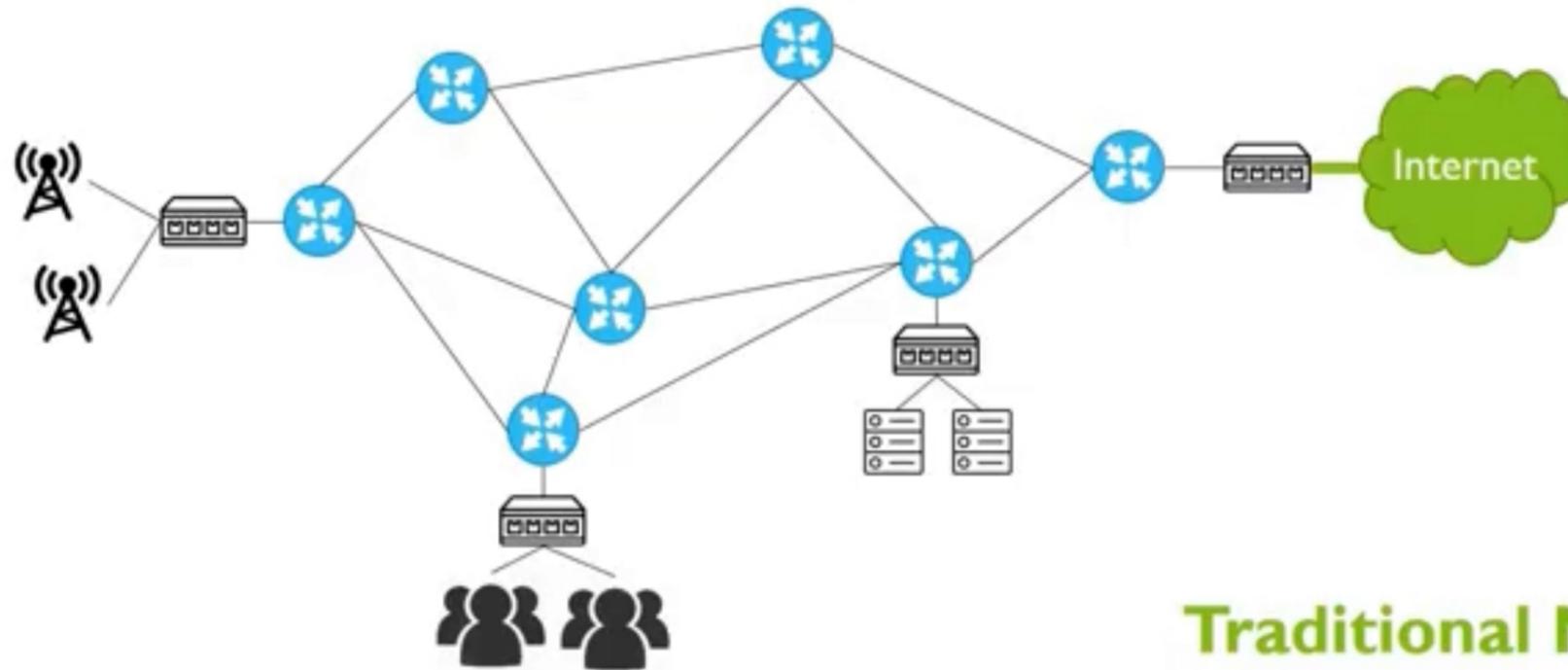
**Network
Operating System**

- Logically centralized
 - Physically can be in a **cluster**



Regional
SDN Controllers





Traditional Network

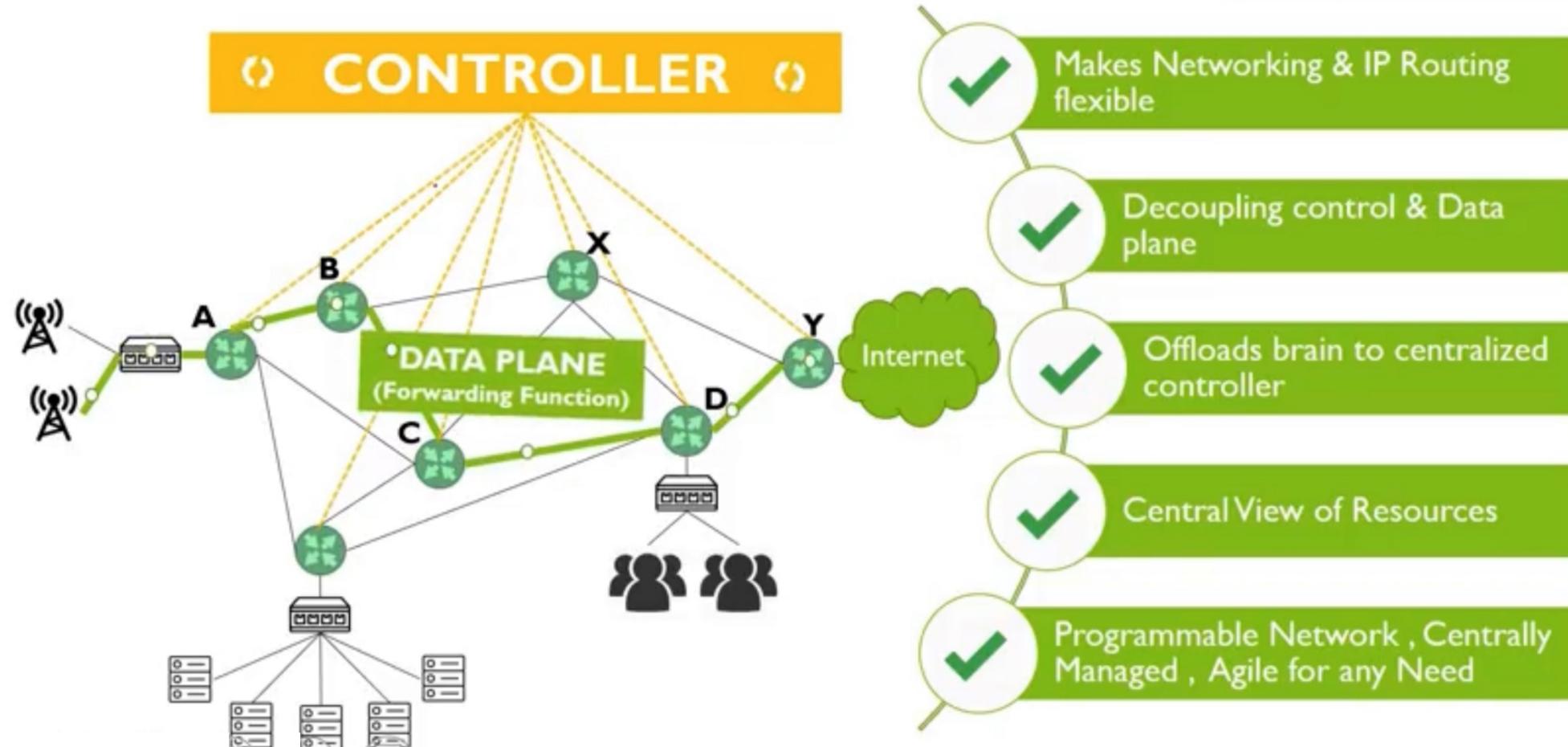
Use of Integrated Hardware & Software



- Data or Forwarding Plane
- Control Plane
- Management Plane

SDN : Separation of Control & Data layer

Features of SDN



Why Is SDN a Big Deal ?

- **Greater speed and faster delivery of services:**

- Configuring hundreds of routers can be done all at once

- **Accuracy and reliability:**

- The human error that comes with traditional manual methods can be vastly reduced with a validated network application that operates with total consistency

- **Simplicity:**

- The SDN controller manages complex rules and policies behind the scenes so that you can focus on the higher-level aspects of what you want to do

Why Is SDN a Big Deal ?

- **Ability to optimize the network:**

- Programmable networks allow you to adjust to changes automatically for optimal use of resources and maximum efficiency and speed.

- **Dynamic prioritizing of traffic:**

- Shape traffic depending upon current need
 - One interface allows you to configure all network equipment at one place
 - "I want this protocol to have maximum priority right now!"

- **Better analytics:**

- Deeper data and faster insights, plus improves security visibility



Blockchain

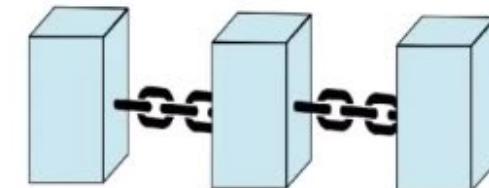
- Fundamentals
- Elements
 - Block
 - Genesis-block
- Block Tampering
- Public and Private Blockchains

Blockchain

- Blockchain is a **distributed immutable ledger**
- Allows to **track anything tangible and intangible**
 - i.e. Bitcoin transactions, blockchain in logistics



collection of records



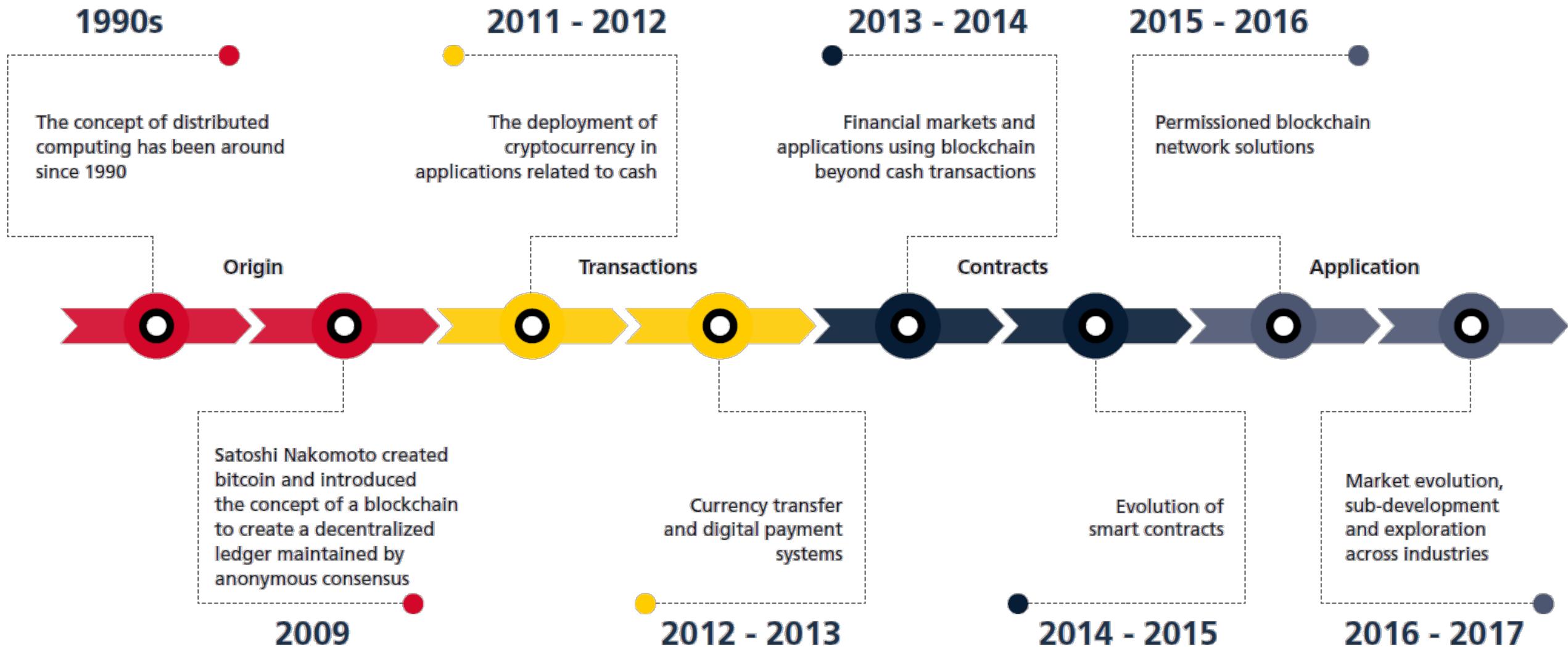
linked with each other



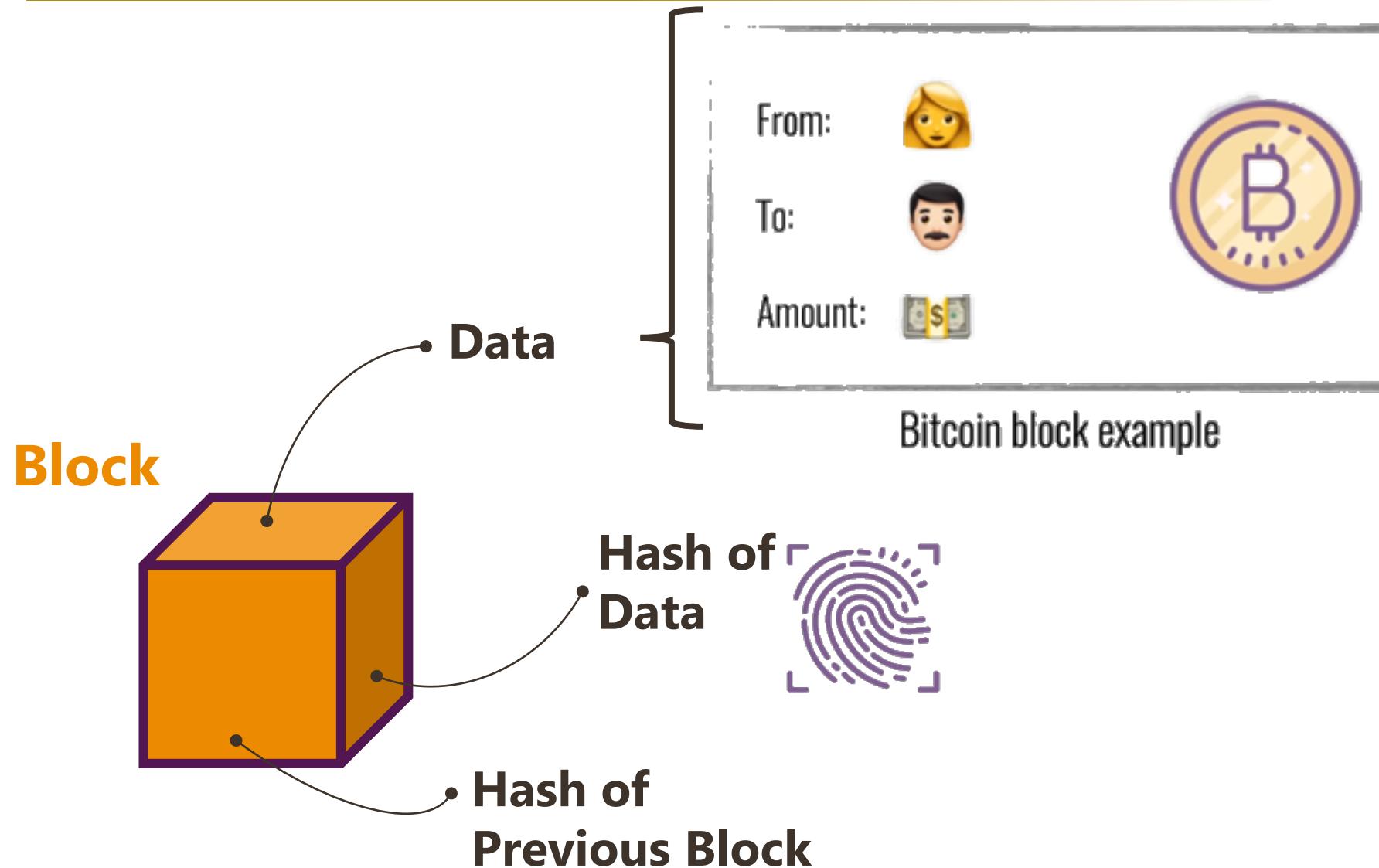
strongly resistant
to alteration



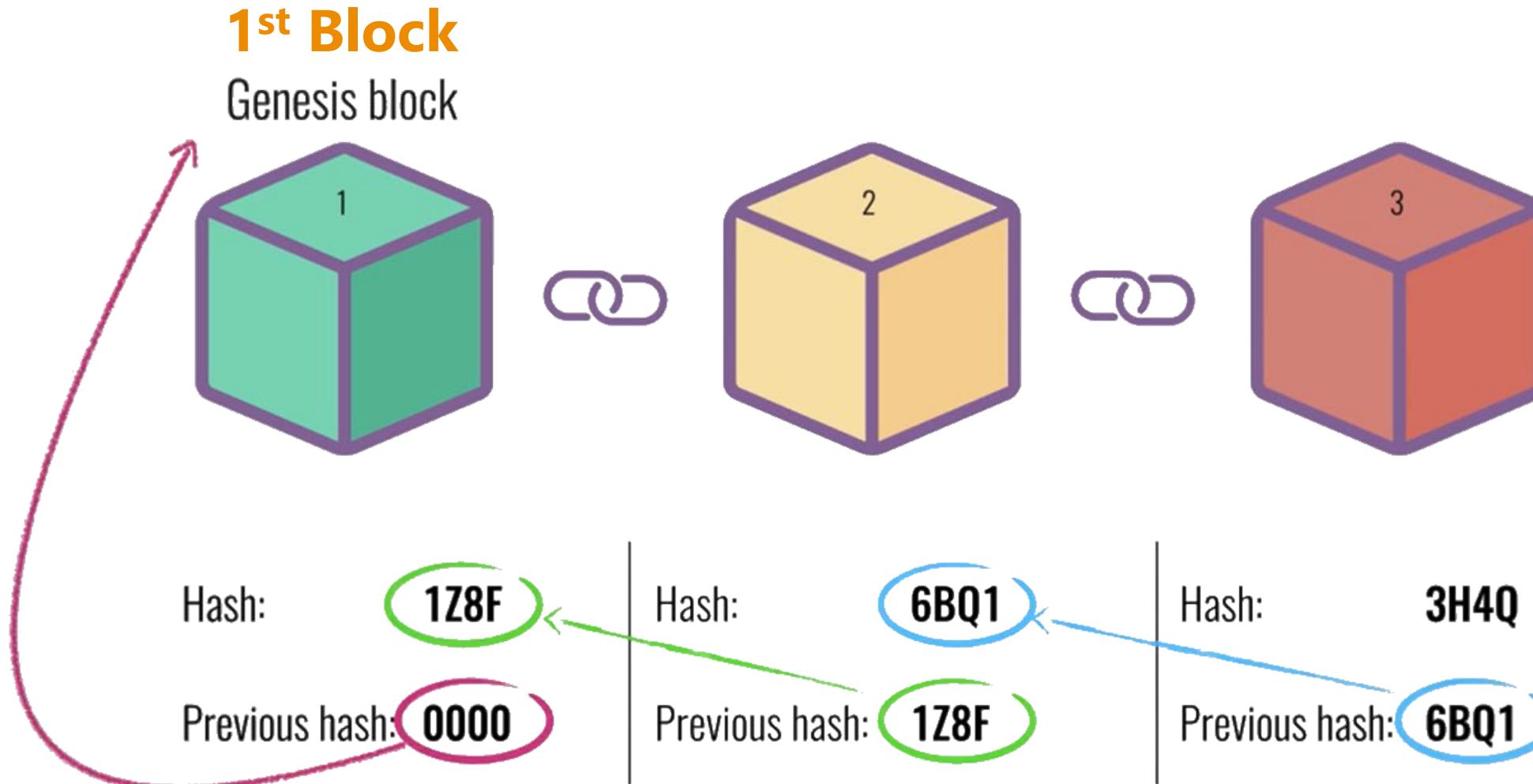
protected using
cryptography



Blockchain – Cont.

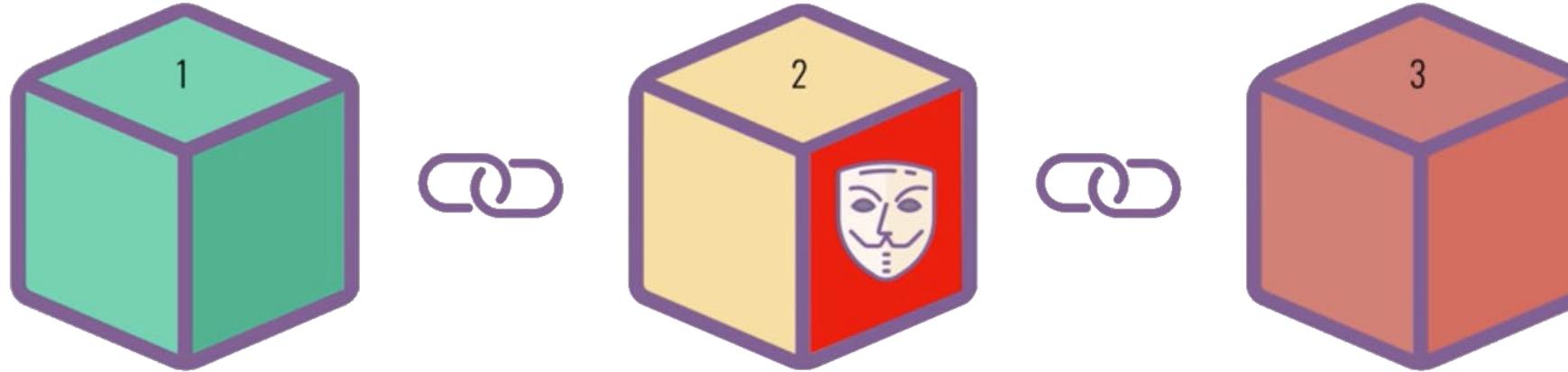


Chained Blocks



Block Tampering

This invalidates all the blocks ahead!



Hash: **1Z8F**

Previous hash: **0000**

Hash: ~~6BQ1~~ **H62Y**

Previous hash: **1Z8F**

Hash: **3H4Q**

Previous hash: **6BQ1**

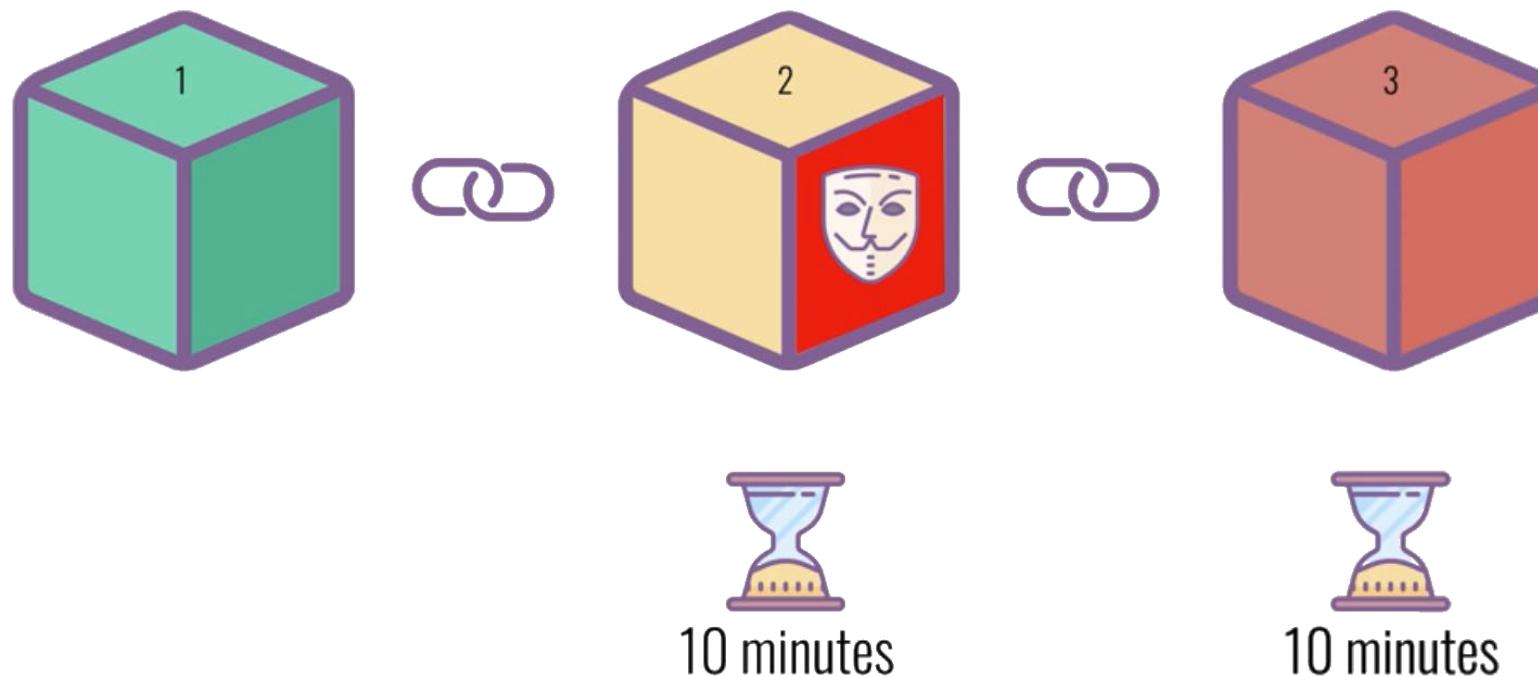
Uh that's
not right??

Block Tampering – cont.

Problem: Can recalculate all hashes of affected blocks

Solution 1: Proof Of Work (PoW)

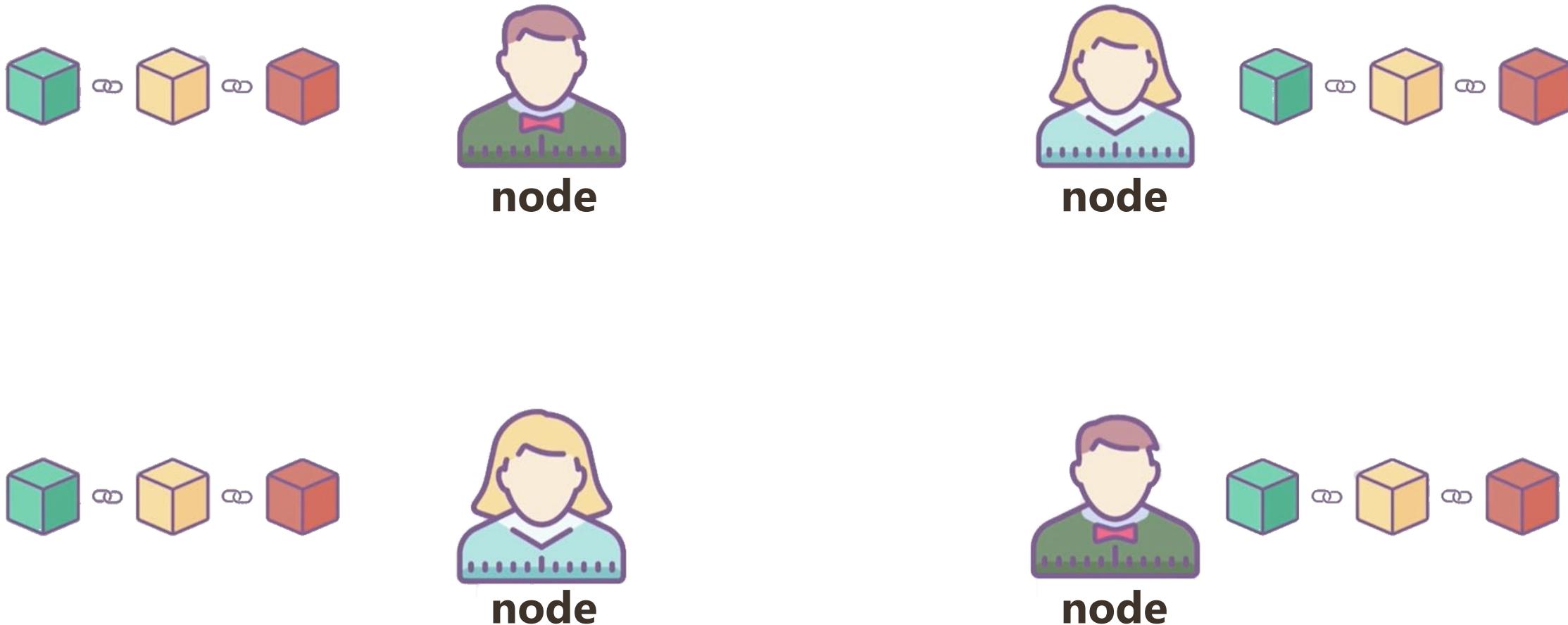
Mechanism to increase difficulty and slow down the creation of new blocks



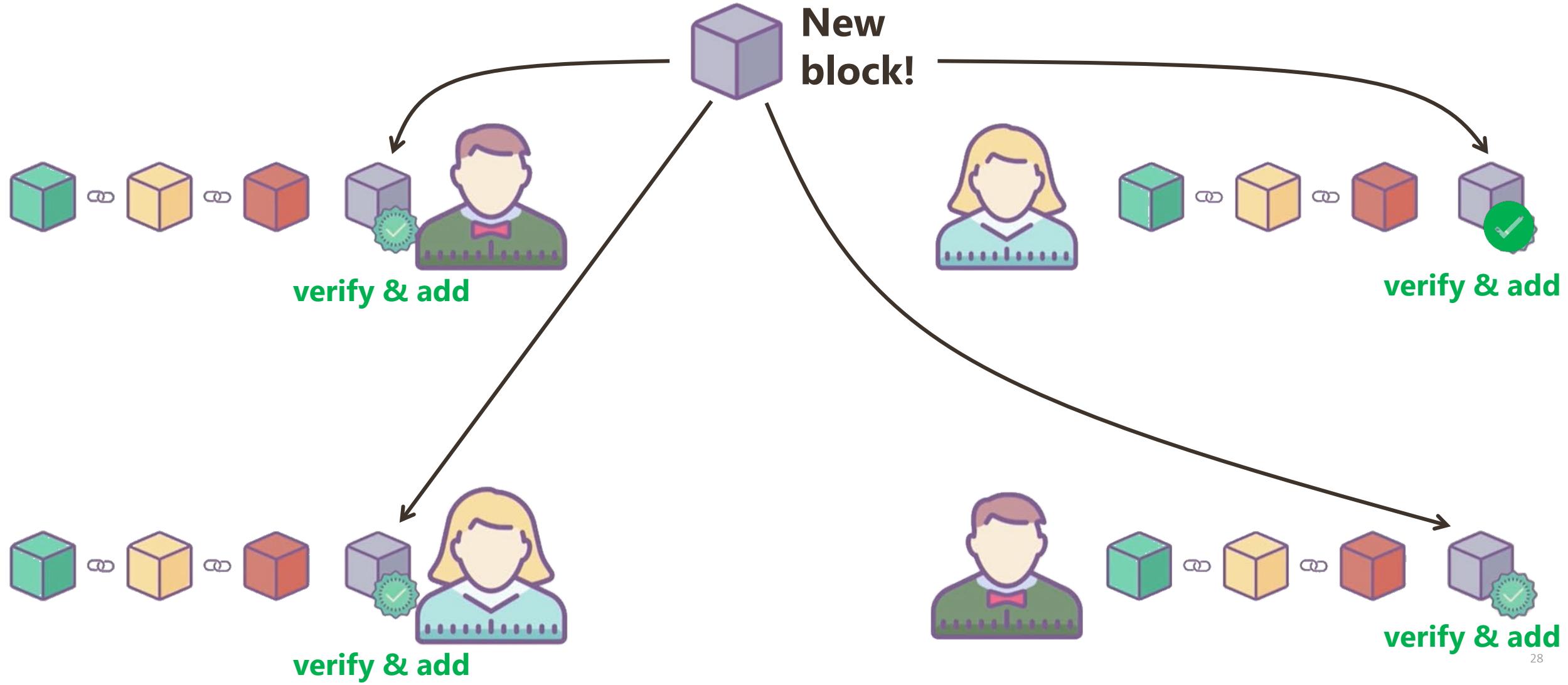
Block Tampering – cont.

Solution 2: Being distributed (P2P Network)

Let everyone (nodes) keep a copy of the ledger



Being Distributed...



Distributed Consensus on valid/invalid blocks

- Nodes able to come to agreement on what the state of the ledger should be
- No need to trust 3rd party
- Trustworthiness **without a 3rd party**



Block Tampering

- To be successful:

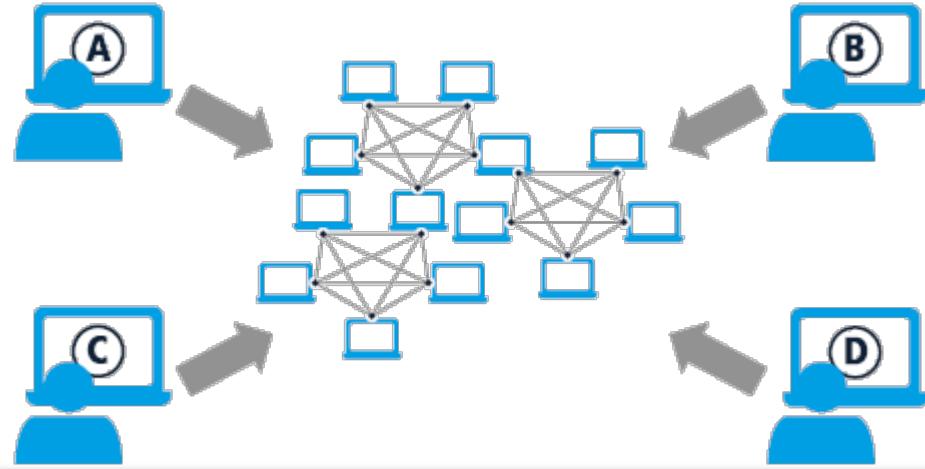
- ✓ Tamper with all the blocks in the chain
- ✓ Redo the proof of work (PoW) for each block
- ✓ Take control of > 50% of the P2P network

Only then the tampered block will be accepted by other nodes

- **Blockchain technology will provide**

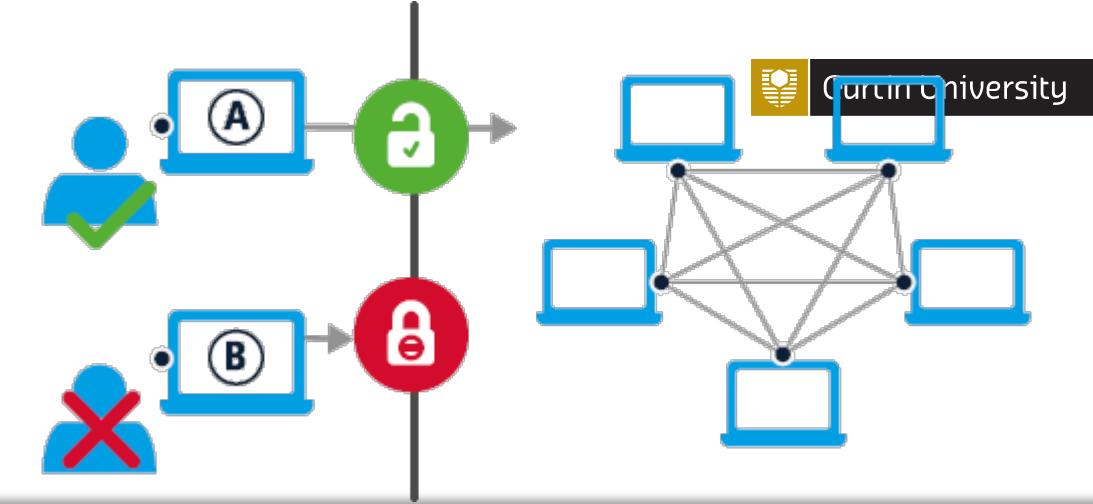
- ✓ Decentralization
- ✓ Transparency
- ✓ Immutability





PUBLIC BLOCKCHAIN

- Anyone can join the network and submit transactions
- Anyone can contribute computing power to the network and broadcast network data
- All transactions are broadcast publicly



PRIVATE BLOCKCHAIN

- Only safe-listed (checked) participants can join the network
- Only safe-listed (checked) participants can contribute computing power to the network and broadcast network data
- Access privileges determine the extent to which each safe-listed participant can contribute data to the network and access data from the network



Blockchain in-depth with Crypto Currency

- Fundamentals
- Bitcoin
 - Ledger
 - Adding/Verifying Transactions
 - Proof of Work (Miners / Participants)
 - Bitcoin Protocol Summary
 - Bitcoin vs. Legacy Financial System
 - Challenges of Bitcoin
 - Proof of Stake

Crypto Currency

- Digital Currency, Currency on bit-torrent like protocol
- Leverage blockchain technology
- No bank, **No third party**
- Can transfer directly between two parties via the use of private and public keys (no processing fees)



cryptocurrencies are
immune to counterfeiting



don't require a
central authority



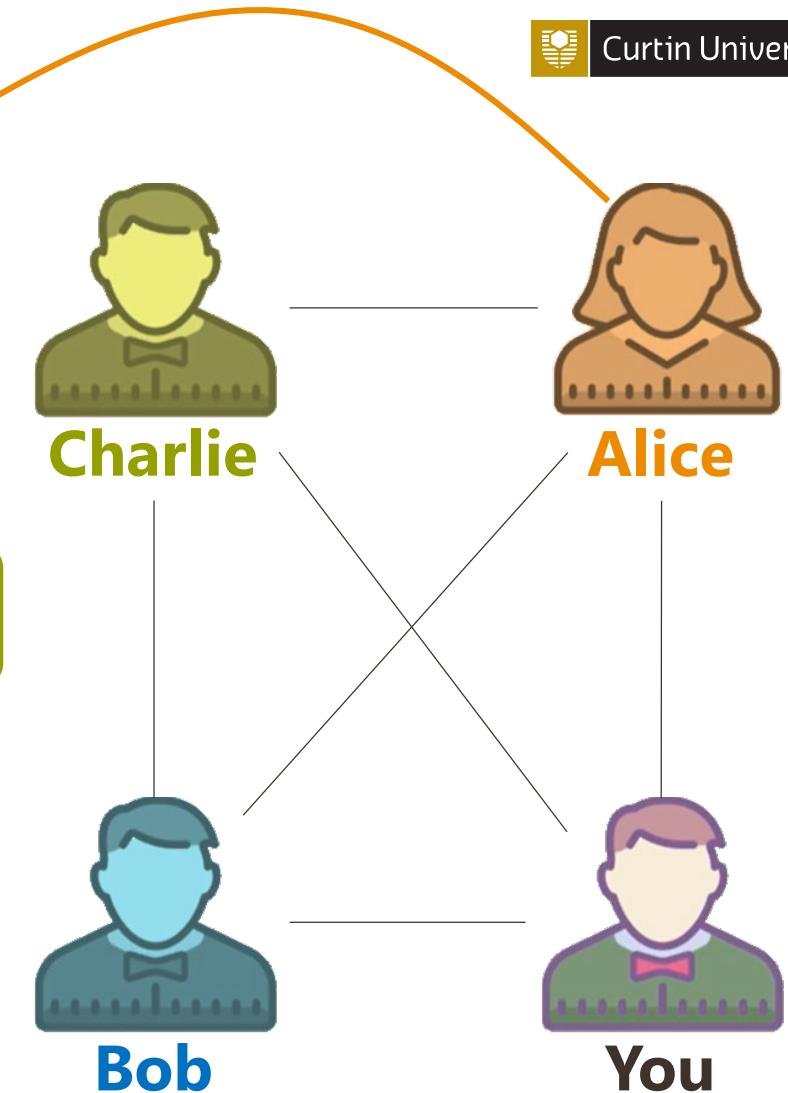
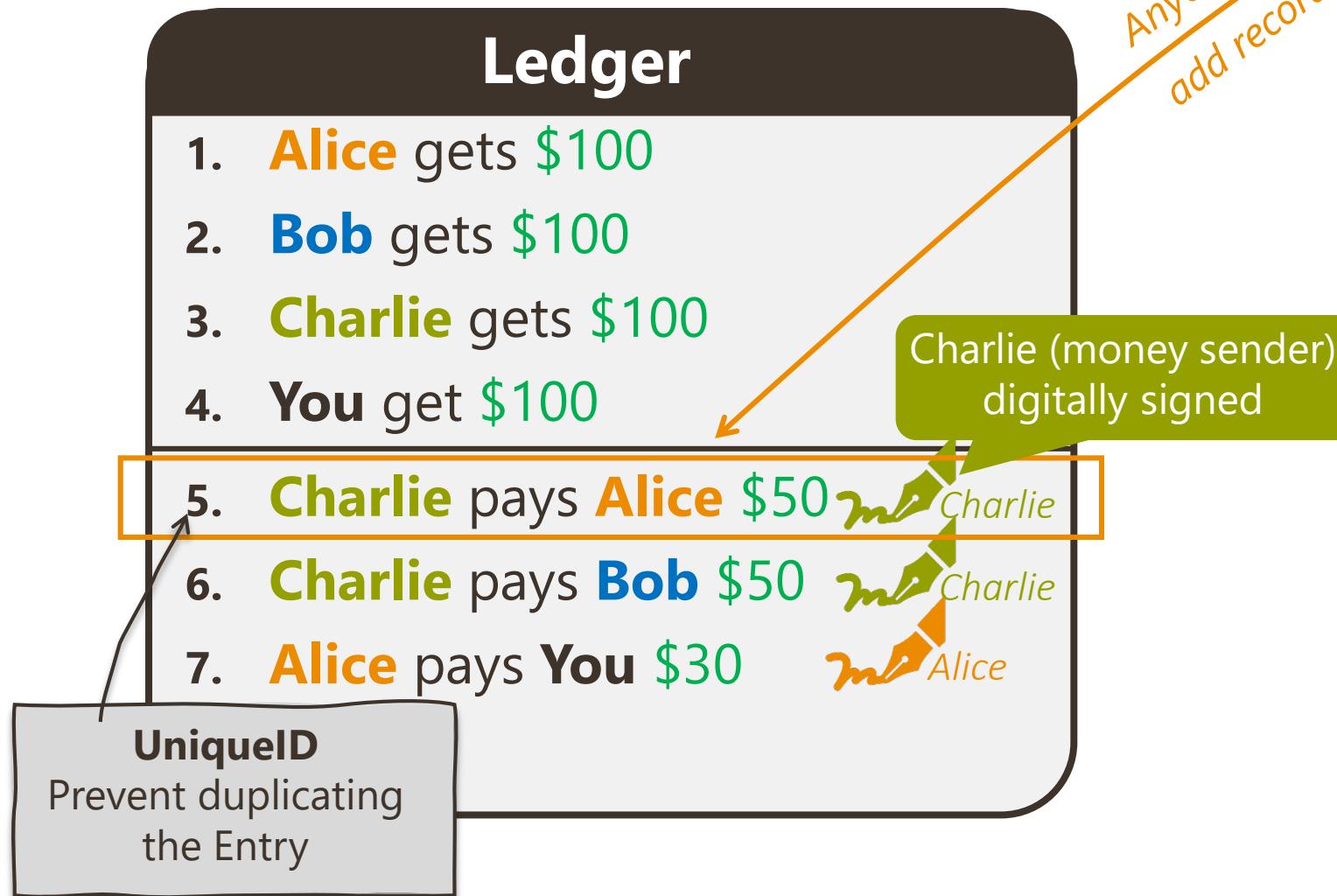
protected by strong and
complex encryption algorithms

Bitcoin

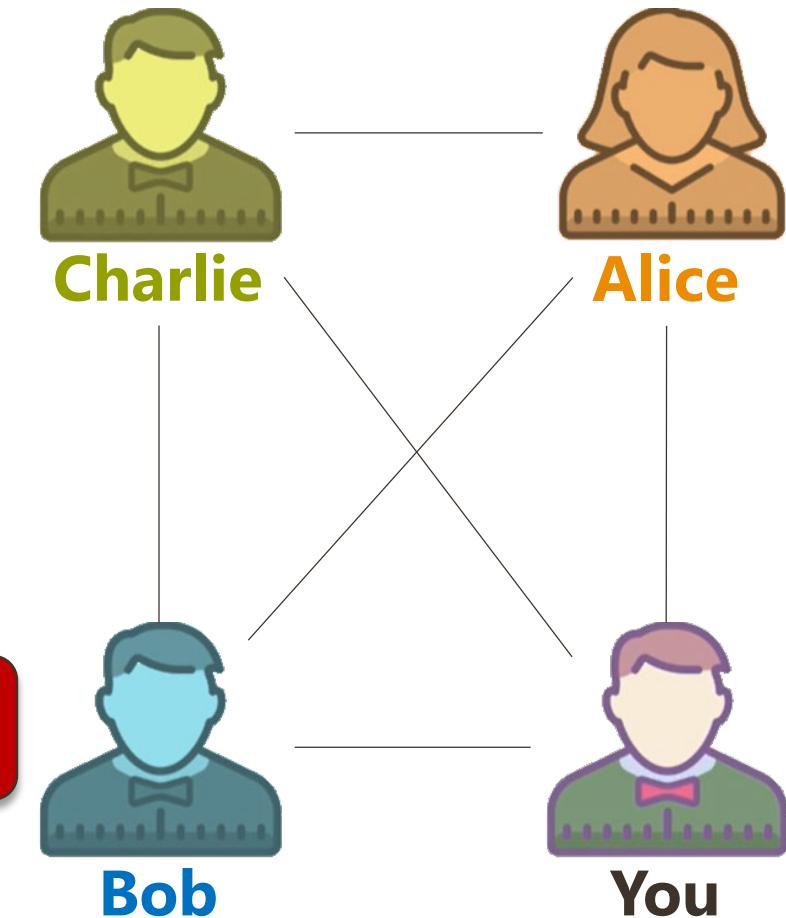


- **Leading digital currency** stored on a global, decentralized **P2P** blockchain
- **Blockchain is the underlying technology**
 - *enables transactions to take place in a secure and trusted manner between pseudo-anonymous parties*
- **Anyone can participate in the bitcoin (no intermediary)**
 - *Anyone can participate in the bitcoin blockchain and ownership can be digitally transferred without the need for an intermediary*
- **The creation or 'mining' of bitcoins is done through computers**
 - *solving complex equations; Currently, it is heavily energy-intensive, requiring improvements in energy efficiency*

Bitcoin Ledger



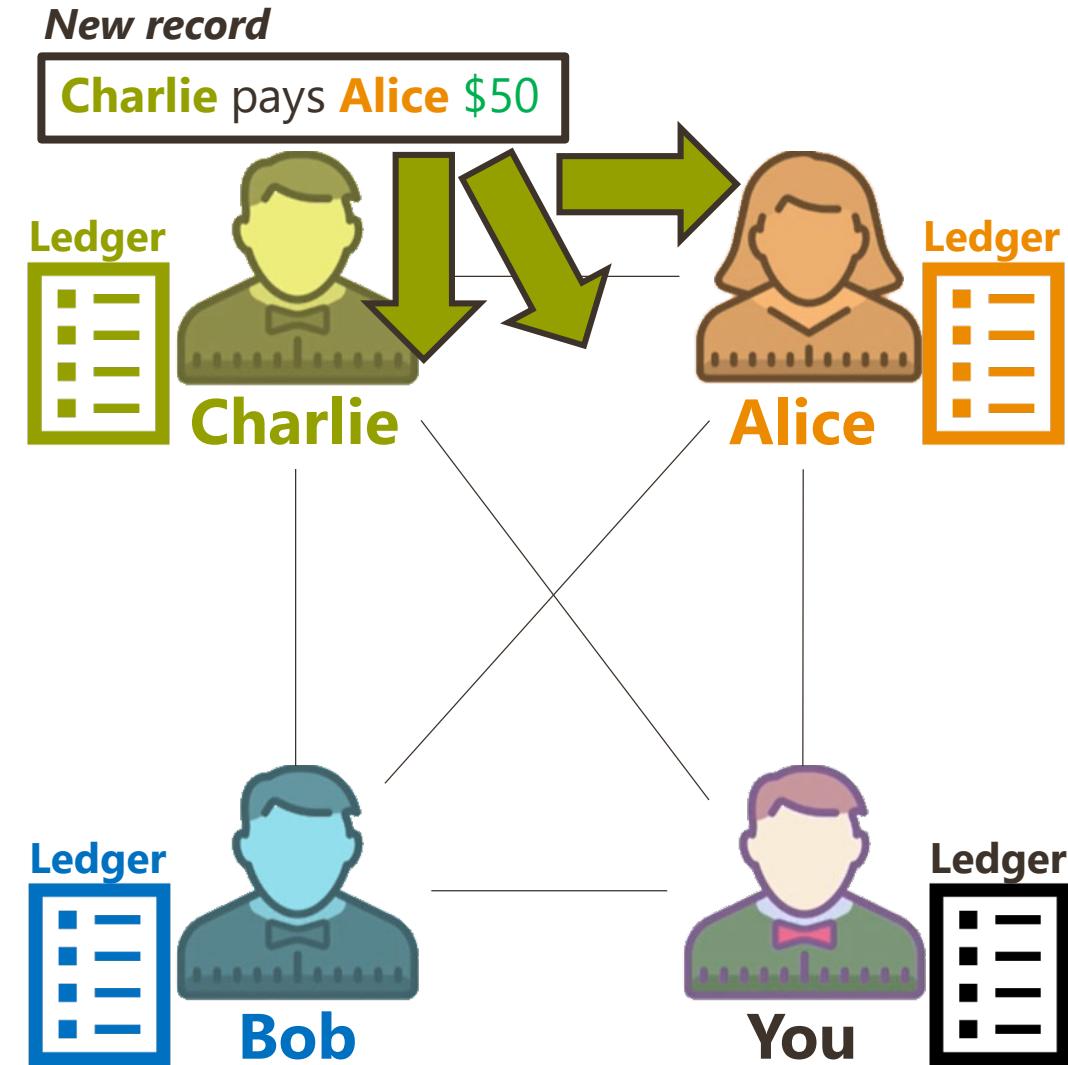
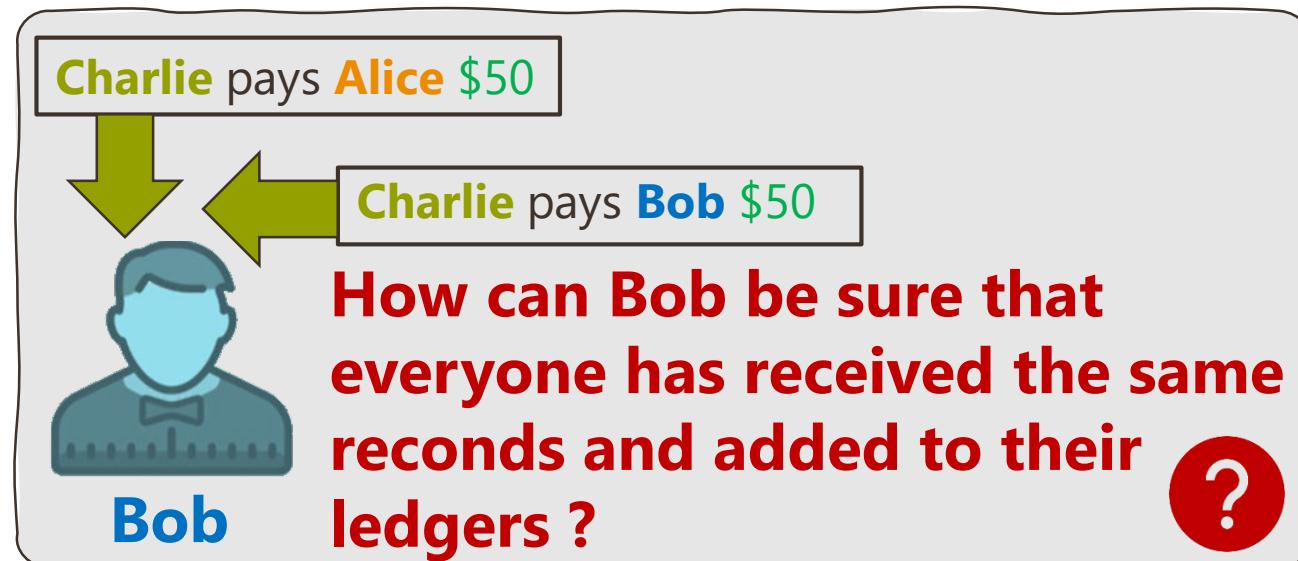
Verifying a Transaction



**Verifying a transaction requires
knowing the full history of Charlie**

Ledger

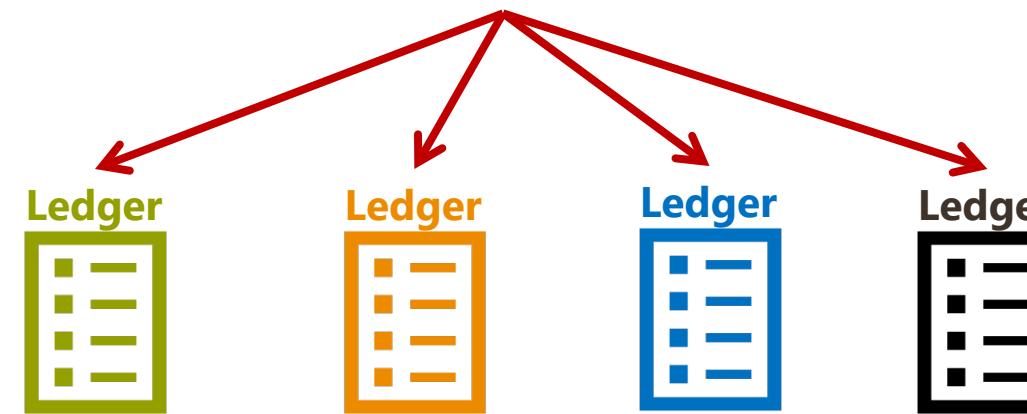
- Who owns the ledger?
 - ✓ Everyone (transparency)
- When a new record is added
 - ✓ Broadcast it to others
 - ✓ Others can verify and add to their ledgers



Ledger – cont.

- How can you get everyone to agree on what the correct ledger is?

Are these the same ?



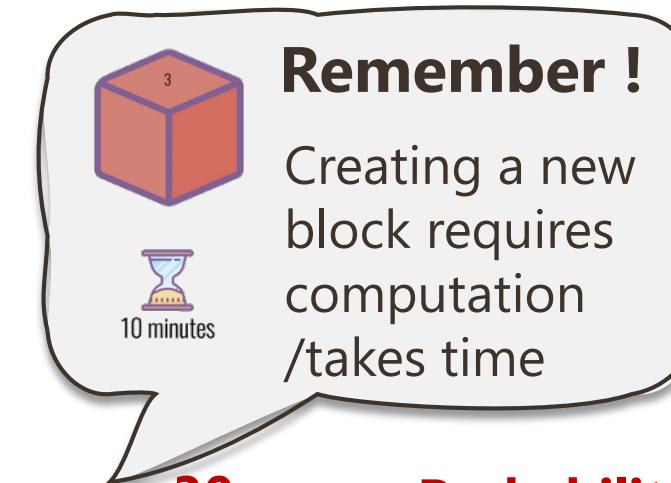
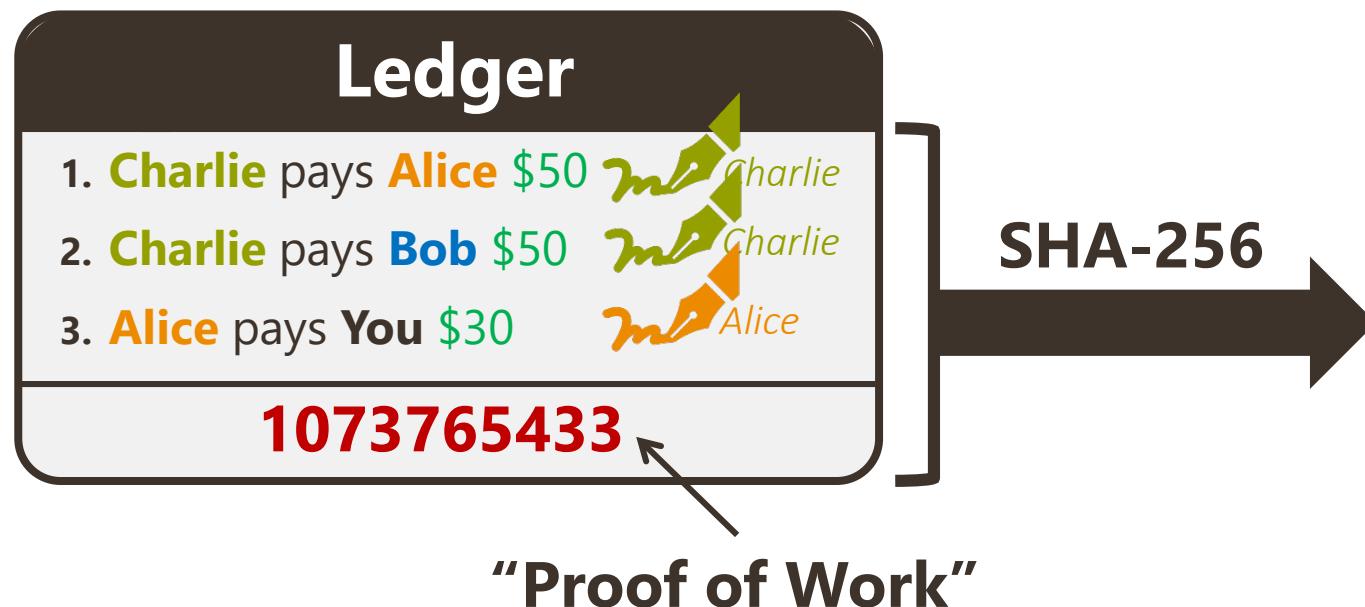
Trust whichever ledger has most work put into it

“Proof of Work”

Proof of Work

- Find **the special number** such that:

- ✓ Put it to the end of transactions
- ✓ Apply SHA-256 to whole ledger/block
- ✓ Hash value must start with 30 zeros



30 zeros, Probability = $\frac{1}{2^{30}}$

000000000000000000000000000011
111101011110110000110010001011
00110101101100110110010011010
11011000110110110000110010001000
01110001101101100100110010001000

Proof of Work – cont.

- **PoW verifies** you went through a large amount of work
- All work in **PoW is intrinsically tied** to the list of transactions in ledger
 - ✓ If you change one of those transactions, you will have to find the special number again
- **PoW takes approx. 10 min**
 - ✓ PoW: "Hash value must start with **30 zeros**"

xx number of zeros
which will take 10
min approx.

Ledger into Chain of Blocks

- Organize a given ledger into blocks & chain the blocks

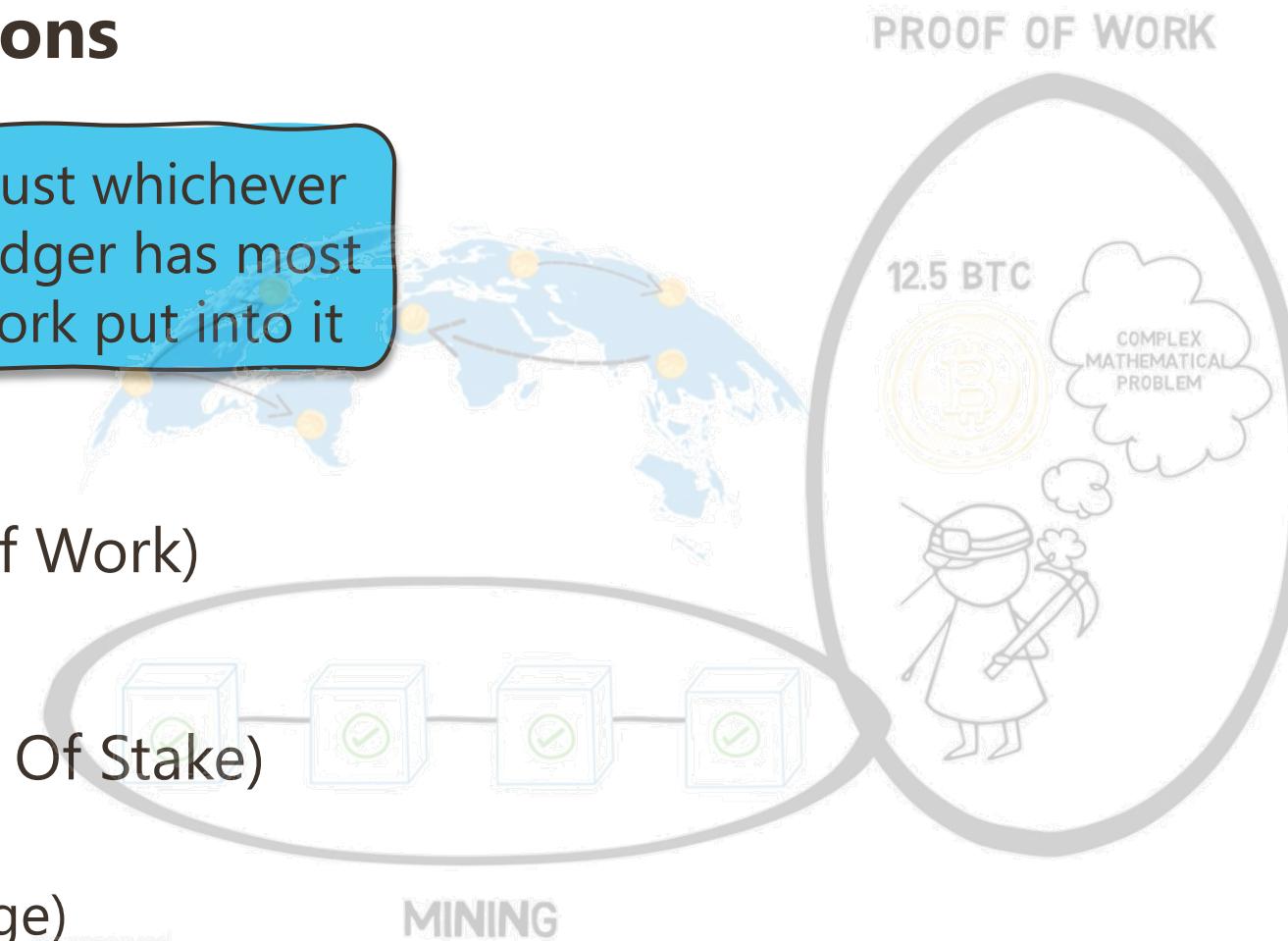


- Transaction** is **only valid** if digitally signed by its sender
- Block** is **only valid** if it has a proof of work

Bitcoin Protocol

- ✓ **Broadcast Transactions**
- ✓ **Only Accept Signed Transactions**
- ✓ **No Overspending**
- ✓ **Distributed Consensus**
- ✓ **Mechanism to Prove Work**

Trust whichever ledger has most work put into it



1. **Miners** to perform **PoW** (Proof Of Work)

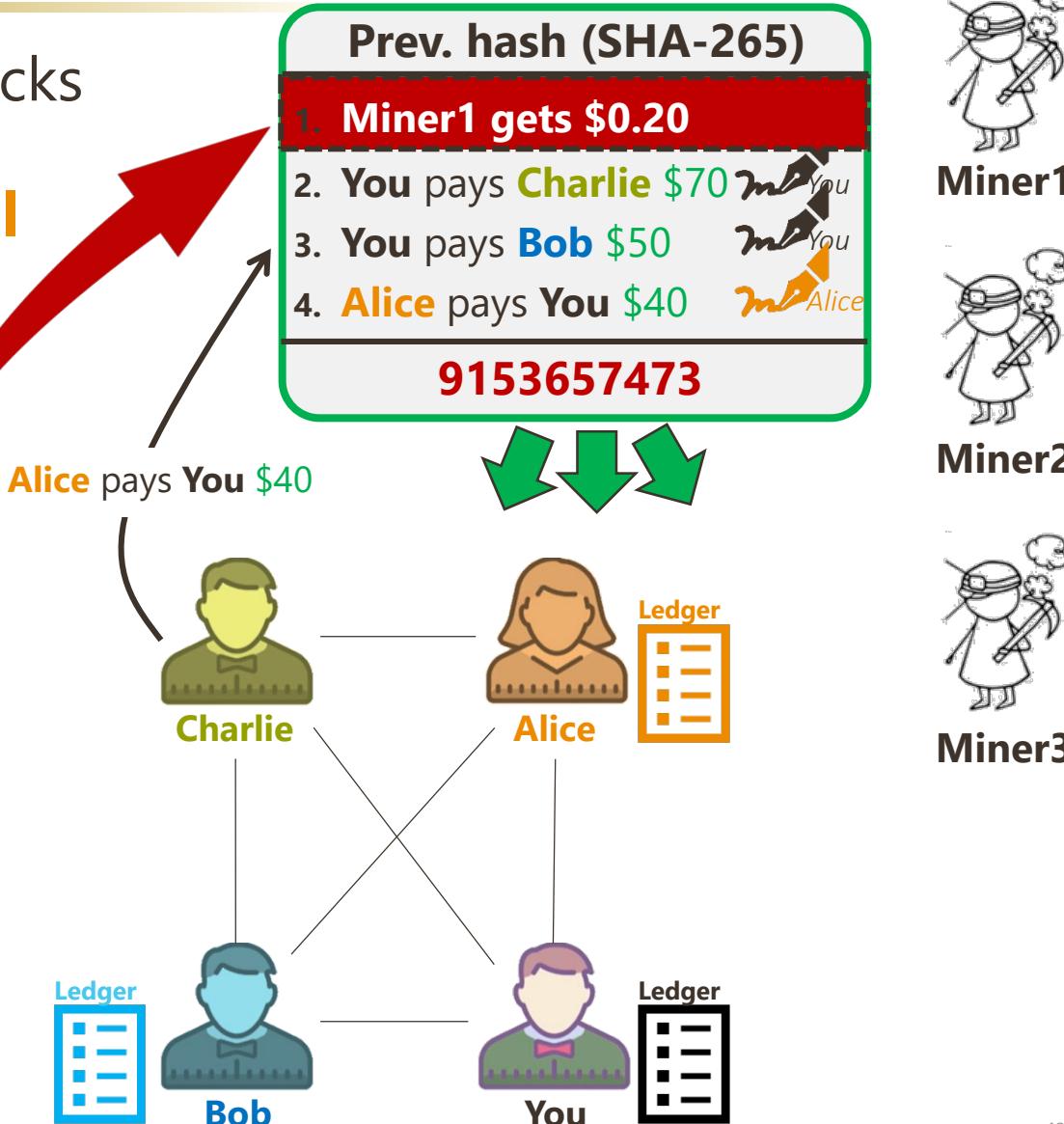
- **Reward:** Bitcoin (Block Reward)

2. **Validators** to perform **PoS** (Proof Of Stake)

- **Hold a stake**
- **Reward:** Bitcoin (transactions charge)

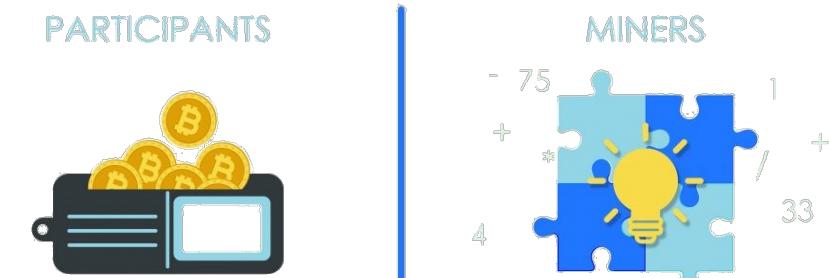
Miner (Block Creator)

- Listen to transactions, collect them into blocks
- Mine until one of the miners find the special number
- Block Reward: Get \$0.2 from the network
 - ✓ No sender / signature
 - ✓ Adds to total money supply
- Create a block (mining), broadcast to Alice, Charlie, Bob and You
- A node could be a miner or participant

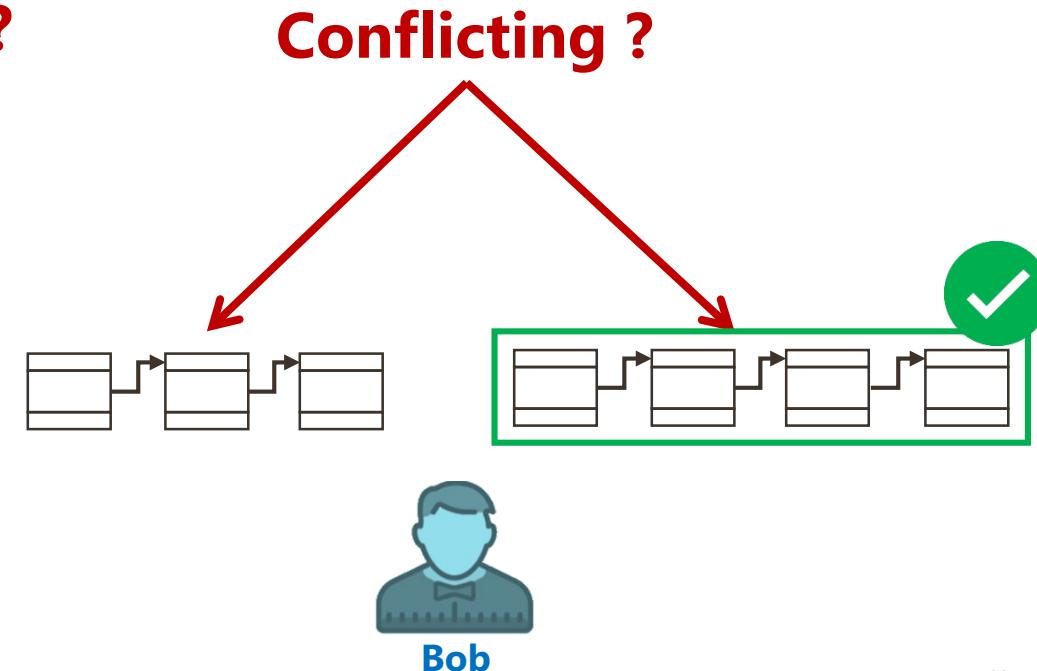


Participant (Non-Miner)

- People who use the system to make payments
- **Listen to blocks** broadcasted by miners,
 - Update the personal copies of the block chain



- What if you hear **two conflicting block chains?**
 - Refer to the longer one (one with more work)
 - If there is a tie, wait a few round to see who is longer



- **Distributed Decentralized Consensus**
 - Everyone agrees to give preference to the blockchain with most work put into it

Let's try to fool Bob

- **If Alice try to fool Bob with a fraud block**

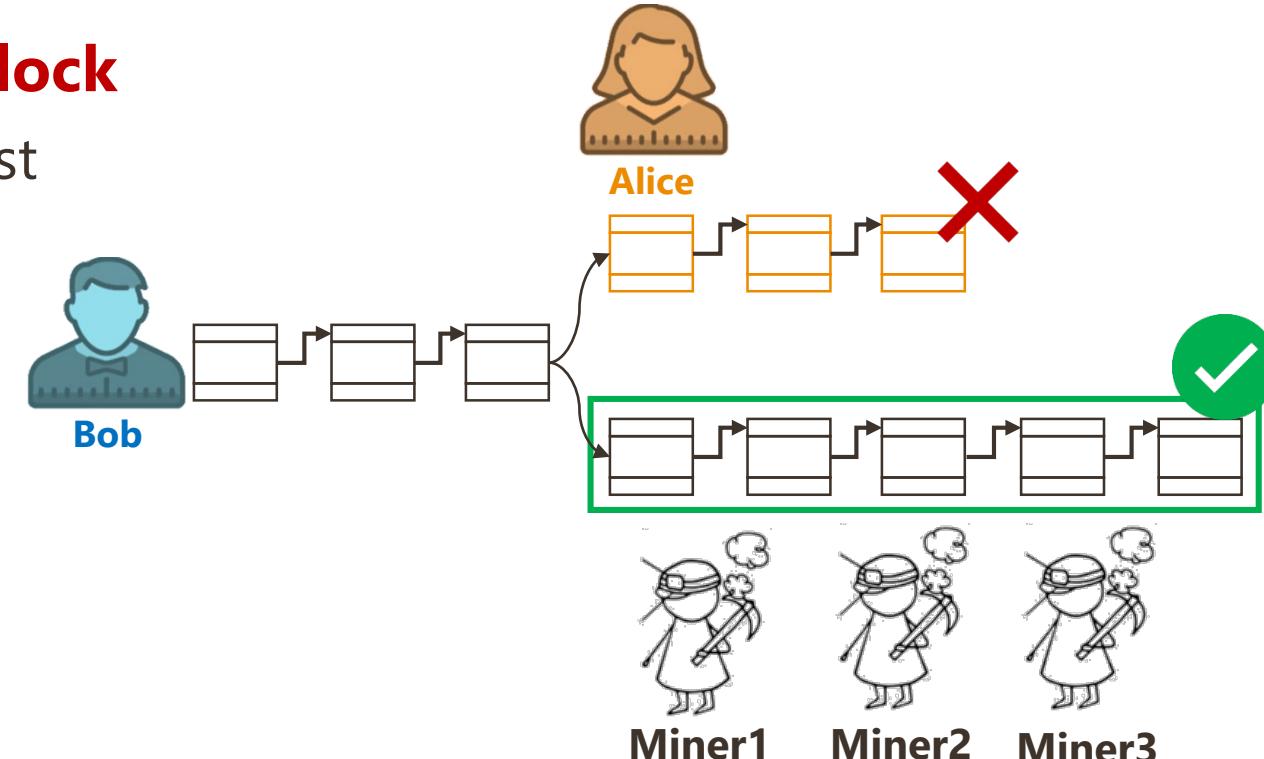
- by sending it only to Bob, but no broadcast

- Alice may be able to create a block
 - Find the special number (lucky winner)

- But to be the longest chain
 - Alice must have **>50%** of the computing

resources among all miners

- Bob will eventually **reject** Alice's chain **in favor of the longer chain**

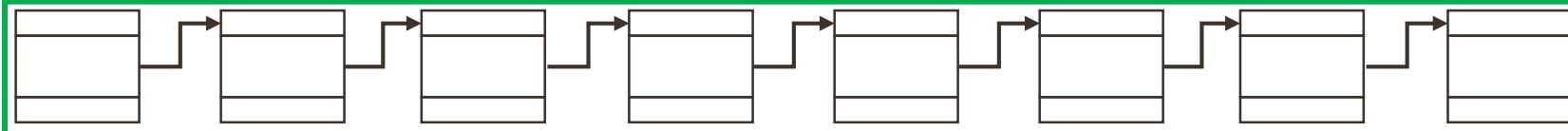


When to Trust?

Don't Trust Yet



...a little more...



Safe

This must be the
one everyone is
working on

Bitcoin Generation

- **Miners generate Bitcoin via mining**

- Inject Bitcoin to the total economy

- **Bitcoin Halving**

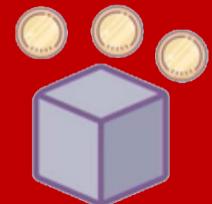
- Happens after every 210,000 blocks
 - It cuts the supply of new Bitcoins in half via halving the miner's block production rewards
 - 50% fewer BTC for verifying transactions

- **Bitcoin – The 21 million upper cap**

- There will only be 21 million Bitcoins that will ever exist



Miners would still be rewarded with transaction fees associated with the block they mine

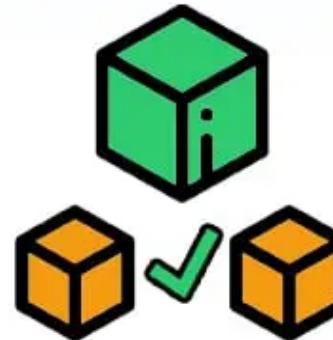




Someone requests a transaction.



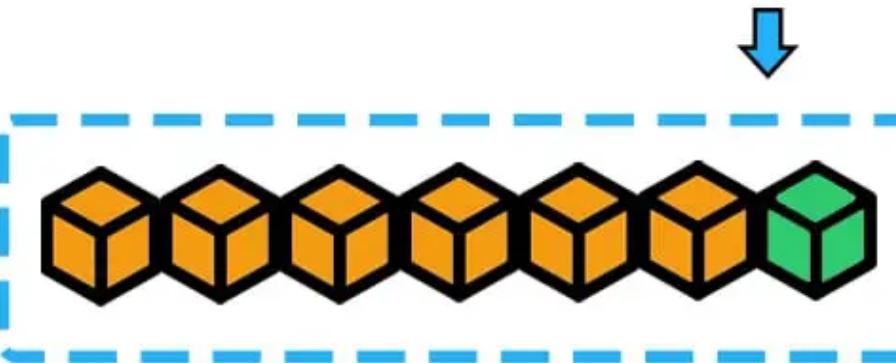
The requested transaction is broadcast to a P2P network consisting of computers known as nodes.



The P2P network of nodes validates the transaction and the user's status using known algorithms.



The transaction is complete!



The new block is then added to the existing blockchain in a way that is permanent and unalterable.

A verified transaction can involve cryptocurrency, contracts, records, or other information.



Cryptocurrency



Has no intrinsic value in that it is not redeemable for another commodity.



Has no physical form and exists only in the network.



Its supply is not determined by a central bank, and the network is completely decentralized.

Bitcoin Summary

Bitcoin vs. Legacy Financial System

- 33% credit card transactions are fraud (nobody wants to accept)
- With bitcoin, funds are legit
- Challenging the legacy financial system

User Facing		\$
Verification	Decentralized Trust List Verification <i>Digital signatures</i> <i>Cryptographic hash functions</i>	Card 
Underlying System	Bitcoin Protocol	Bank (3 rd party) Banking System

Challenges

- Every transaction needs **computational power**

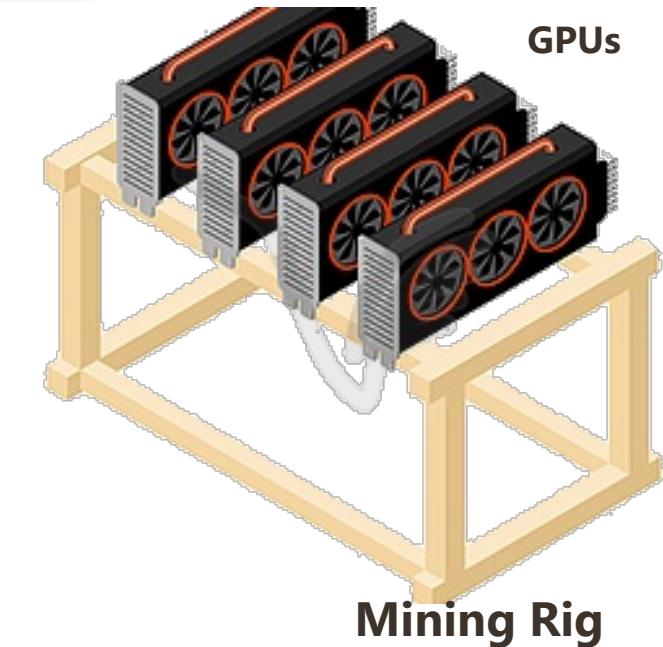
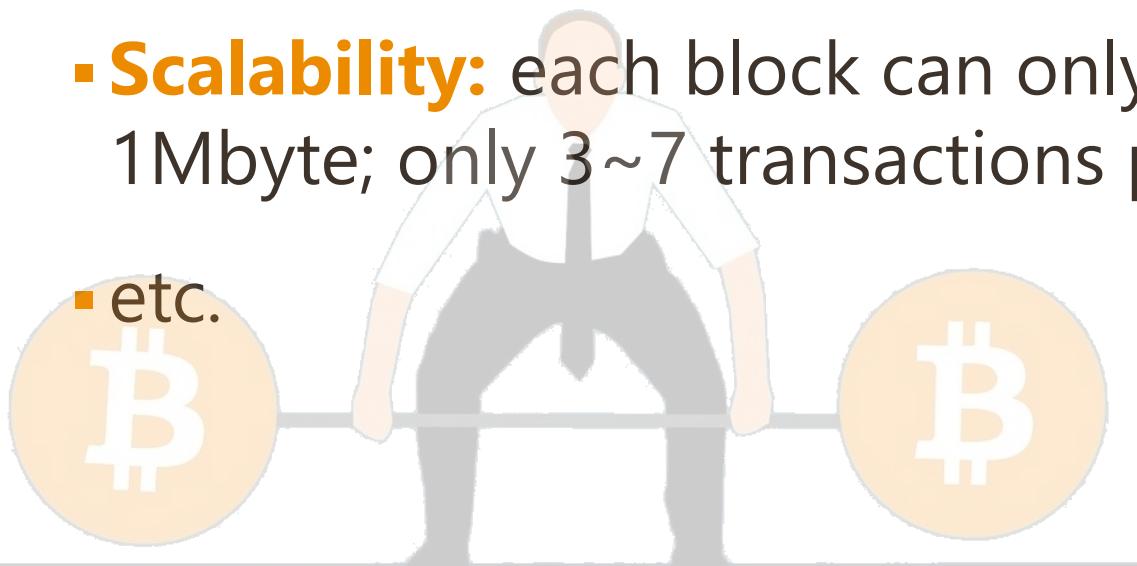


- **Latency** per transaction

- Unstable pricing make it too popular for speculators but not helping as a currency

- **Scalability:** each block can only contain 1Mbyte; only 3~7 transactions per second

- etc.

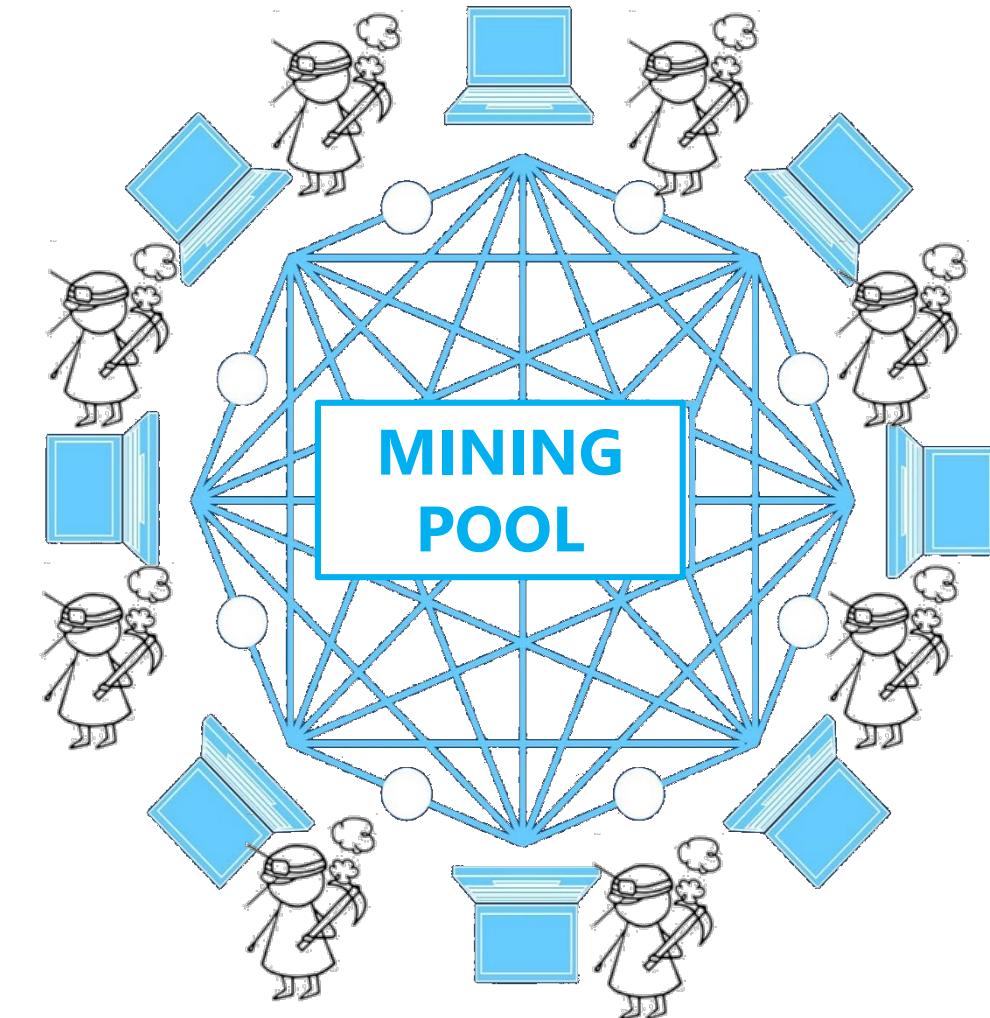


Computational requirement

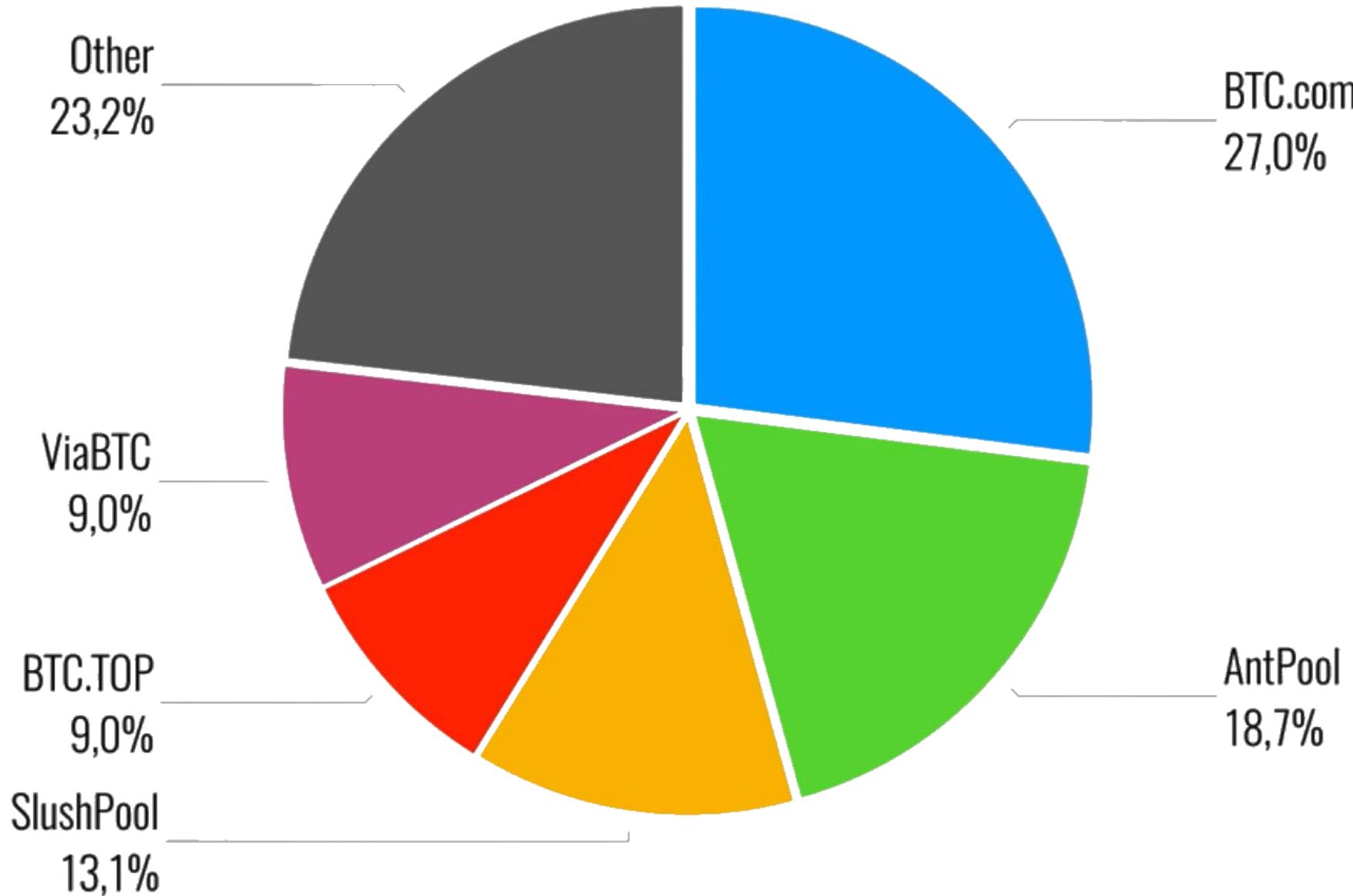


Proof of Work - Issues

- High energy usage
 - Bitcorn miners use 54TWh – enough to power 5M households in US
- Encourage users to use **mining pools**
 - This will make **blockchain more centralized** ☹
- Need an alternative method
 - **Proof of Stake**



Mining Pools



Proof of Stake

▪ Terminology



~~Miners~~

~~Mining~~

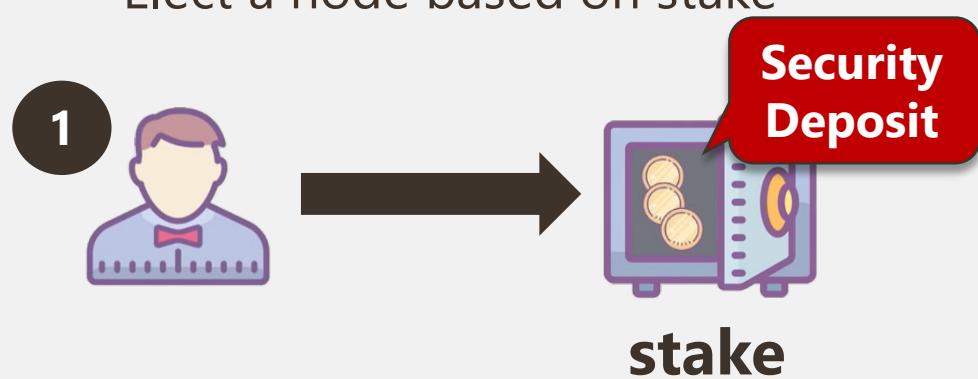


Validators

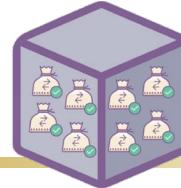
Minting / Forging

▪ Validator Node

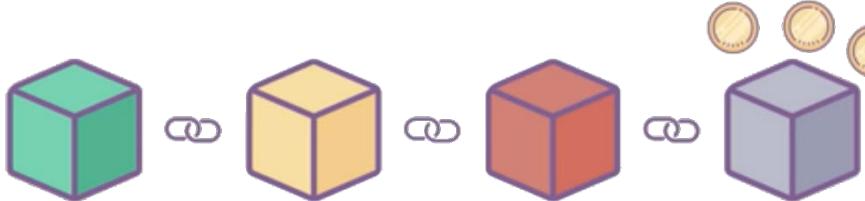
- Elect a node based on stake



Proof of Stake



- Validator validates all transactions inside a block



- **Block Reward:**

- Fees associated with the transactions inside the block

- **How to trust other validators?**

- **Validators will lose** part of their **stake** if they approve fraudulent transactions
 - Stake will be held until sum of **all transaction fees > stake**

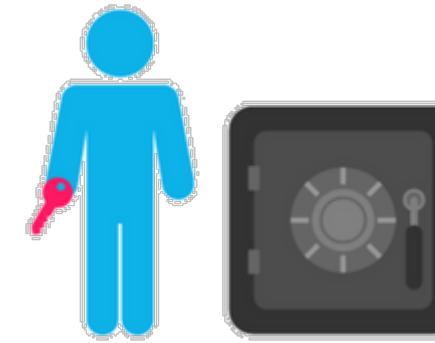
- All transaction fees + stake will be released after certain period of time



PoW



Proof of work is a requirement to conduct an expensive computer calculation, also called mining



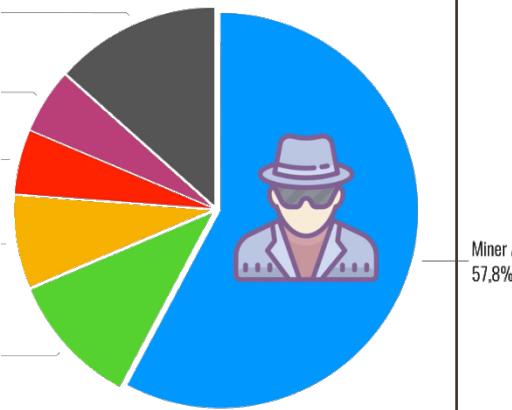
PoS

Proof of stake, the creator of a new block is chosen in a deterministic way, depending on its wealth, also defined as stake

Everyone can mine

More centralized with Mining Pools

- ✓ Control large portion of blockchain
- ✓ If 3 join, can start approving fraudulent transactions
- ✓ Acquiring 51% is easier hence highly risky



Only select a few "validators"

More decentralized 51% is practically impossible

= \$79,826,299,343.76

Expensive mining equipment

Encourage more people to mine/validate

PoS Shortfalls ?

1. Richer will get high priority to be chosen

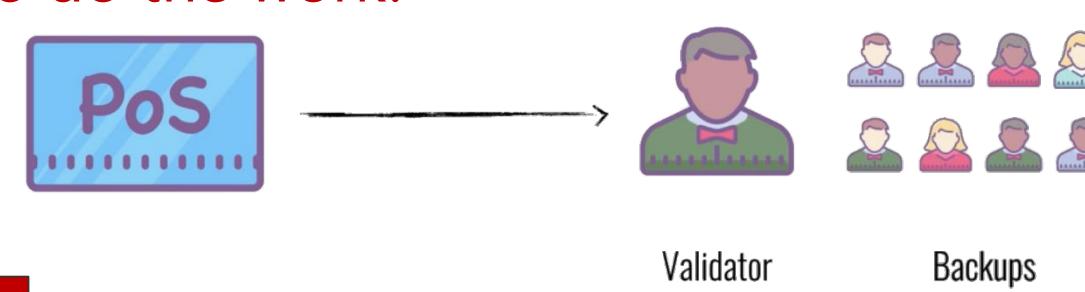
- If chosen, Richer can earn more money

- **Proposal: Coin age-based selection**

- proof-of-stake system combines randomization with the concept of "**coin age**"
- a number derived from the number of coins multiplied by the number of days the coins have been held

2. Validator is chosen but **doesn't show up to do the work!**

- **Use backup validators**

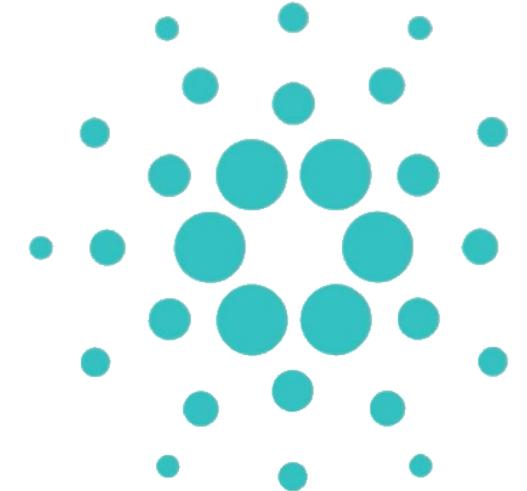


In short, PoS brings different risks compared to PoW

PoS Systems



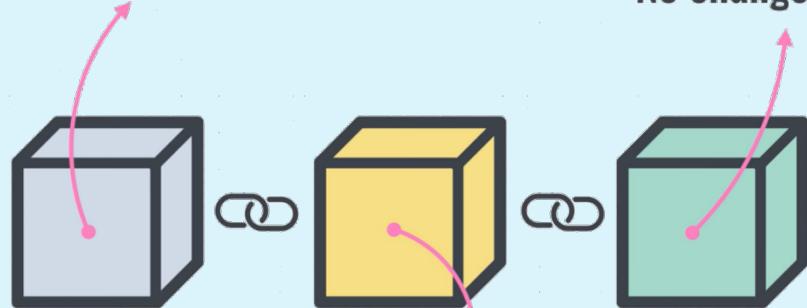
Casper



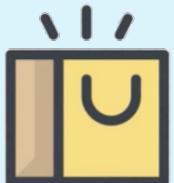
Cardano

Distributed database

No changes allowed



Everyone can add



Curtin University

Blockchain Applications

- Smart Contract
- Other Blockchain Applications
- Blockchain in Logistics

Smart Contract

- **Digital Contracts**, tiny computer program stored in Blockchain (**immutable, distributed**)

- Once smart contract is **created**, it **can never be changed**

- Output of the contract is **validated by everyone** in the network
- A single **person cannot force** the contract to release the funds!



- i.e. **Ethereum** (smart contract approach)
- i.e. Kick Starter (manual approach, 3rd party)



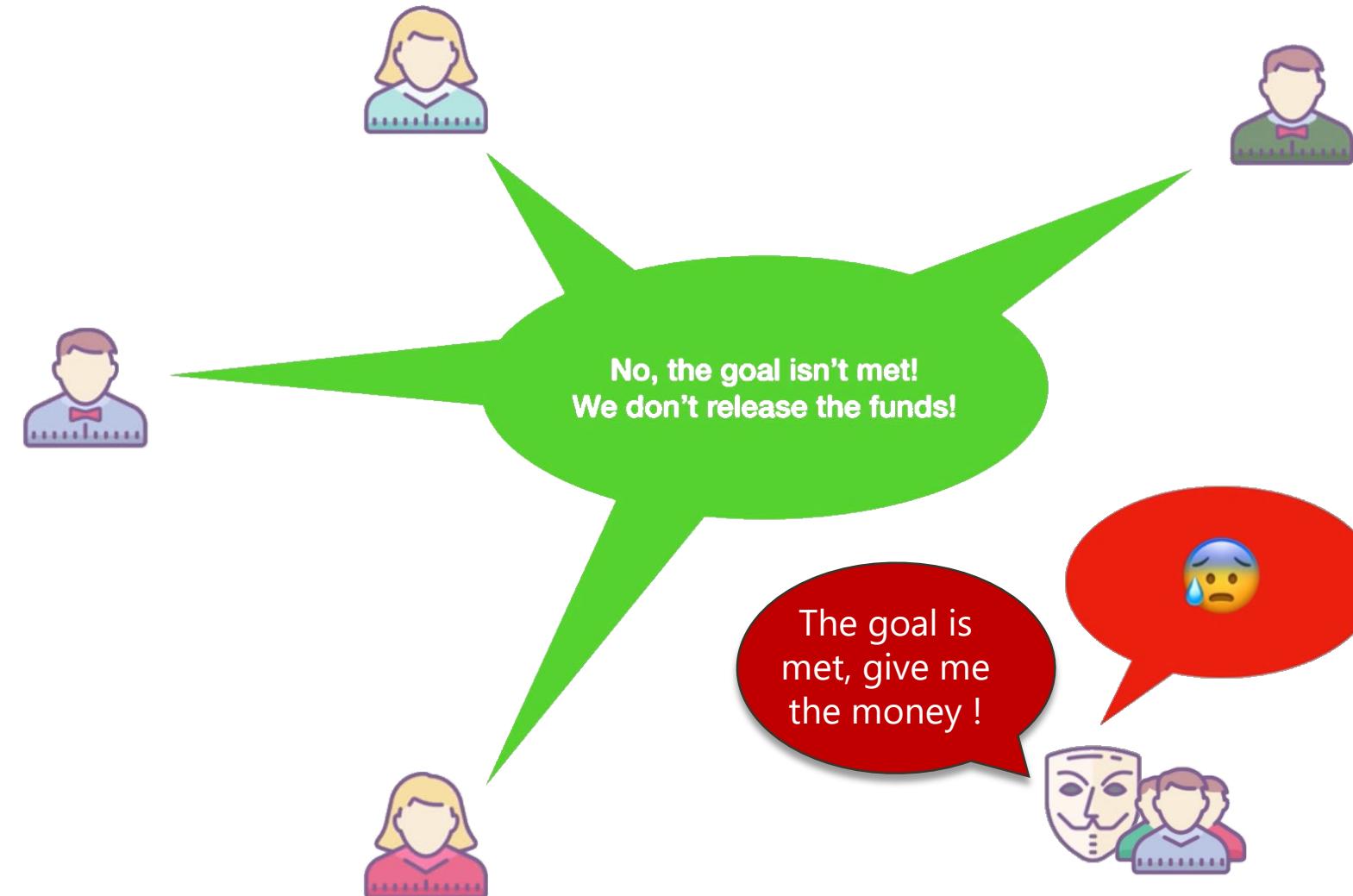
Ethereum



KICKSTARTER

Smart Contract – Kick Starter

- Hold all the **money** until a certain goal is reached
- If reached, money will be debited to the project
- If failed, money will be refunded
- This is an **automatic process**



Extending the Idea

- Banks could use to issue loans / automatic loan payments



Banks
Loans
Automatic payments



Insurance
Process claims



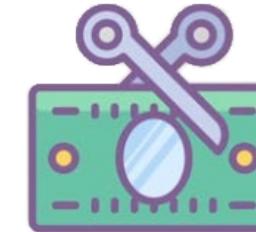
Postal
Payment on delivery



Medical records



E-notary



Collecting taxes

Japan votes Blockchain

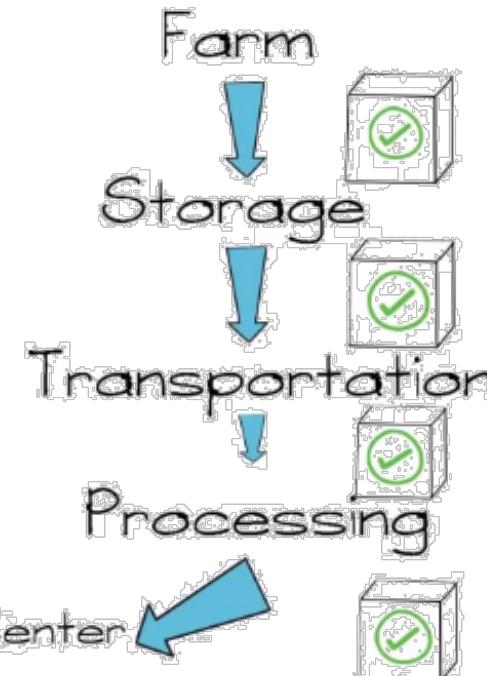




Blockchain in Logistics



THEY WERE UNABLE TO
DETERMINE THE POINT OF FAILURE





Future of Networking

- More Emerging Networking Technology
 - Network Automation
 - Wireless Technology
 - Networking Hardware

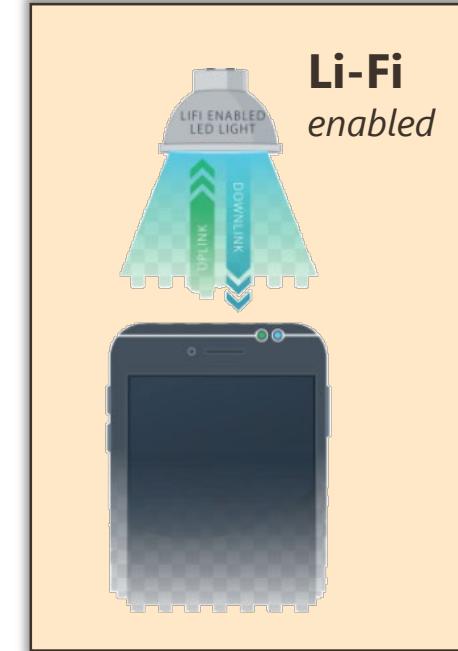
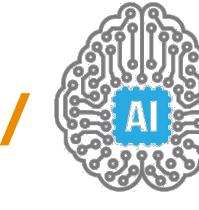
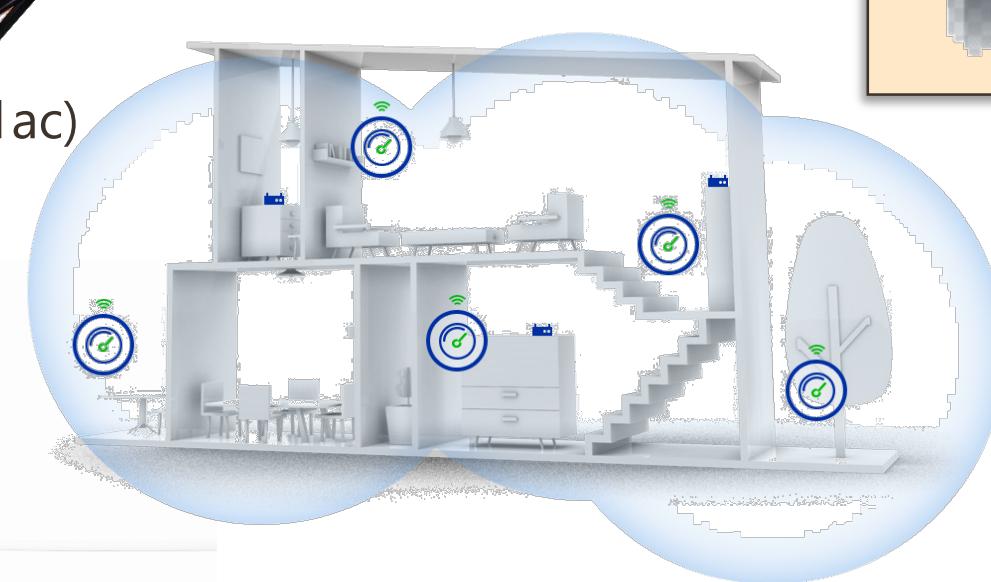
More Emerging Network Tech

- Software-Defined Wide Area Network (**SD-WAN**)
- Networking **automation with Machine Learning /**
- **Wireless Technology**

- Wi-Fi 6
- Li-Fi

▪ Home Networking Hardware

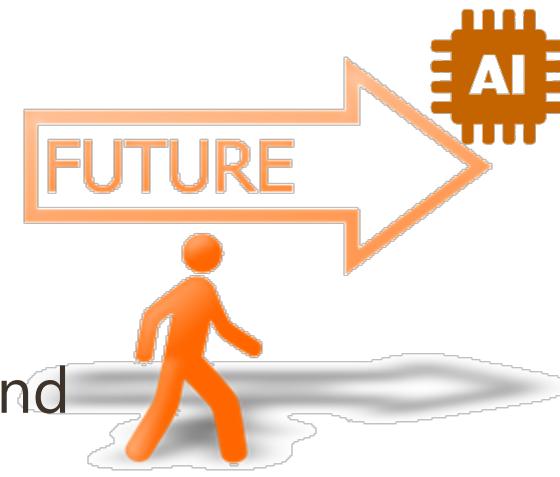
- Gaming Routers (tri-band, MU-MIMO, 802.11ac)
- Mesh WiFi
Smart wireless extender
- Intelligent Google Nest WiFi
- Networking via power lines



Li-Fi
enabled

What does the network of the future look like?

- Fast, smart and increasingly **autonomous**.
- **Self-driving, self-healing** and **self-operating** most of the time, and help solve problems across all industries and organizations.
- E.g., remotely connecting doctors with their patients or **powering next-level robotics**, to **identifying mission-critical parts** wherever they are within the supply chain on Earth and **sending them up to the international space station** when they're needed.
- The power of these autonomous networks and the freedom they provide to organizations around the world will be **almost limitless**.





- **IoT**
 - Main categories
 - Applications
- **SDN**
 - Fundamentals
 - SDN Model
 - SDN Benefits
- **Blockchain**
 - Fundamentals
 - Block Tampering
 - Proof of Work
 - Distributed Consensus
 - Public and Private Blockchains
- **Blockchain - Bitcoin**
 - Fundamentals
 - Ledger
 - Adding Transactions
 - Verifying Transactions
 - Proof of Work
 - Miners
 - Participants
 - Bitcoin vs Legacy Financial System
 - Challenges
 - Proof of Work Issues
 - Proof of Stake
- **Blockchain – Other Applications**
 - Smart Contract
 - Other Blockchain Applications
 - Blockchain in Logistics
- **Future of Networking**
 - Wireless Technology
 - Networking Hardware

GOOD LUCK
FOR YOUR
EXAM
DO THE BEST



CNCO2000

2021 Semester 1

THE END

THANK YOU

Make tomorrow better.