THE UNIVERSITY OF MELBOURNE

ELEN 90061
Communication Networks

Module 3 – Link Layers

Local Area Networks

Dr. Rajitha Senanayake

Acknowledgement: these slides are a modified version prepared by Prof. Tansu Alpcan

Lesson Overview

- Serial communications in embedded systems and IoT
- 802.11 WiFi
- Ethernet, hubs and switches
- Delay, loss, throughput

Suggested Reading

Note that there is overlap between these reading materials. It is a comprehensive list and you can use slides as a guideline for what to focus on.

- Chapter 4 from Tanenbaum
- Chapters 5 and 6 from Kurose-Ross

THE UNIVERSITY OF MELBOURNE

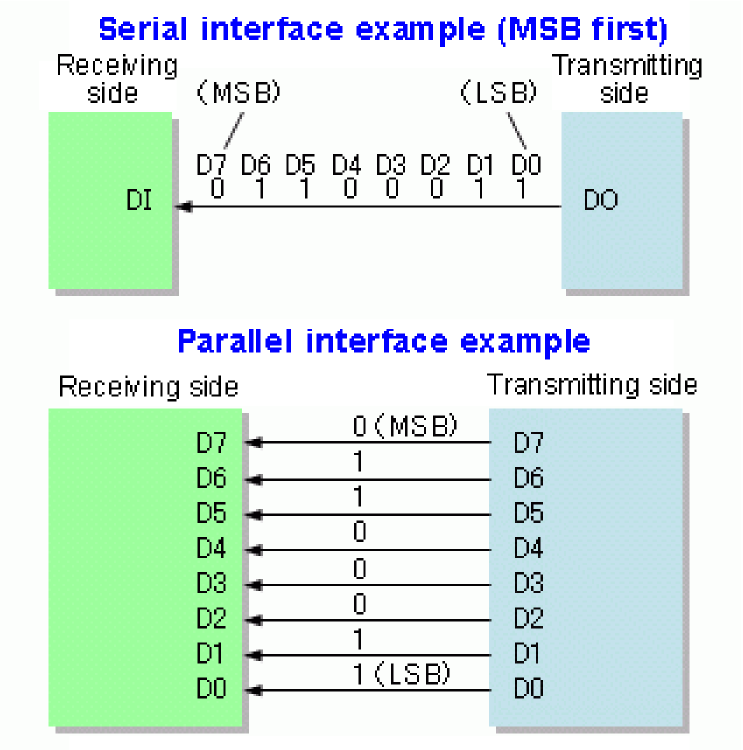
Serial Communications in Embedded Systems and IoT

application
transport
network
link
physical

M2-L3

Serial vs Parallel Communication

- **Serial communication** is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus.
- In contrast, in **parallel communication** several bits are sent as a whole, on a link with several parallel channels.

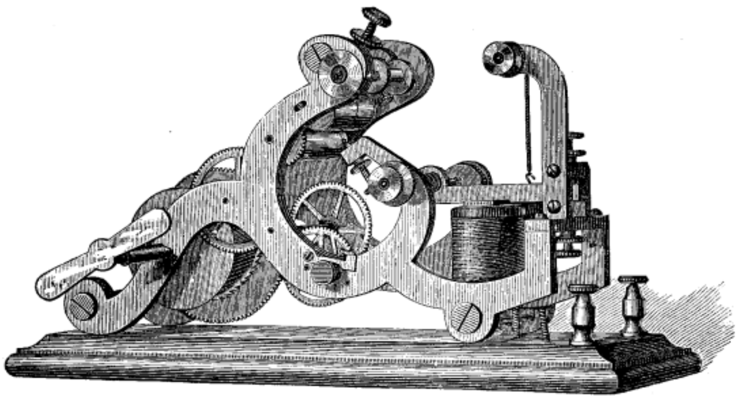


MSB/LSB: Most/Least significant Bit

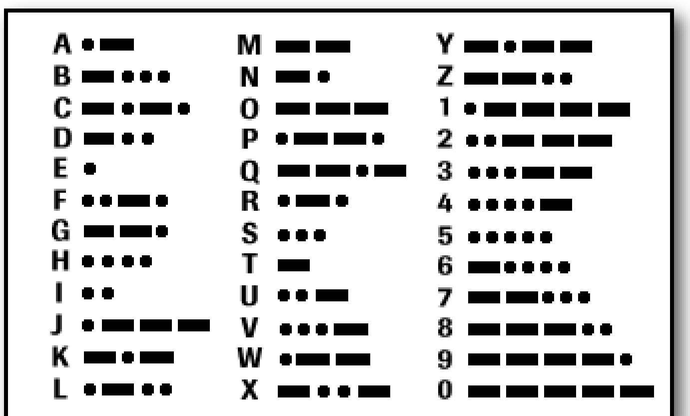
Classic serial communications was widely used, e.g. for PC modems and peripherals. These days it is still relevant to PCs (in the form of USB, HDMI), embedded systems, and Internet-of-Things (IoT) applications.

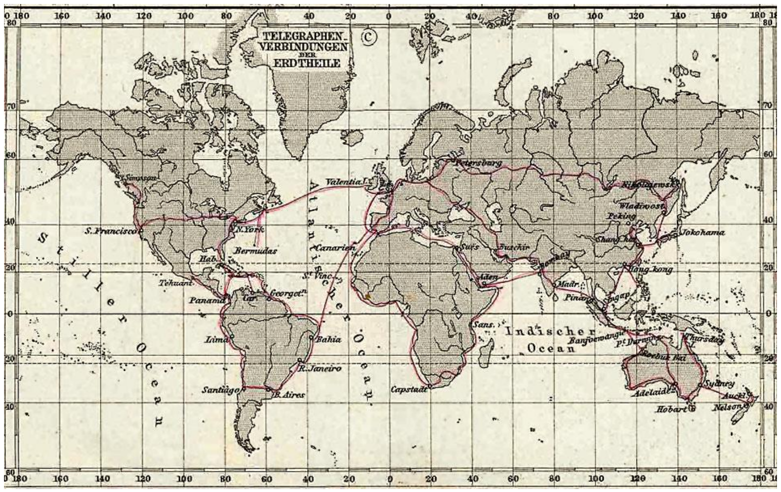
Telegraph as Serial Communication

- Single line with two states: on-off (closed circuit=marking, open circuit=spacing)
- Morse code: serial code of dots and dashes; non-binary (in a sense precursor to ASCII).



MORSE REGISTER.
Manufactured by L. G. Tillotson & Co., New York.

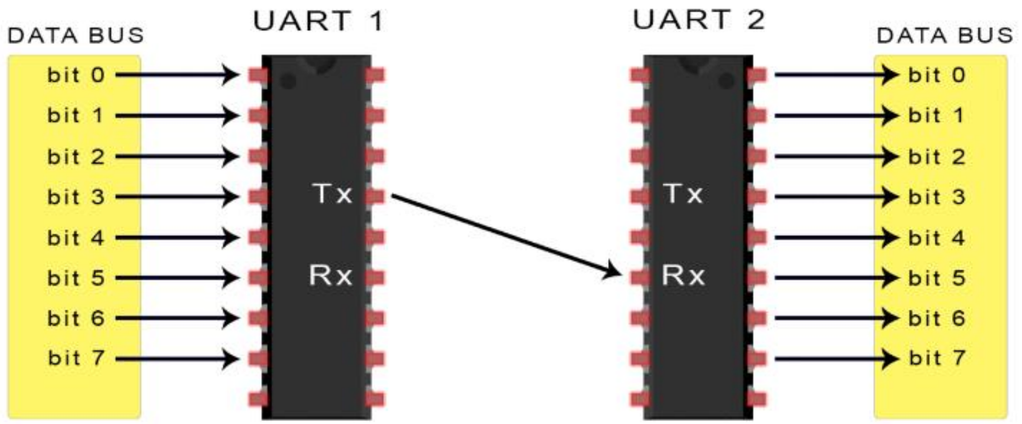


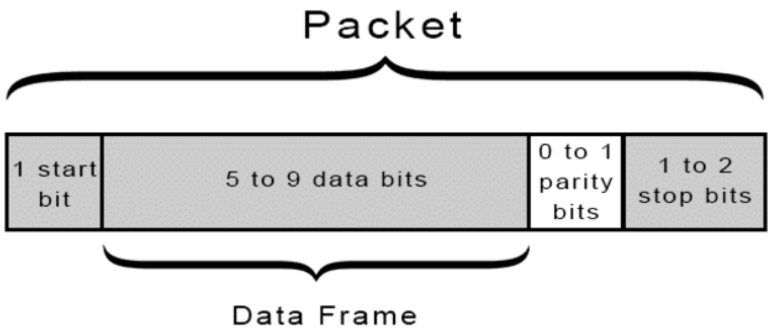


Major telegraph lines in 1891.

Universal asynchronous receiver/transmitter (UART)

- **Universal Asynchronous Receiver/ Transmitter (UART)** converts between parallel and serial data and handles other low-level details of serial communications.
- U**S**ART (Universal **Synchronous**/Asynchronous Receiver/Transmitter)
  - supports synchronous operation.
- It is a hardware component (often within *CPU* or *SoC*) that implements a variety of serial protocols in embedded systems, e.g. RS232, RS485.
- Low speed up to 115200 baud, usually 9600 baud.



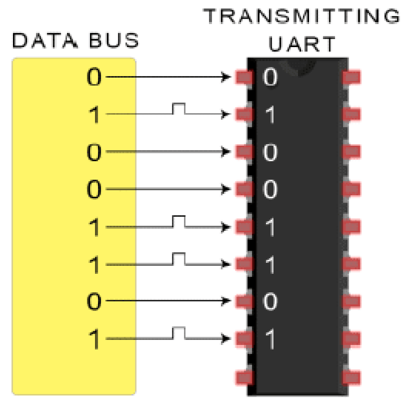


SoC *- System on a Chip*

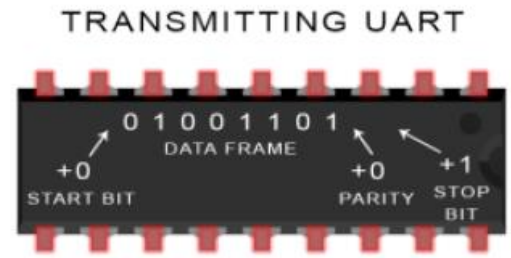
https://en.wikipedia.org/wiki/Baud
Baud = symbols per second
Same as bits per second if there are only two symbols (0-1).

UART Communication Steps

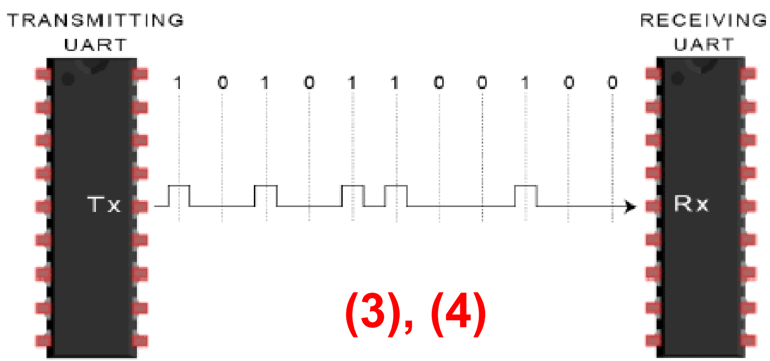
1. Transmitter receives data in parallel from data bus
2. adds the start bit, parity bit, and the stop bit(s) to the data frame.
3. The entire frame is sent serially from the transmitting UART to the receiving UART.
4. The receiving UART samples the data line at the pre-configured baud rate.
5. The receiving UART discards the start bit, parity bit, and stop bit from the data frame.
6. The receiving UART converts the serial data back into parallel and transfers it to the data bus.



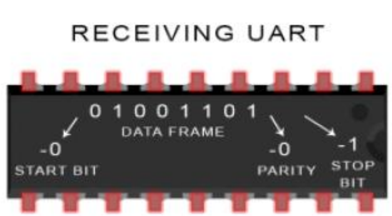
(1)



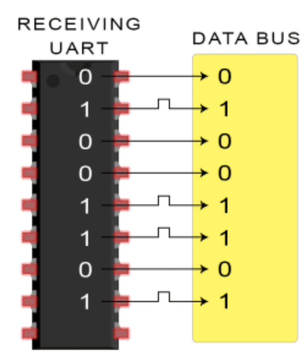
(2)



(3), (4)



(5)



(6)

UART Advantages and Disadvantages

Advantages

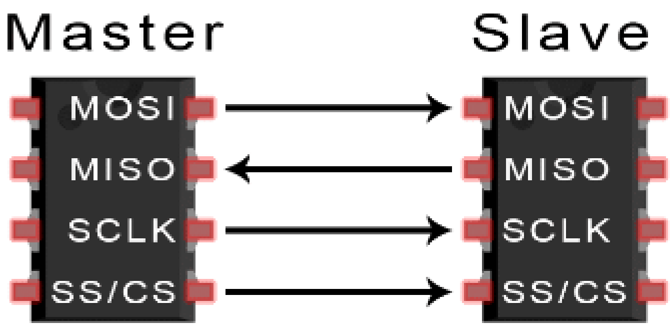
- Only uses two wires (can be simplex, half or full-duplex)
- No clock signal is necessary
- Has a parity bit to allow for error checking
- Well documented and widely used

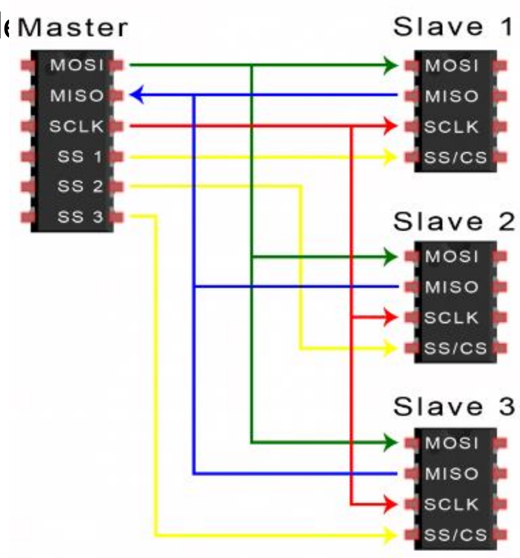
Disadvantages

- The size of the data frame is limited to a maximum of 9 bits
- Doesn't support multiple slave or multiple master systems
- The baud rates of each UART must be within 10% of each other.
- It is *slow*!

Serial Peripheral Interface (SPI)

- Small displays, SD card modules, RFID card reader modules, and 2.4 GHz wireless transmitter/receivers all use SPI to communicate with microcontrollers.
- One master (leader) can control more than one (theoretically unlimited) slaves (followers).
- Four wires communicating in serial in synchronous manner.
  - MOSI (Master Output/Slave Input) – Line for the master to send data to the slave.
  - MISO (Master Input/Slave Output) – Line for the slave to send data to the master.
  - SCLK (Clock) – Line for the clock signal.
  - SS/CS (Slave Select/Chip Select) – Line for the master to sel
- Data can be transferred without interruption in a continuous stream (up to 10 Mbps).
- Can operate in half or full-duplex modes.

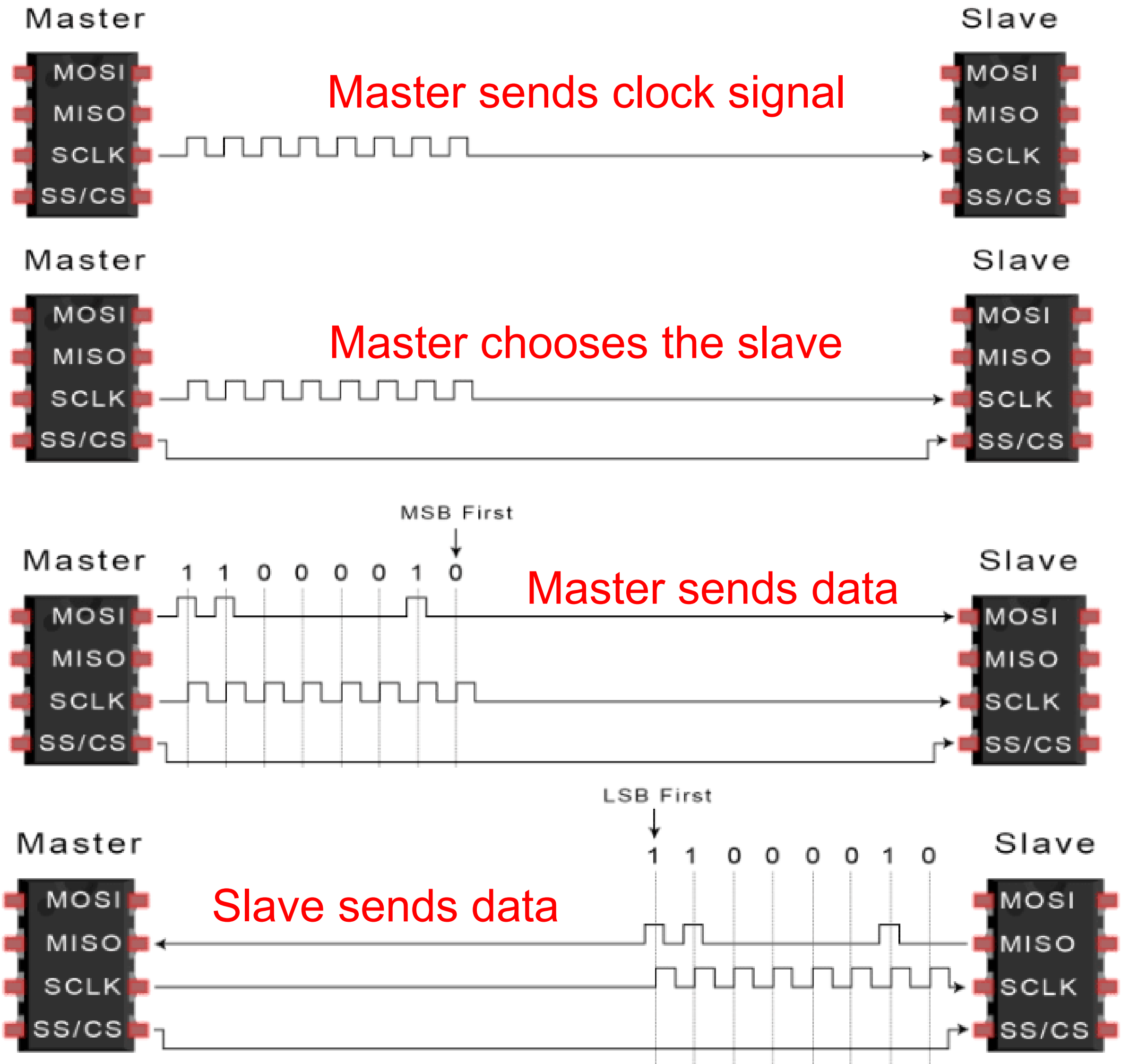




Serial Peripheral Interface (SPI)

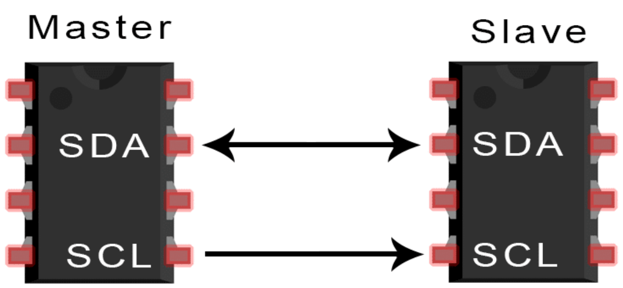
Advantages
- Continuous streaming
- Full duplex

Disadvantages
- 4 wires
- Single master (leader)
- No error checking
- No ack

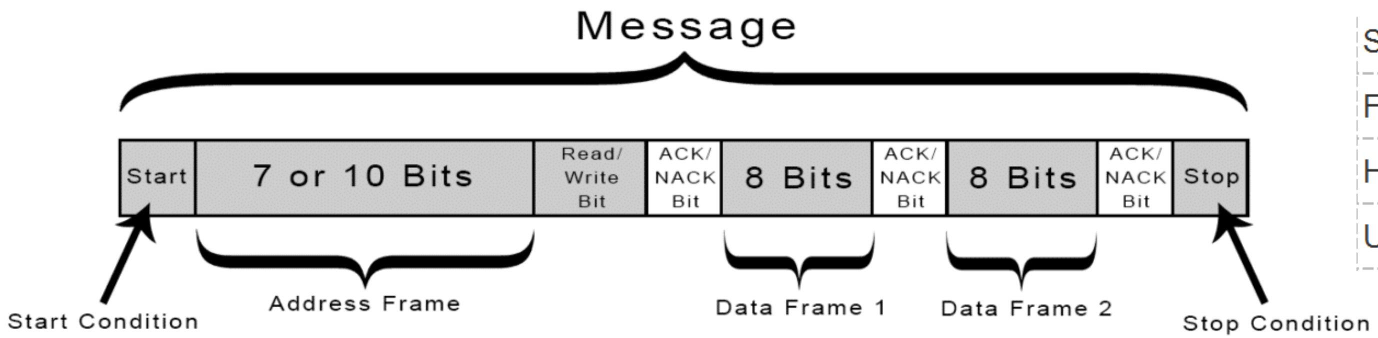


Inter-Integrated Circuit (I2C)

- All I2C-bus compatible devices have an on-chip interface which allows them to communicate directly with each other via the I2C-bus.
- Simple bidirectional 2-wire bus
  - SDA (Serial Data)
  - SCL (Serial Clock)



- Half-duplex.
- Unlimited masters (leaders) and maximum 1008 slaves (followers).
- With I2C, data is transferred in messages, which are broken up into frames of data. Each message has an address frame that contains the binary address of the slave.

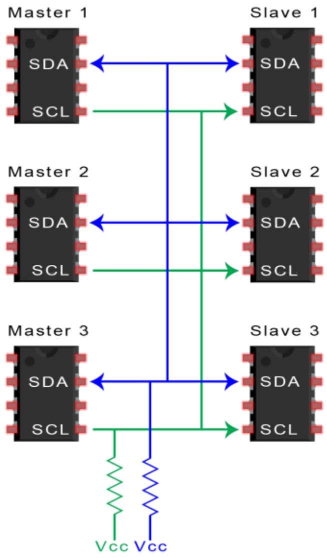


Standard mode= 100 kbps
Fast mode= 400 kbps
High speed mode= 3.4 Mbps
Ultra fast mode= 5 Mbps

Serial Communications Further Reading

Q: in I2C, what happens when there are multiple masters?

A: Arbitration is needed!



Further Reading on Serial Communication

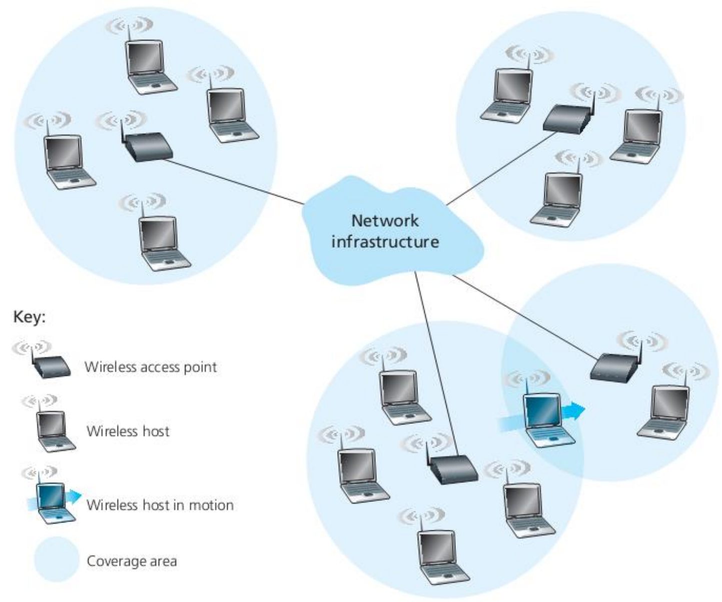
- **Book**: Serial Port Complete: The Developer's Guide, by Jan Axelson, 2nd Edition, Lakeview Research LLC, 2007.
- UM10204 I 2C-bus specification and user manual @ NXP https://www.nxp.com/docs/en/user-guide/UM10204.pdf
- http://www.circuitbasics.com/basics-of-the-spi-communication-protocol
- http://www.circuitbasics.com/basics-uart-communication/
- http://www.circuitbasics.com/basics-of-the-i2c-communication-protocol

IEEE 802.11
Wireless Networks

application
transport
network
link
physical

Wireless Network Overview

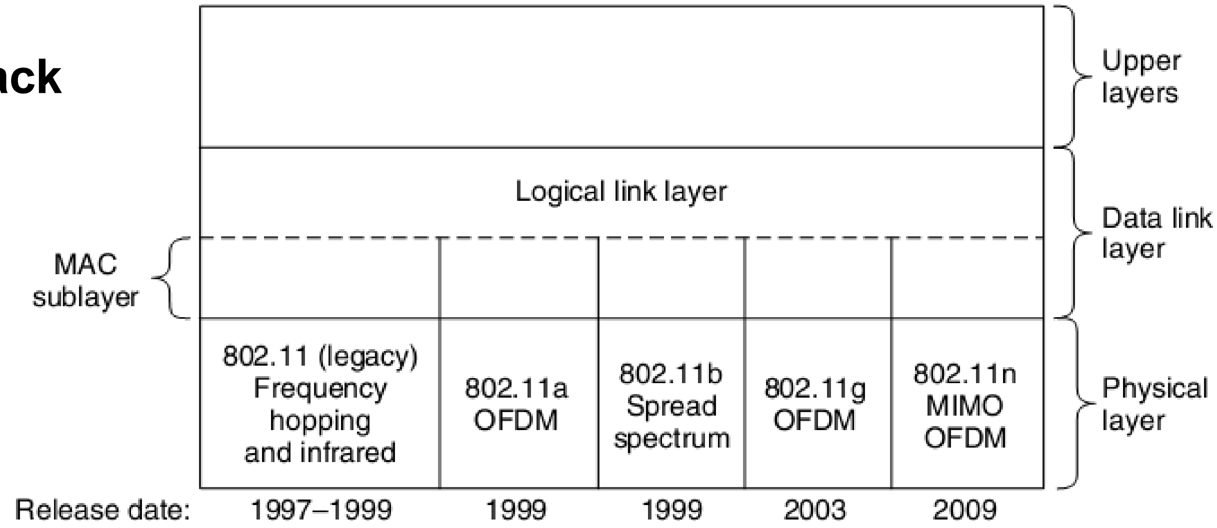
- Wireless network **elements**:
  - Wireless hosts: end devices
  - Base station (access point or AP)
  - Wireless links (*broadcast*)
- Wireless network **types**:
  - *Single-hop, infrastructure-based*: classic WiFi
  - *Single-hop, infrastructure-less*: e.g. WiFi direct, bluetooth
  - *Multi-hop, infrastructure-based*: mesh networks
  - *Multi-hop, infrastructure-less*: e.g. mobile ad hoc networks (MANETs) or vehicular ad hoc network (VANET)



IEEE 802.11 WiFi

IEEE 802.11 Protocol Stack

Most end devices support all variants these days!



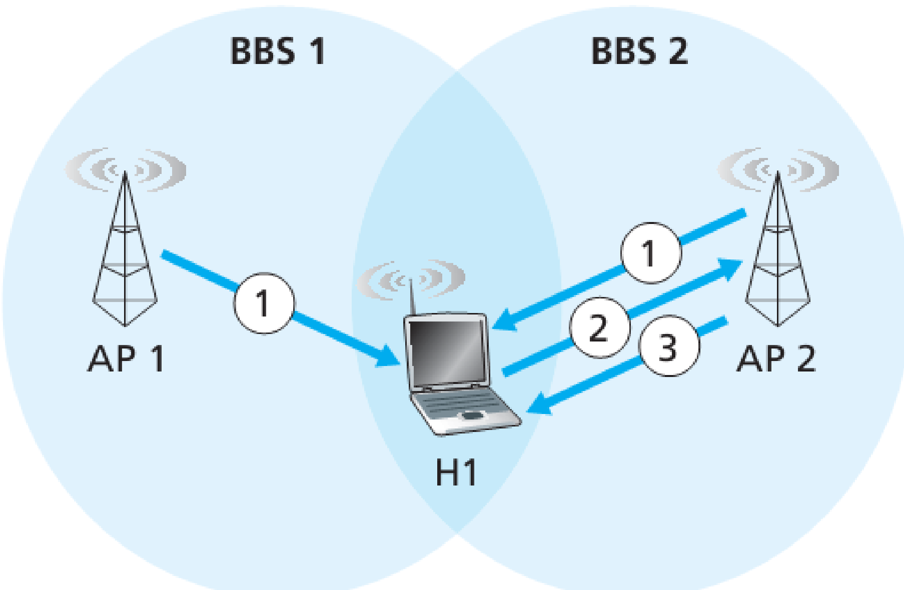
IEEE802.11 Data Frame

Bytes	2	2	6	6	6	2	0–2312	4
	Frame control	Duration	Address 1 (recipient)	Address 2 (transmitter)	Address 3	Sequence	Data	Check sequence

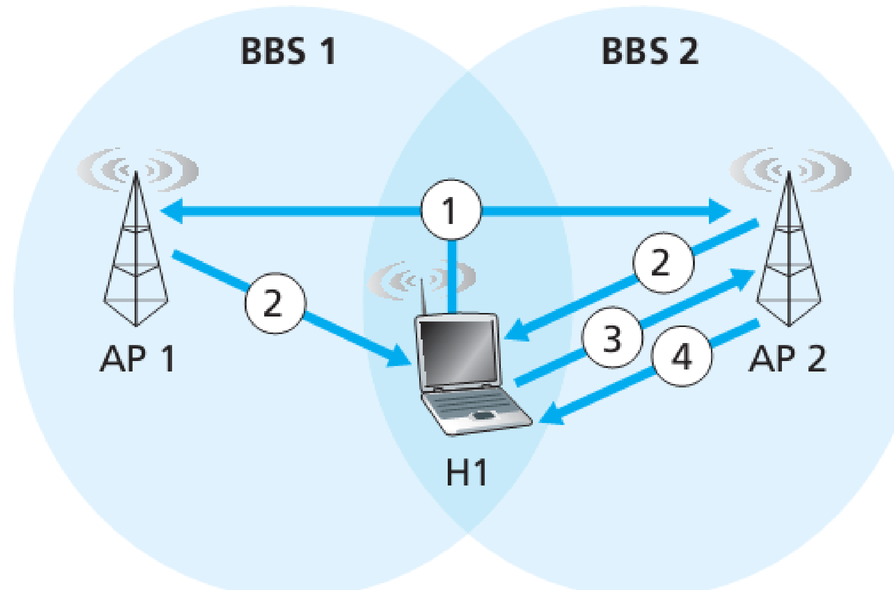
	Version = 00	Type = 10	Subtype = 0000	To DS	From DS	More frag.	Retry	Pwr. mgt.	More data	Protected	Order
Bits	2	2	4	1	1	1	1	1	1	1	1

802.11 Architecture

- Each AP has a one-or two-word **Service Set Identifier (SSID).**
- Each device needs to associate with exactly one of the nearby APs.



a. Passive scanning
1. Beacon frames sent from APs
2. Association Request frame sent: H1 to selected AP
3. Association Response frame sent: Selected AP to H1



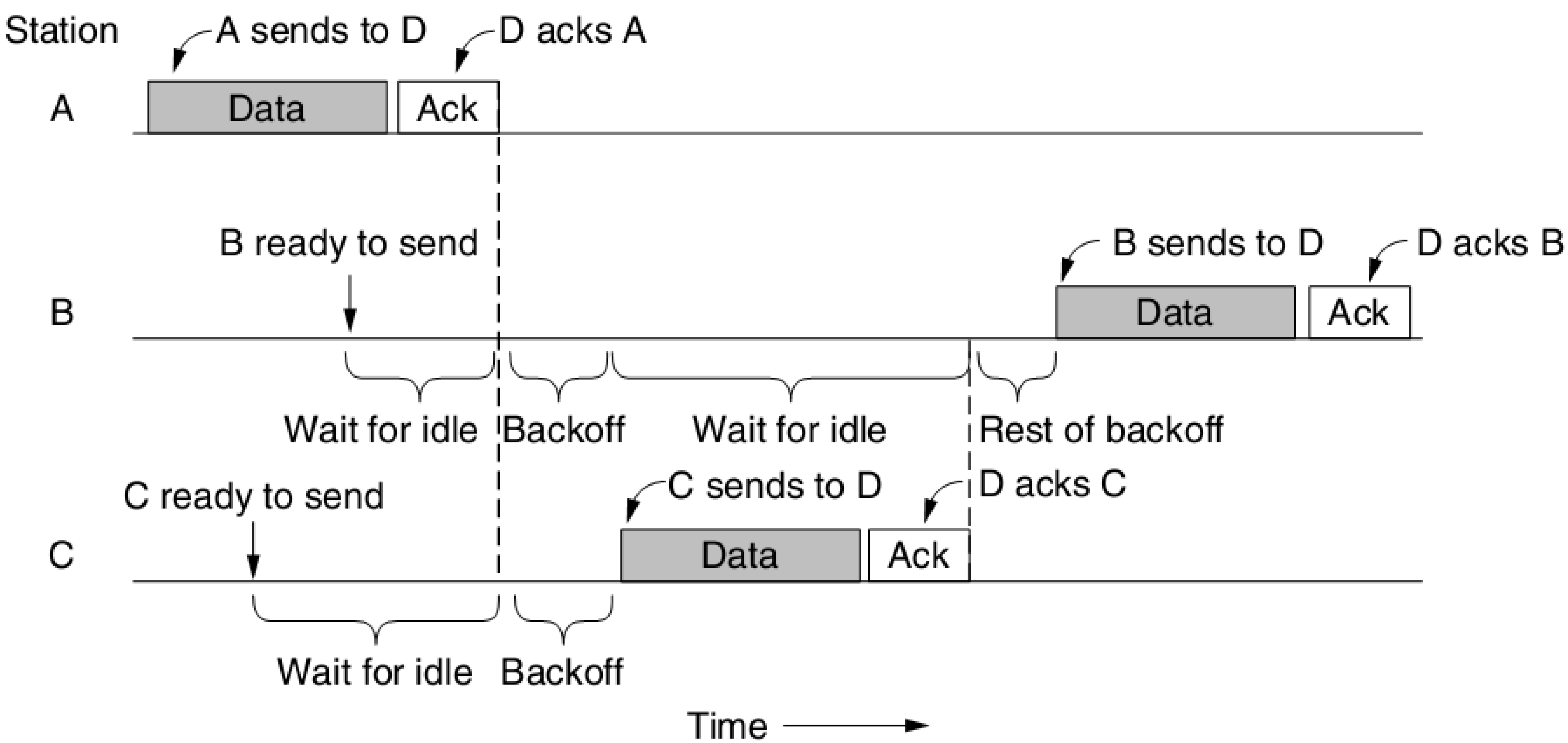
a. Active scanning
1. Probe Request frame broadcast from H1
2. Probes Response frame sent from APs
3. Association Request frame sent: H1 to selected AP
4. Association Response frame sent: Selected AP to H1

CSMA/CA (Collision Avoidance)

802.11 uses **CSMA/CA** (CSMA with Collision Avoidance):

- Channel sensing before sending
- Random back-off
- Sends the frame in full and waits for the ack
- Exponential back-off after collisions.
- It does not sense the channel while transmitting like in CSMA/CD.

CSMA/CA in 802.11 MAC

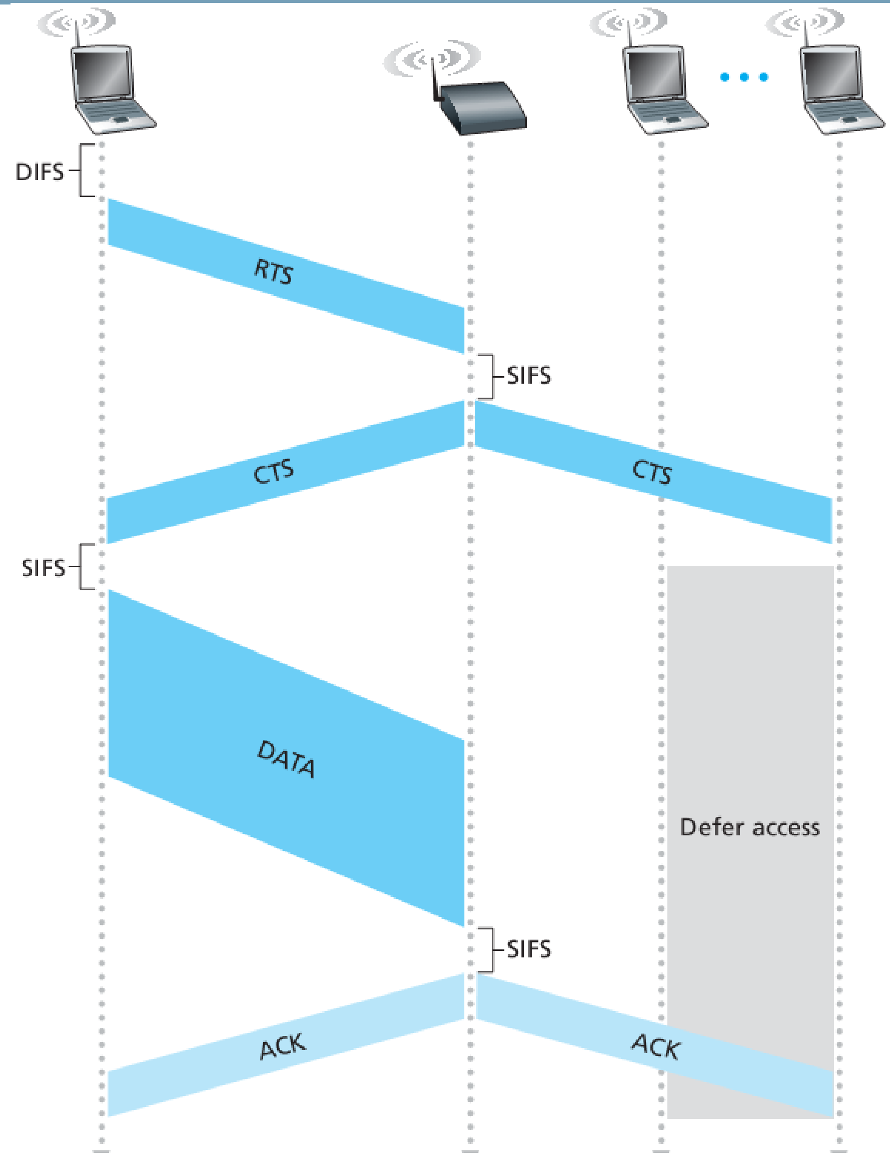


Note the starting backoff and acknowledgements.

802.11 RTS/CTS

- In order to **avoid hidden terminal problem**, 802.11 protocol allows a station to use short **Request to Send (RTS)** and **Clear to Send (CTS)** control frames to reserve access to the channel.
- The CTS frame serves two purposes:
  - *It gives the sender explicit permission to send.*
  - *Instructs the other stations not to send for the reserved duration.*
- RTS/CTS sounds good in theory, but is not used in practice:
  - Only useful for long frames
  - Slows down operation.

Short Inter-frame Spacing (SIFS)



IEEE 802.11 WiFi – new naming

Wi-Fi generations

V · T · E

Gen.[45]	Visual	IEEE standard	Adopt.	Link rate (Mbit/s)	RF (GHz)
Wi-Fi	—	802.11	1997	1–2	2.4
Wi-Fi 1	—	802.11b	1999	1–11	2.4
Wi-Fi 2	—	802.11a	1999	6–54	5
Wi-Fi 3	—	802.11g	2003		2.4
Wi-Fi 4		802.11n	2009	6.5–600	2.4, 5
Wi-Fi 5		802.11ac	2013	6.5–6933	5[b]
Wi-Fi 6		802.11ax	2021	0.4–9608	2.4, 5
Wi-Fi 6E[c]					6
Wi-Fi 7		802.11be	2024[d]	0.4–23,059	2.4, 5, 6
Wi-Fi 8[46][47]	—	802.11bn		100,000	2.4, 5, 6

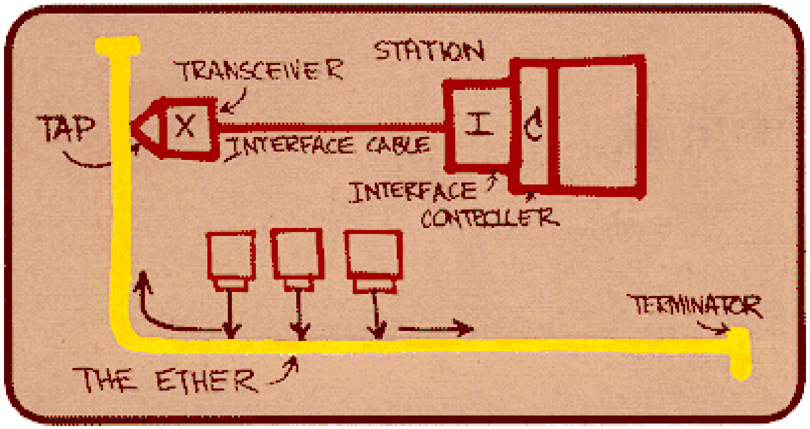
Ethernet

application
transport
network
link
physical

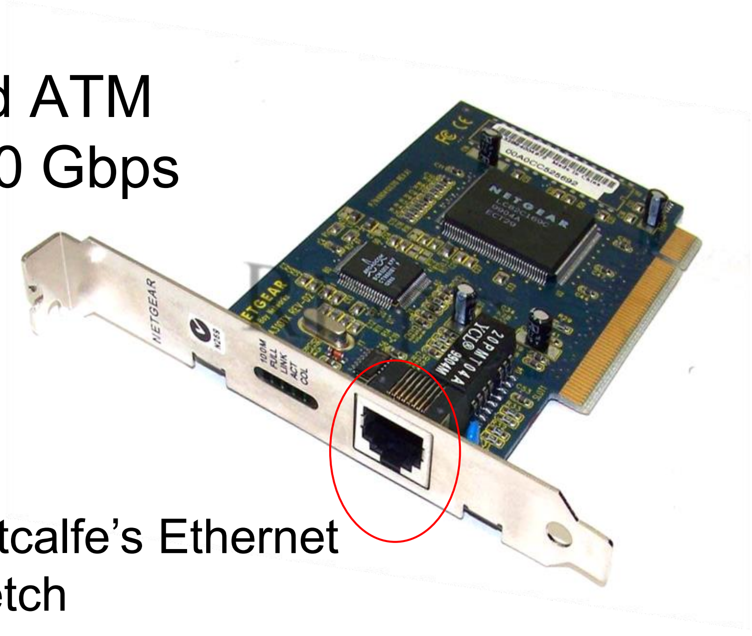
Ethernet, the "dominant" **wired** LAN technology:

- cheap NIC – ***these days embedded to motherboard***
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

IEEE 802.3
https://en.wikipedia.org/wiki/IEEE_802.3



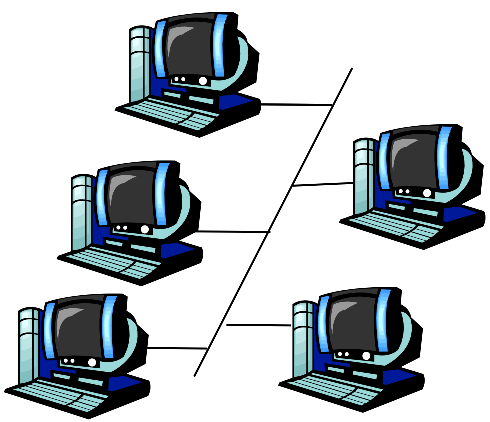
Metcalfe's Ethernet sketch



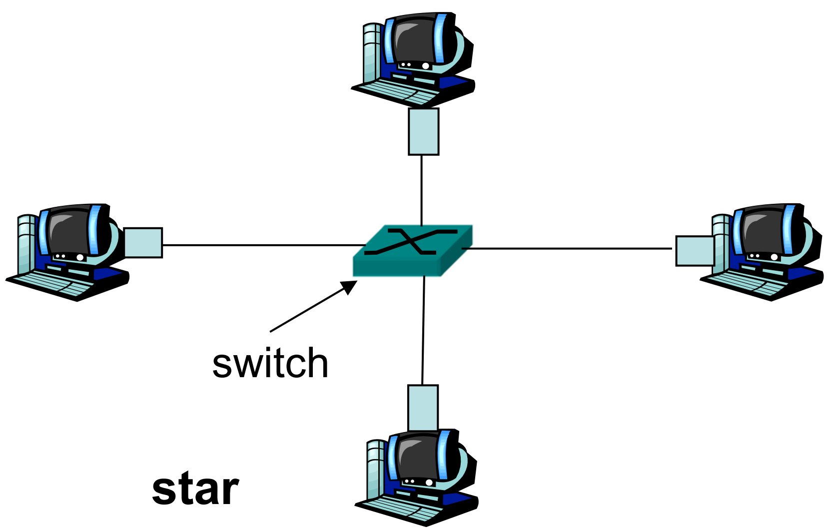
Original paper*: Robert M. Metcalfe and David R. Boggs. 1976. Ethernet: distributed packet switching for local computer networks. Commun. ACM 19, 7 (July 1976), 395-404.*

Star topology

- bus topology popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- today: star topology prevails
  - active *switch* in center
  - each "spoke" ***runs a separate Ethernet protocol*** and the nodes do not collide with each other!



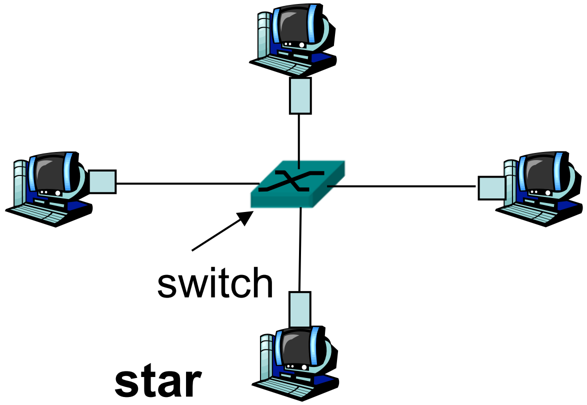
bus: coaxial cable (historical)



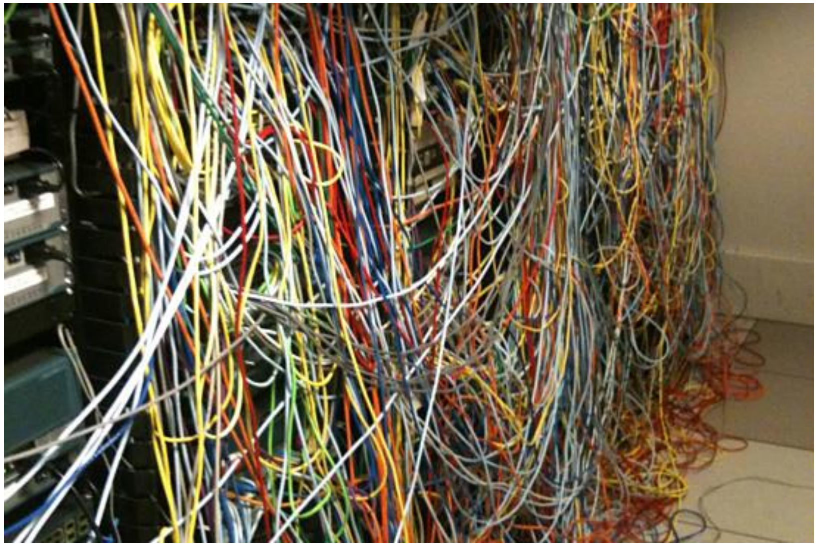
star

Star topology

Q: What is the downside of modern star topology?



Lots and lots of cables!

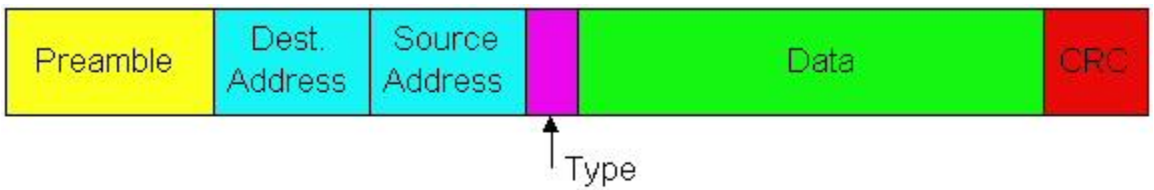


It can become a spaghetti nightmare
if not organised ☺

https://www.itrw.net/2016/06/27/organized-cabling-is-better-cabling-avoid-server-room-spaghetti/

Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

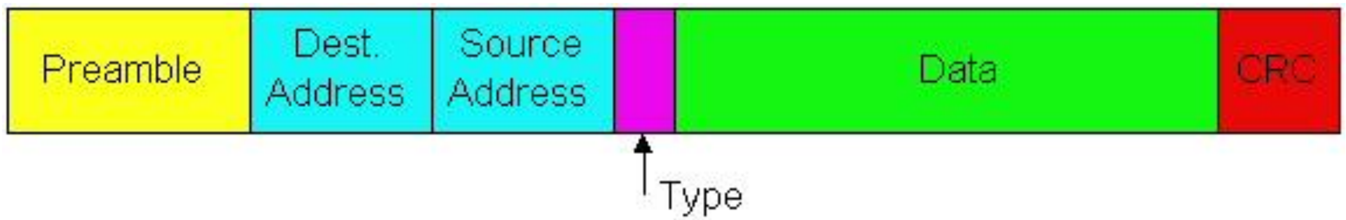


Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- Used to synchronize receiver, sender clock rates

Ethernet Frame Structure

- Addresses: 6 bytes
  - If adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - Otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP)
- CRC: checked at receiver
  - If error is detected, frame is dropped



Ethernet: unreliable, connectionless

- Connectionless: no handshaking between sending and receiving NICs
- Unreliable: receiving NIC doesn't send Acks or Nacks to sending NIC
  - Stream of datagrams passed to network layer can have gaps (missing datagrams)
  - Gaps will be filled if app is using TCP
  - Otherwise, app will see gaps
- Ethernet's MAC protocol (historical):
  - Unslotted CSMA/CD with **binary exponential backoff**
  - **CSMA/CD** was used in now-obsolete shared media Ethernet variants (10BASE5, 10BASE2) and in the early versions of twisted-pair Ethernet which used repeater hubs.

CSMA/CD algorithm (old Ethernet variant)

1. Node receives datagram from network layer, creates frame.
2. If Node senses channel idle, starts frame transmission. If Node senses channel busy, waits until channel idle, then transmits.
3. If Node transmits entire frame without detecting another transmission, Node is done with frame !
4. If Node detects another transmission while transmitting, aborts and sends jam signal.
5. After aborting, Node enters binary **exponential backoff**:
   – after *m*-th collision, Node chooses *K* at random from $\{0,1,2,\ldots,2^m-1\}$. Node waits $K \cdot 512$ bit times, returns to Step 2

CSMA/CD algorithm

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K· 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

Bit transmission time:

0.1 microsec for 10 Mbps Ethernet

Question 1

Question: In a CSMA/CD protocol, the adapter waits K.512 bit transmission times after a collision, where K is a random variable. If K=100, then how long does the adapter wait

1. For a 10Mbps link?
2. For a 100Mbps link?

802.3 Ethernet Standards: Link & Physical Layers

- *Many* different Ethernet variants
  - common MAC protocol and frame format
  - different speeds: 10,100 Mbps, 1,10 Gbps
  - different physical layer media: fiber, cable

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps (Cat 5 UTP)
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

Name	Cable	Max. segment	Advantages
10GBase-SR	Fiber optics	Up to 300 m	Multimode fiber (0.85μ)
10GBase-LR	Fiber optics	10 km	Single-mode fiber (1.3μ)
10GBase-ER	Fiber optics	40 km	Single-mode fiber (1.5μ)
10GBase-CX4	4 Pairs of twinax	15 m	Twinaxial copper
10GBase-T	4 Pairs of UTP	100 m	Category 6a UTP

MTU - Maximum Transmission Unit

- The ***MTU*** *is the maximum payload length for a particular transmission media*.
- The MTU for Ethernet is typically 1500 bytes. That is the maximum payload length including the IP header.
- The MTU for WI-Fi is also typically 1500 bytes, to be compatible with Ethernet.
- If a host wishes to send packet larger than the MTU for a network, the packet must be broken up into chunks no larger than the MTU (fragmentation).
- The smallest MTU between two hosts is known as the **path MTU**.

MTU - Maximum Transmission Unit

Table from: RFC 1191, Path MTU Discovery, November 1990

See also: http://wiki.wireshark.org/MTU

MTU	Comments	Reference
65535	Official maximum MTU	RFC 791
65535	Hyperchannel	RFC 1044
	Just in case	
17914	16Mb IBM Token Ring	ref. [6]
8166	IEEE 802.4	RFC 1042
4464	IEEE 802.5 (4Mb max)	RFC 1042
4352	FDDI (Revised)	RFC 1188
2048	Wideband Network	RFC 907
2002	IEEE 802.5 (4Mb recommended)	RFC 1042
1536	Exp. Ethernet Nets	RFC 895
1500	Ethernet Networks	RFC 894
1500	Point-to-Point (default)	RFC 1134
1492	IEEE 802.3	RFC 1042
1006	SLIP	RFC 1055
1006	ARPANET	BBN 1822
576	X.25 Networks	RFC 877
544	DEC IP Portal	ref. [10]
512	NETBIOS	RFC 1088
508	IEEE 802/Source-Rt Bridge	RFC 1042
508	ARCNET	RFC 1051
296	Point-to-Point (low delay)	RFC 1144
	Official minimum MTU	RFC 791

Table 7-1: Common MTUs in the Internet

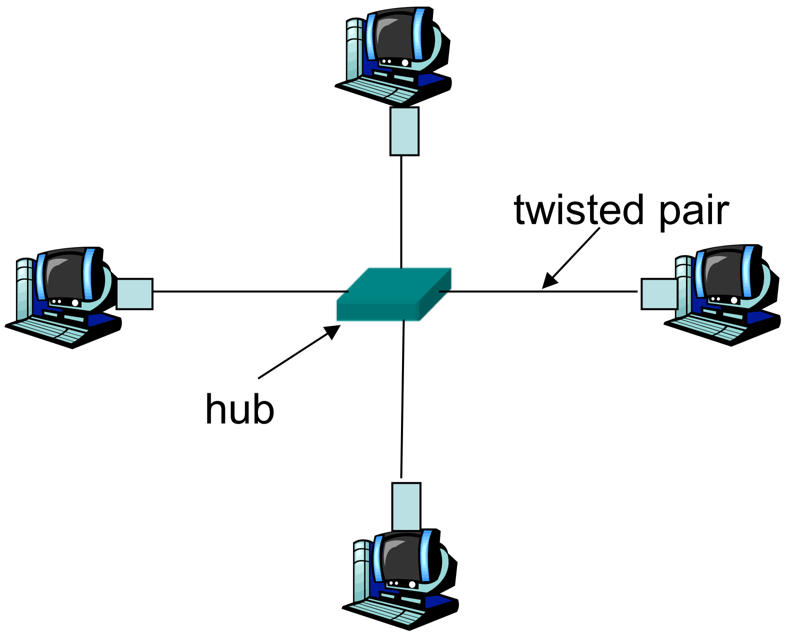
Hubs, Bridges, and Switches

application
transport
network
link
physical

Hubs (historic)

Hubs are ***physical-layer*** (“dumb”) repeaters:

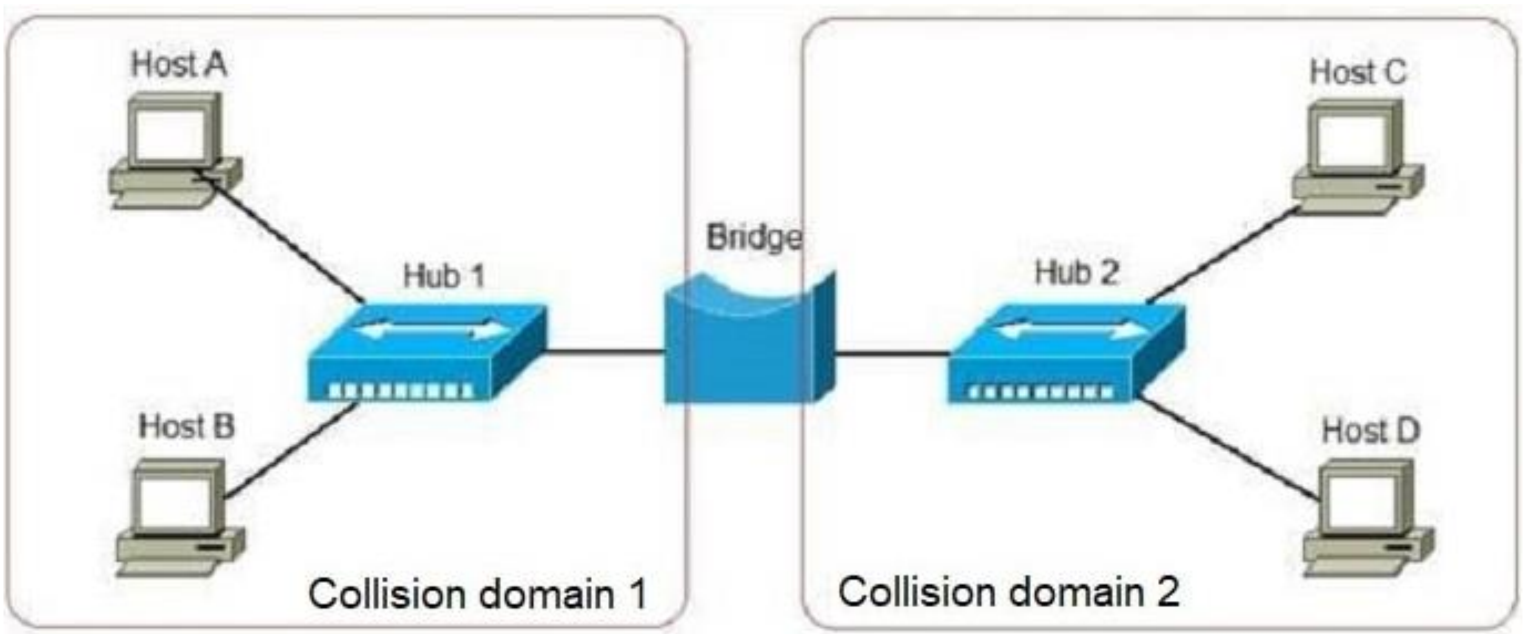
- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub share the same collision domain
- no frame buffering
- Wasted bandwidth



Bridge (historic)

Network Bridge connects two network segments.

- **Each segment is a separate collision domain**
- Maintains a MAC address table
- Different variants support various functions.

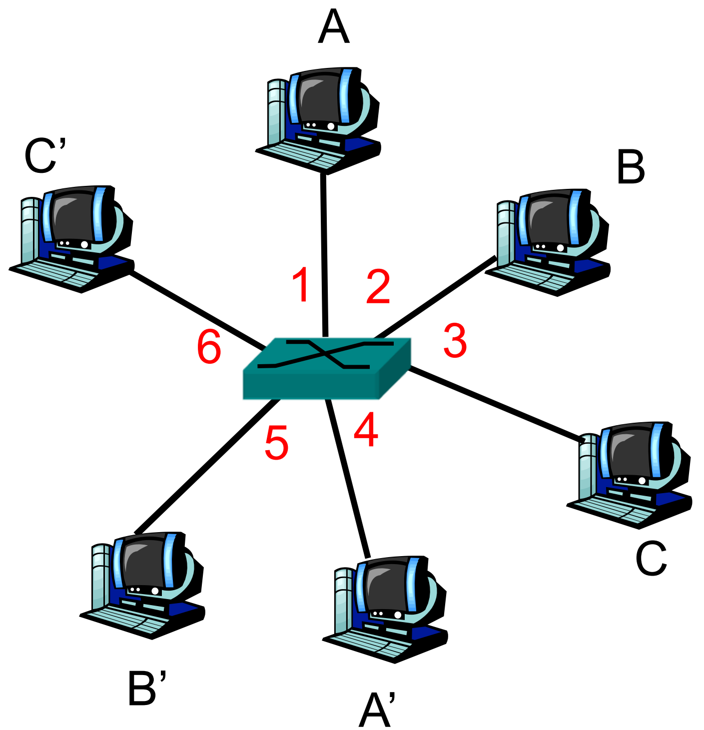


Ethernet Switch

- Link-layer device: smarter than hubs, takes *active* role
  - store, forward link frames
  - examine incoming frame's MAC address, **selectively** forward frame to one-or-more outgoing links when frame is to be forwarded on segment,
- *Transparent*
  - hosts are unaware of presence of switches
- *Plug-and-play, self-learning*
  - switches do not need to be configured

Switch: *multiple* simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- Switches buffer packets
- *Switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with simple hub



switch with six interfaces (1,2,3,4,5,6)

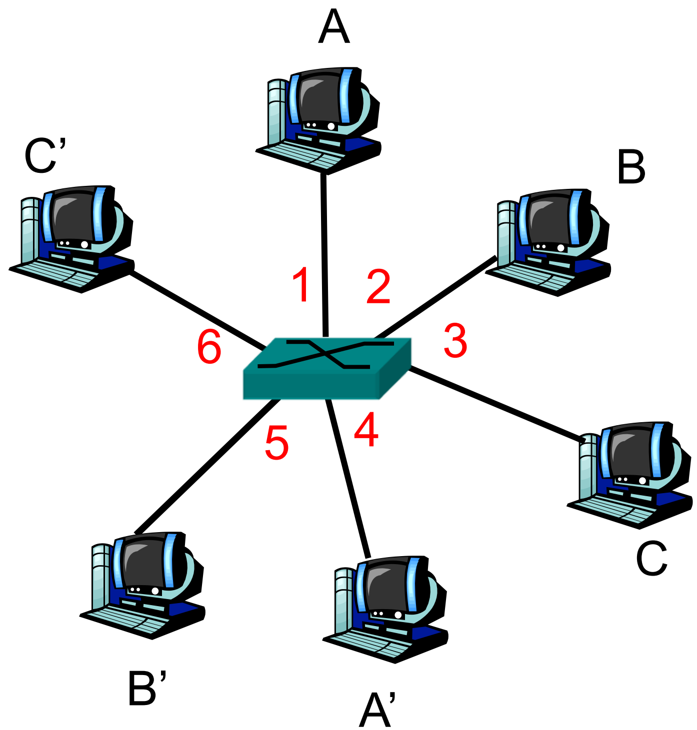
Switch Table

- ***Q:*** how does switch know that A' reachable via interface 4, B' reachable via interface 5?

- ***A:*** each switch has a switch table, where each entry:
  - (MAC address of host, interface to reach host, time stamp)

  It looks like a routing table!

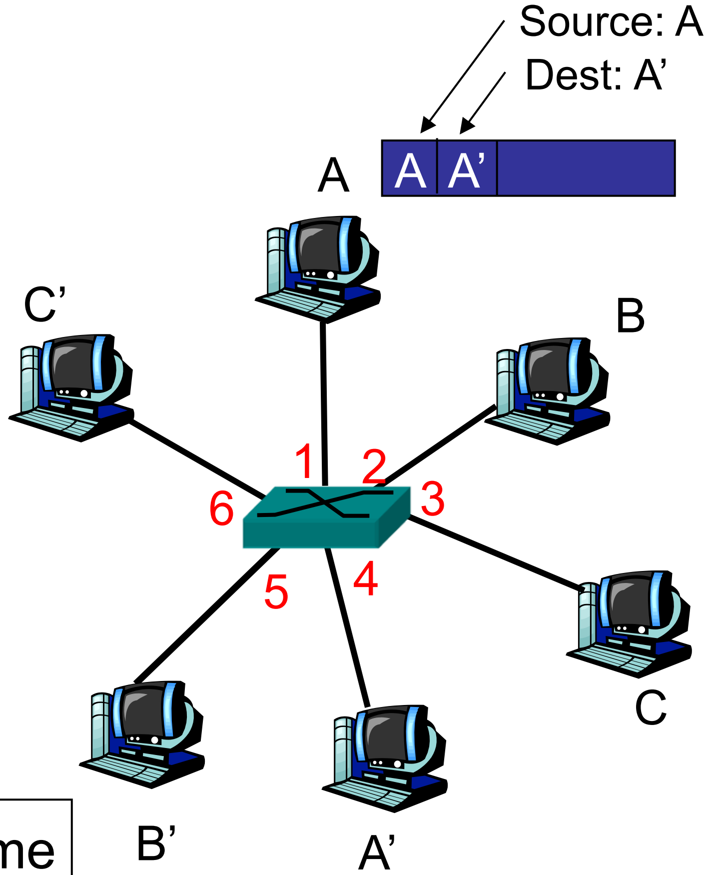
- ***Q:*** how are entries created, maintained in switch table?
  - something like a routing protocol?



switch with six interfaces (1,2,3,4,5,6)

Switch: self-learning

- Switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table



Switch table (initially empty)

MAC addr	interface	Time
A	1	60

Switch: frame filtering/forwarding

When frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. **if** entry found for destination

```
 {
 if dest on segment from which frame arrived
       drop frame     % filtering function
   else                  % forwarding function
       forward frame on interface indicated by entry
}
else                    % forwarding function
   flood
```

forward on all interfaces except the interface on which the frame arrived

Self-learning, forwarding: example

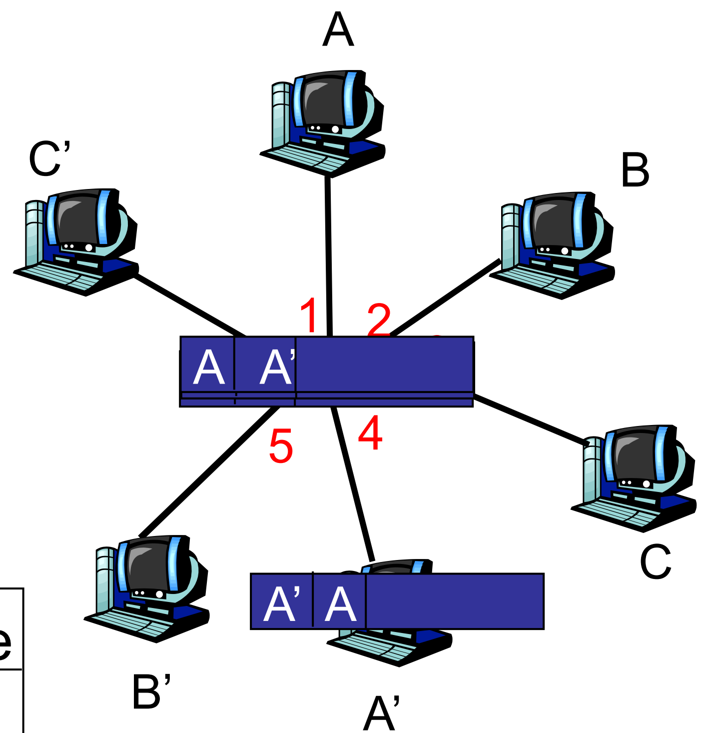
- Frame destination A’ unknown:

  flood

- Intended receiver responds: A’

- For subsequent frames destination A location known:

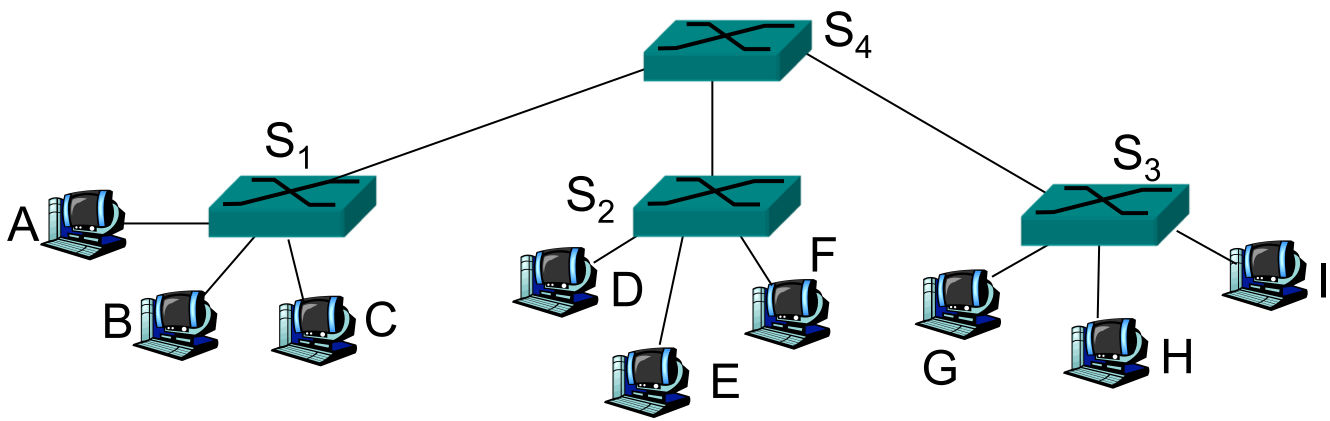
  selective send on one link



MAC addr	interface	Time
A	1	60
A’	4	60

Interconnecting switches

The switches can be connected together:



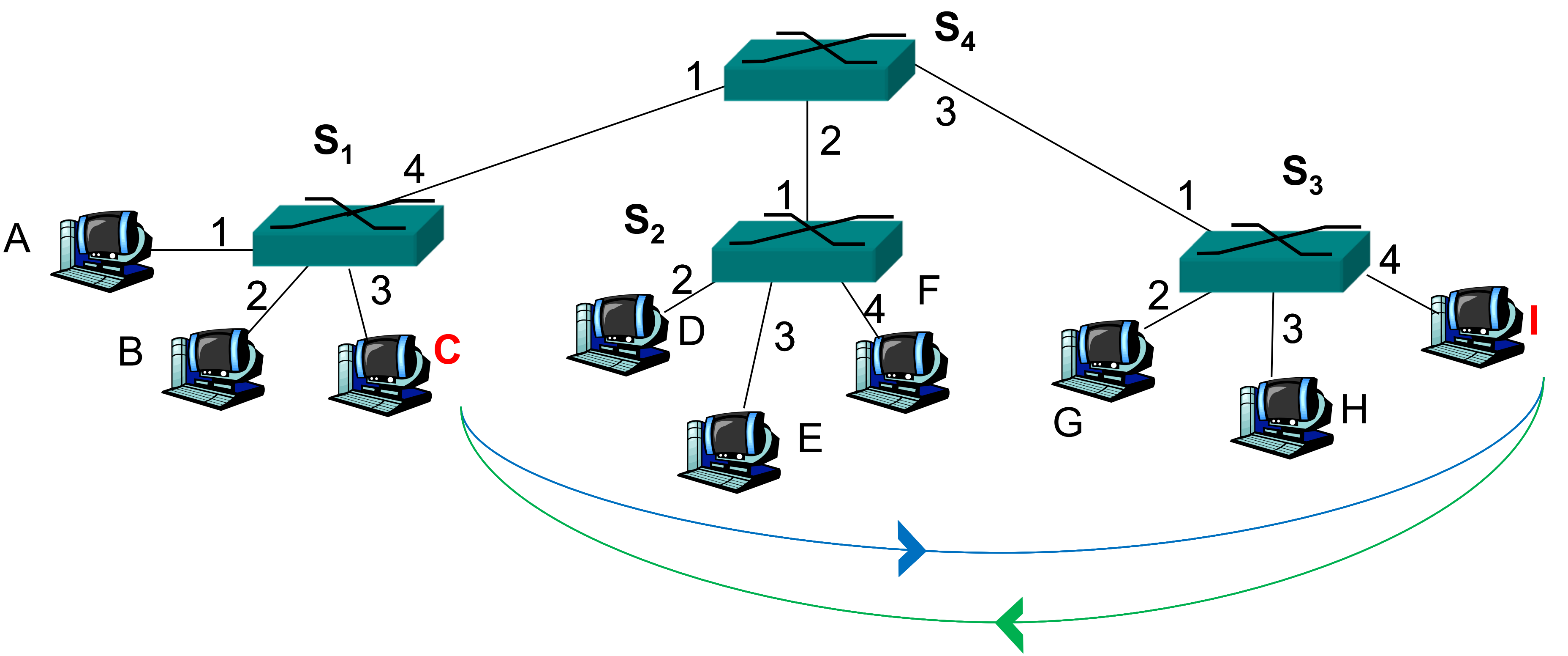
Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

A: self learning! (works *exactly* as in single-switch case!)

- Active topology restricted to spanning tree – otherwise get broadcast loops!
- Looks convenient, why not have bigger LANs?
  - risk of broadcast storms
  - routing inflexibility (no loops, no alternative paths)
  - no network isolation

Self-learning multi-switch example

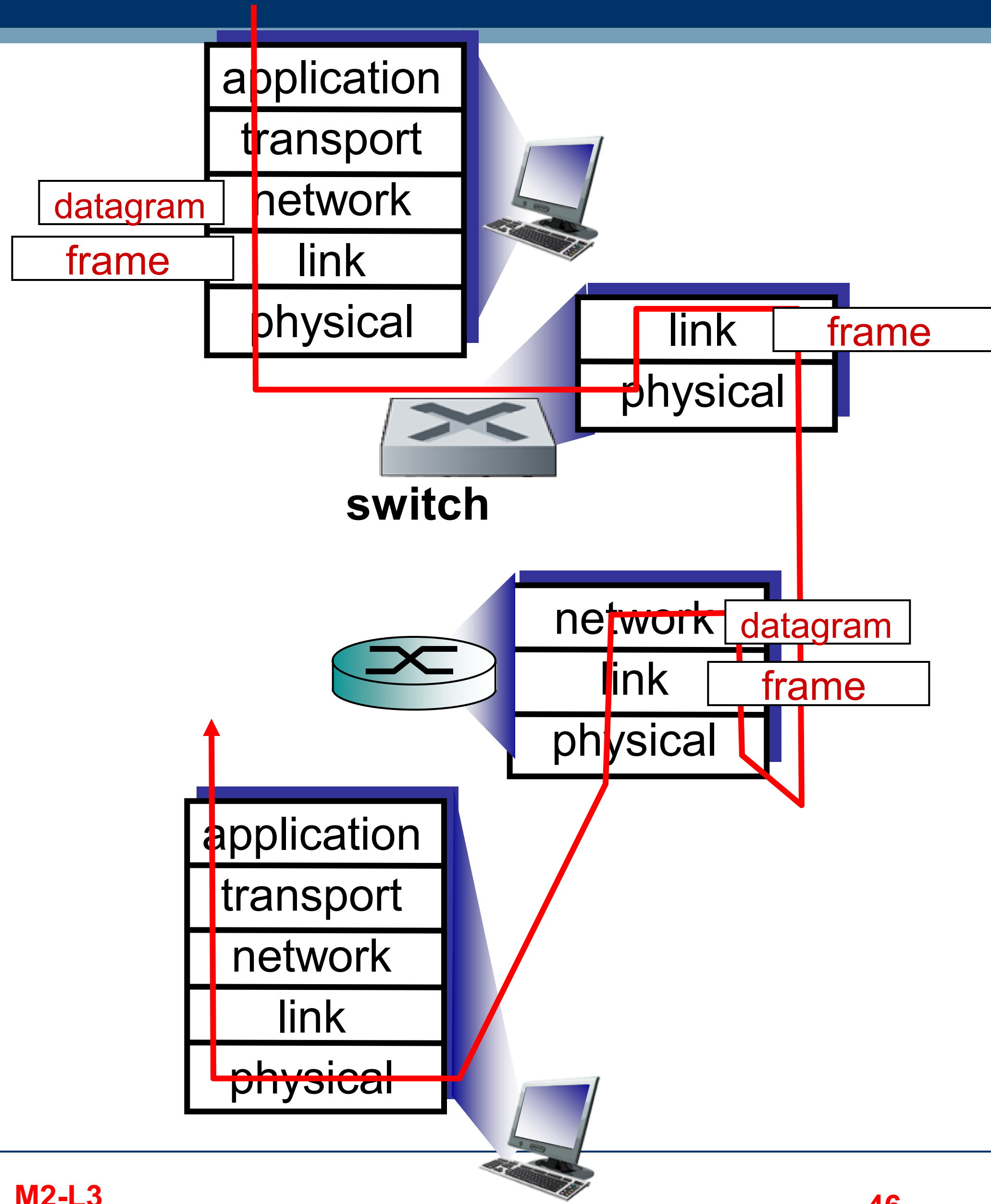
Suppose C sends a frame to I and I responds to C



Q: show switch tables in S_1, S_2, S_3, S_4 .

Switches vs. routers

- Both are store-and-forward:
  - *routers:* network-layer devices (examine network-layer headers)
  - *switches:* link-layer devices (examine link-layer headers)
- Both have forwarding tables:
  - *routers:* compute tables using routing algorithms, IP addresses
  - *switches:* learn forwarding table using flooding, learning, MAC addresses

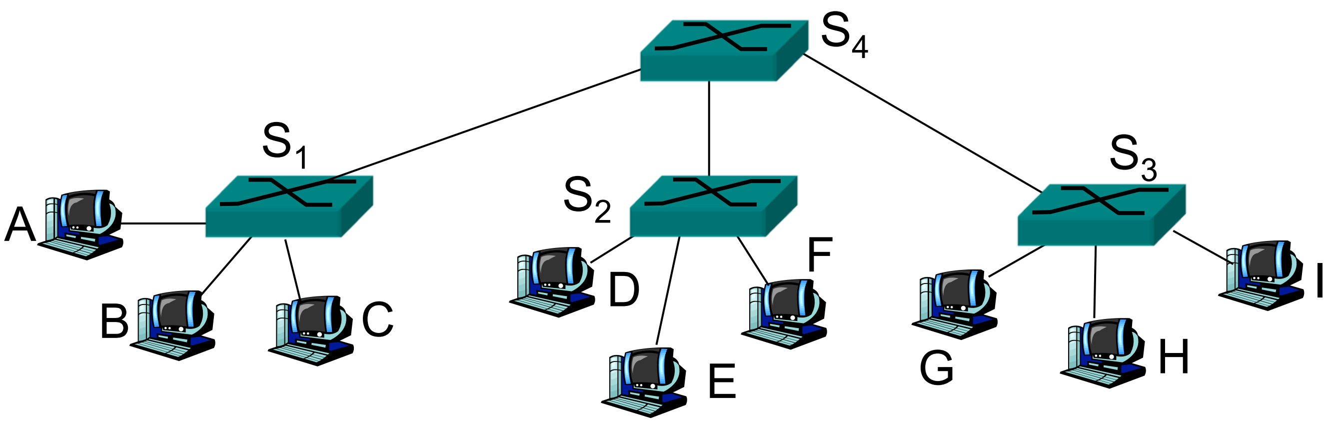


THE UNIVERSITY OF MELBOURNE

Virtual LAN (VLAN)

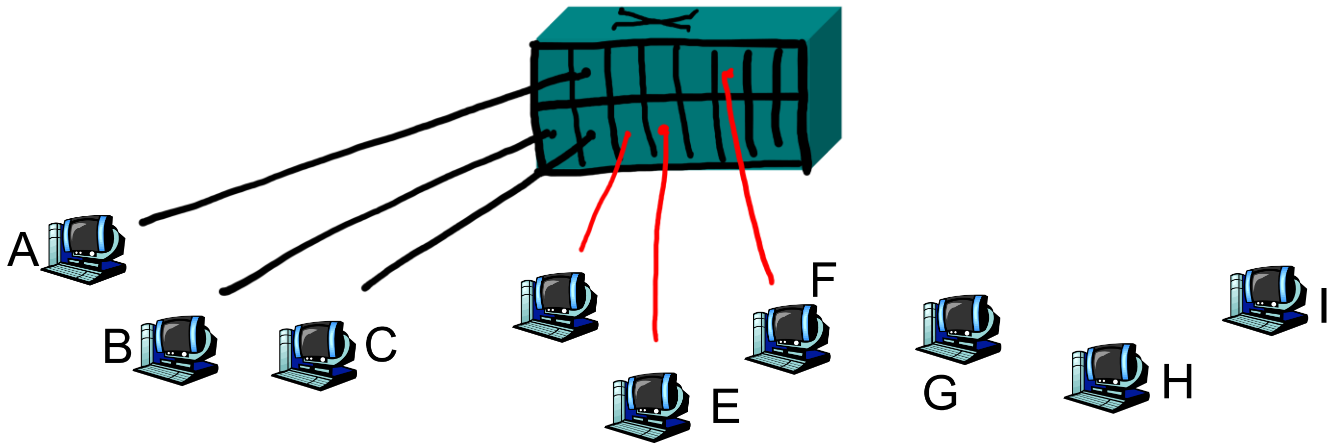
application
transport
network
link
physical

Virtual LANs (VLANs)



- Lack of traffic isolation
- Broadcast storms
- Inefficient use of switches
- Difficulty in managing users

Virtual LANs (VLANs)



Multiple Virtual LANs defined over a single physical LAN infrastructure

- Hosts within a VLAN communicate as if they are connected to a switch
- Separate broadcast domains
- You need a router for the departments to communicate with each other

THE UNIVERSITY OF MELBOURNE

Delay, Loss, and Throughput

application
transport
network
link
physical

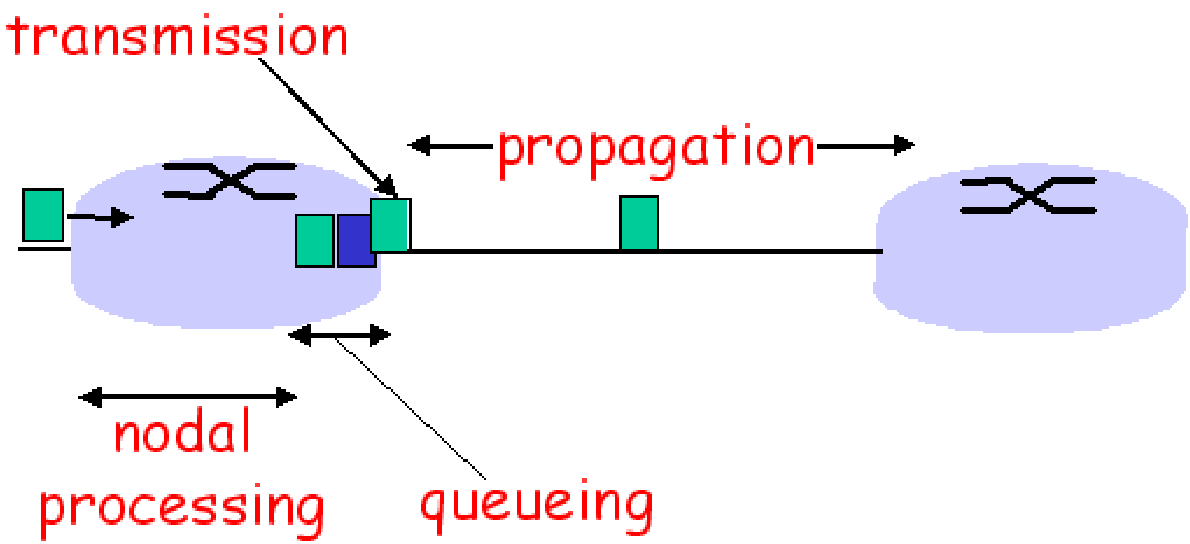
Four sources of packet delay

1. Processing delay:

- Examine header
- check for bit errors
- determine output link

2. Queueing delay:

- time waiting at output link for transmission
- depends on congestion level of router



Delay in packet-switched networks

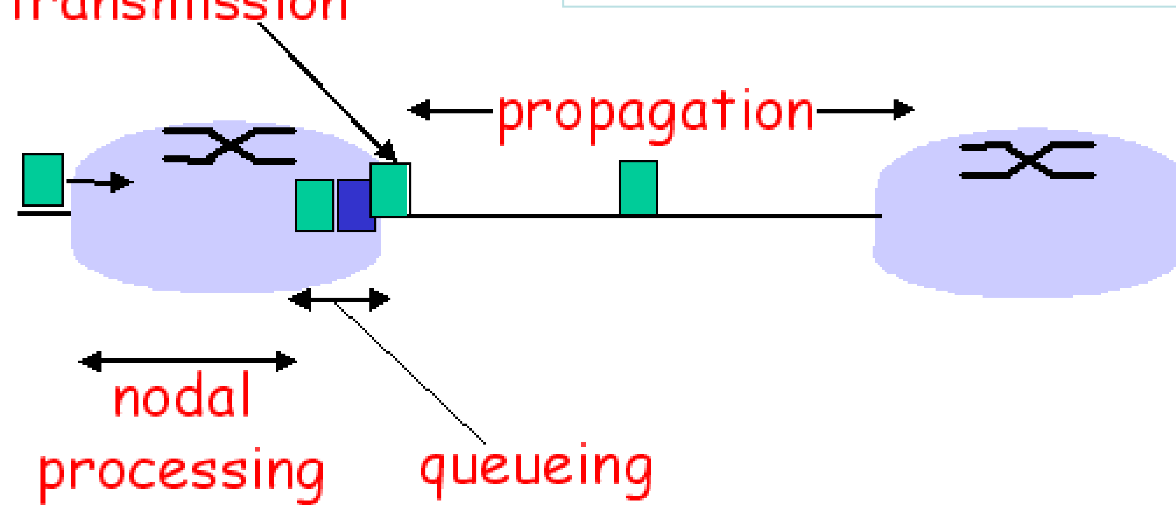
3. Transmission delay:

- R=link bandwidth (bps)
- L=packet length (bits)
- time to send bits into link = L/R

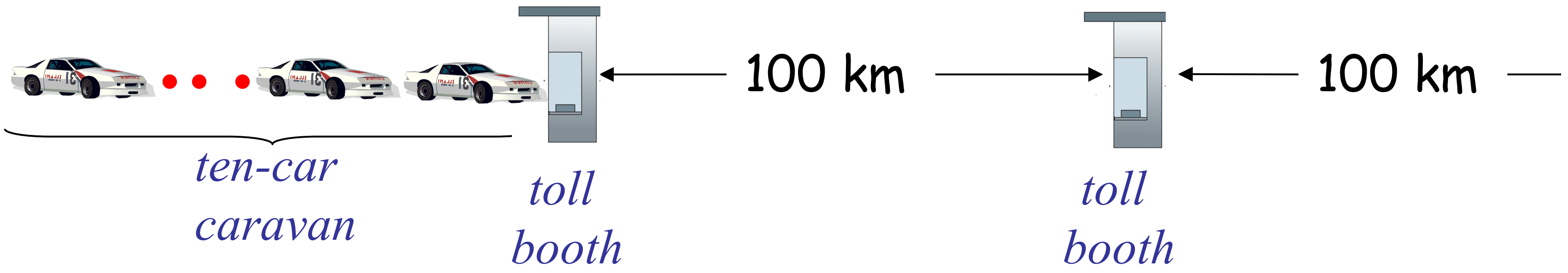
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (2~3x108 m/sec)
- propagation delay = d/s

Note: s and R are *very* different quantities!



Caravan analogy



- Car~bit; caravan~packet
- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service car (transmission time)
- Q: How long until caravan is lined up before 2nd toll booth?

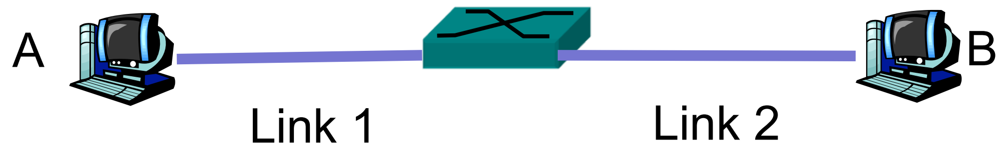
- Time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- A: 62 minutes

Nodal delay

$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$

- d_{proc} = processing delay
  - typically, a few micro-seconds or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few micro-seconds to hundreds of milli-seconds

Question 2

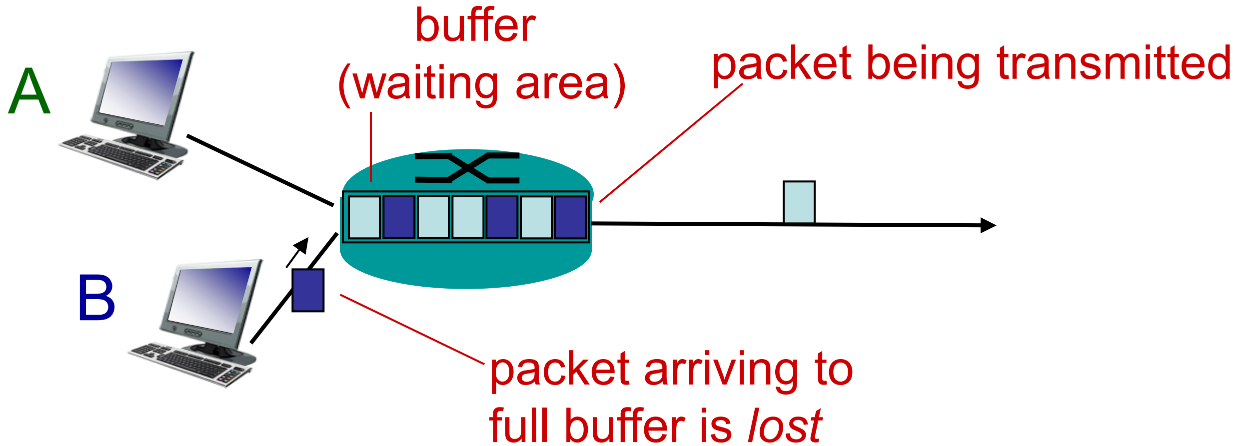


- Consider a packet of length L which begins at end system A, travels over one link to a packet switch, and travels from the packet switch over a second link to a destination end system B.
- Let d_i, s_i, and R_i denote the length, propagation speed, and the transmission rate of link i, for i = 1,2. The packet switch delays each packet by d_{proc} due to processing.

1. Provide the formula for the total end-to-end delay experienced by a packet in this system.
2. Suppose the packet size is 1000 bytes, the propagation speed on both link is 2.5×10^8m/s, the transmission rate of both link is 1Mbps, the packet processing delay is 1msec, the length of the first link is 4000km and the length of the second link is 1000km. For these values, what is the end-to-end delay?

Packet loss

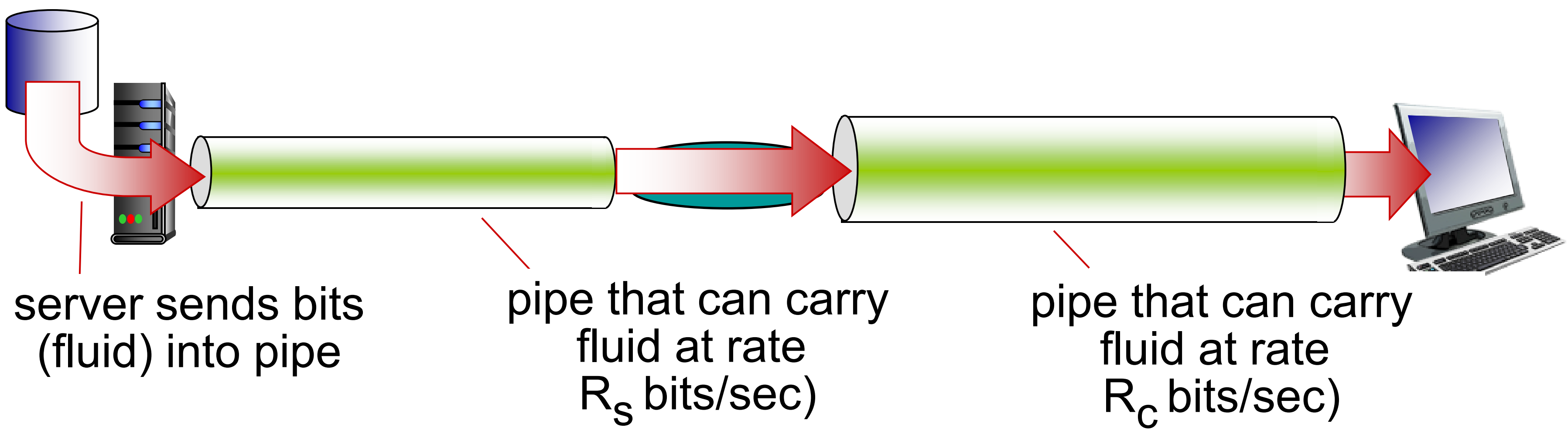
- Queue (aka buffer) preceding link has finite capacity
- Packet arriving to full buffer dropped/lost
- Lost packet may be retransmitted by previous node, by source end system.



Throughput

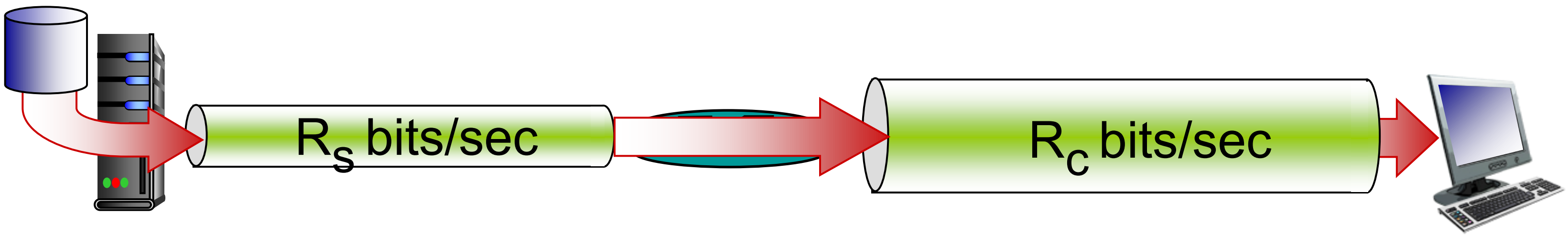
Throughput: rate (bits/time unit) at which bits transferred between sender/receiver

- *instantaneous:* rate at given point in time
- *average:* rate over longer period of time

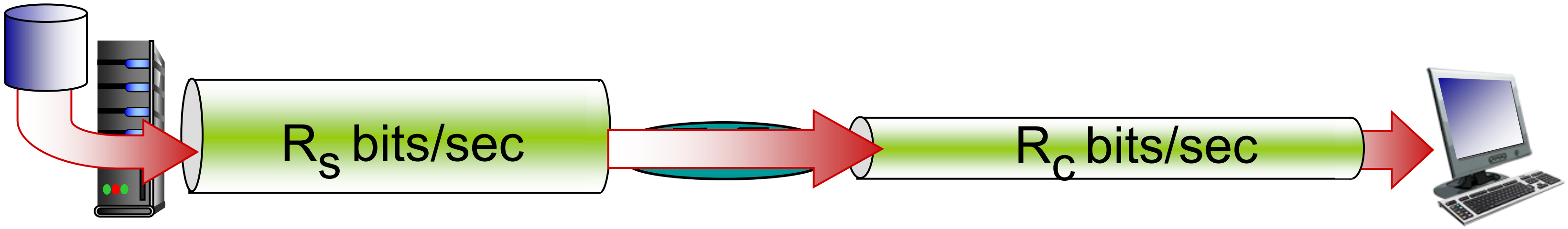


Throughput bottleneck

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?



bottleneck link

link on path that constrains end-to-end throughput