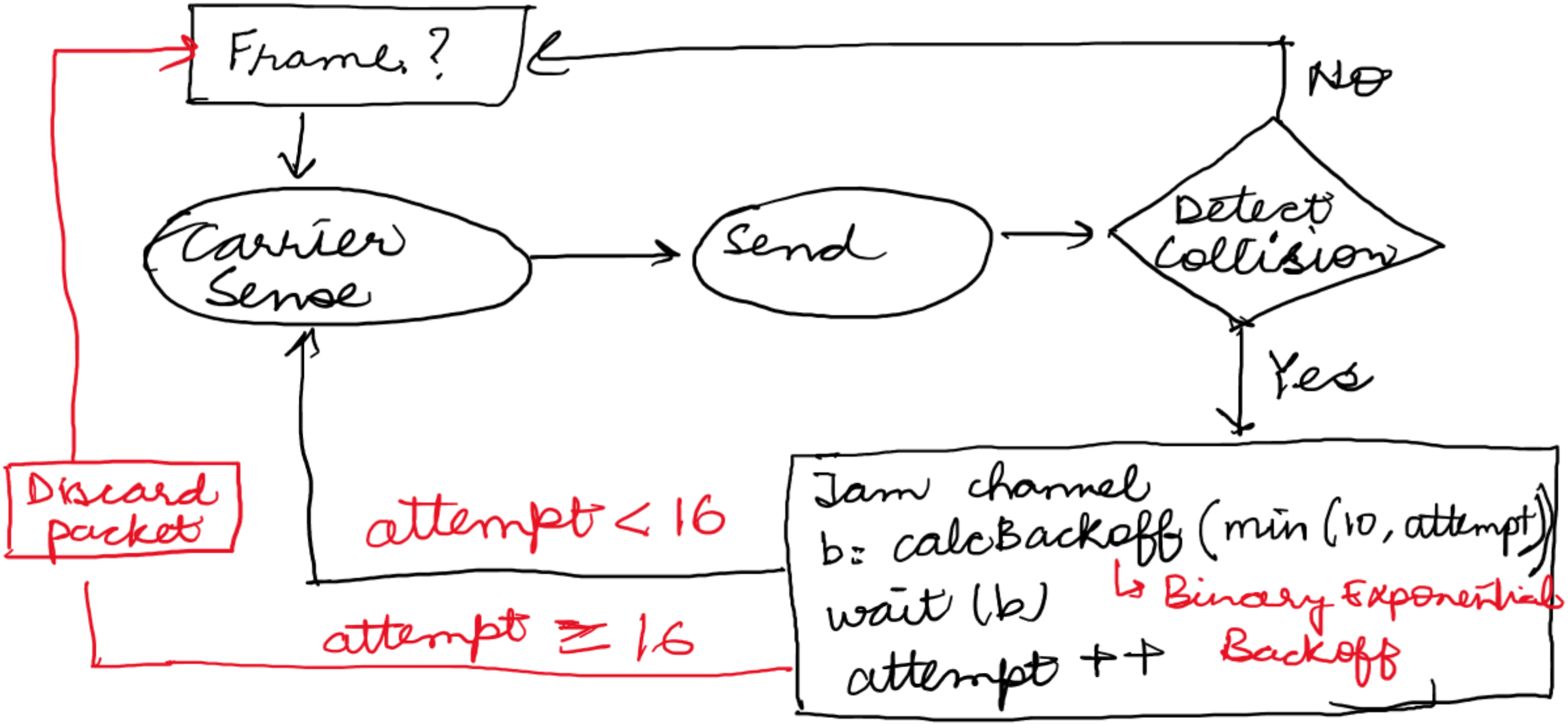
Computer Networks
COL 334/672

Medium Access Control

Slides adapted from KR

Sem 1, 2025-26

State Diagram for Ethernet MAC



Summary: Ethernet Frame Structure

802.3 Ethernet packet and frame structure

Layer	Preamble	Start frame delimiter (SFD)	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap (IPG)
Length (octets)	7	1	6	6	(4)	2	42–1500[c]	4	12
Layer 2 Ethernet frame	(not part of the frame)		← 64–1522 octets →						(not part of the frame)
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

Framing ←
[check for yourself how Ethernet does' it

Can you detect headers related to link layer functions?

- Framing
- Error detection
- Link access

CSMA/Collision Avoidance (CA): MAC for wireless network

- Two challenges
  - Detecting collisions
  - Hidden terminal problem

- How to know packet has been correctly transmitted?
  - Rely on acknowledgements, no ack ➔ loss
  - But it is slow

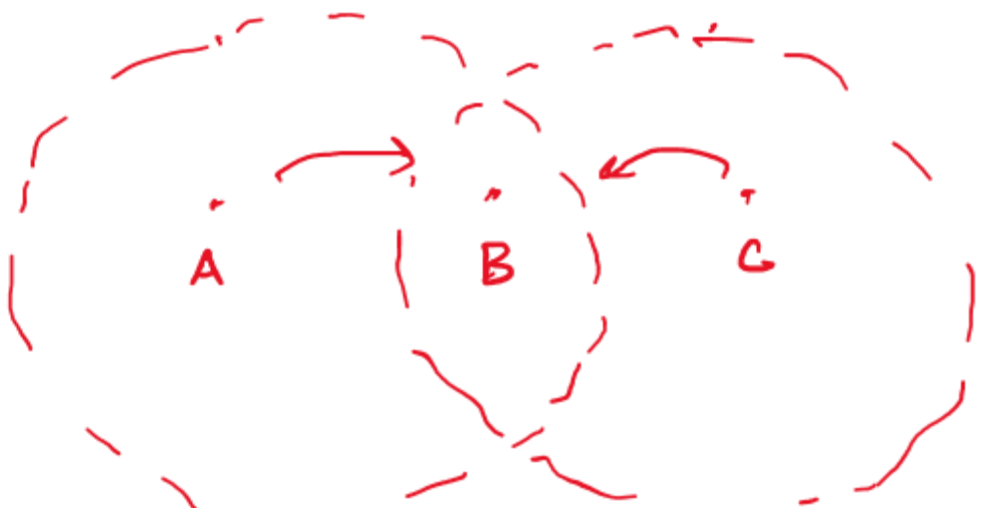
- ***Can we do better?***

$P_{Recv} \propto 1/r^2$

Transmission power at antenna is much higher to detect any collision.

Transmission power → 100 mW

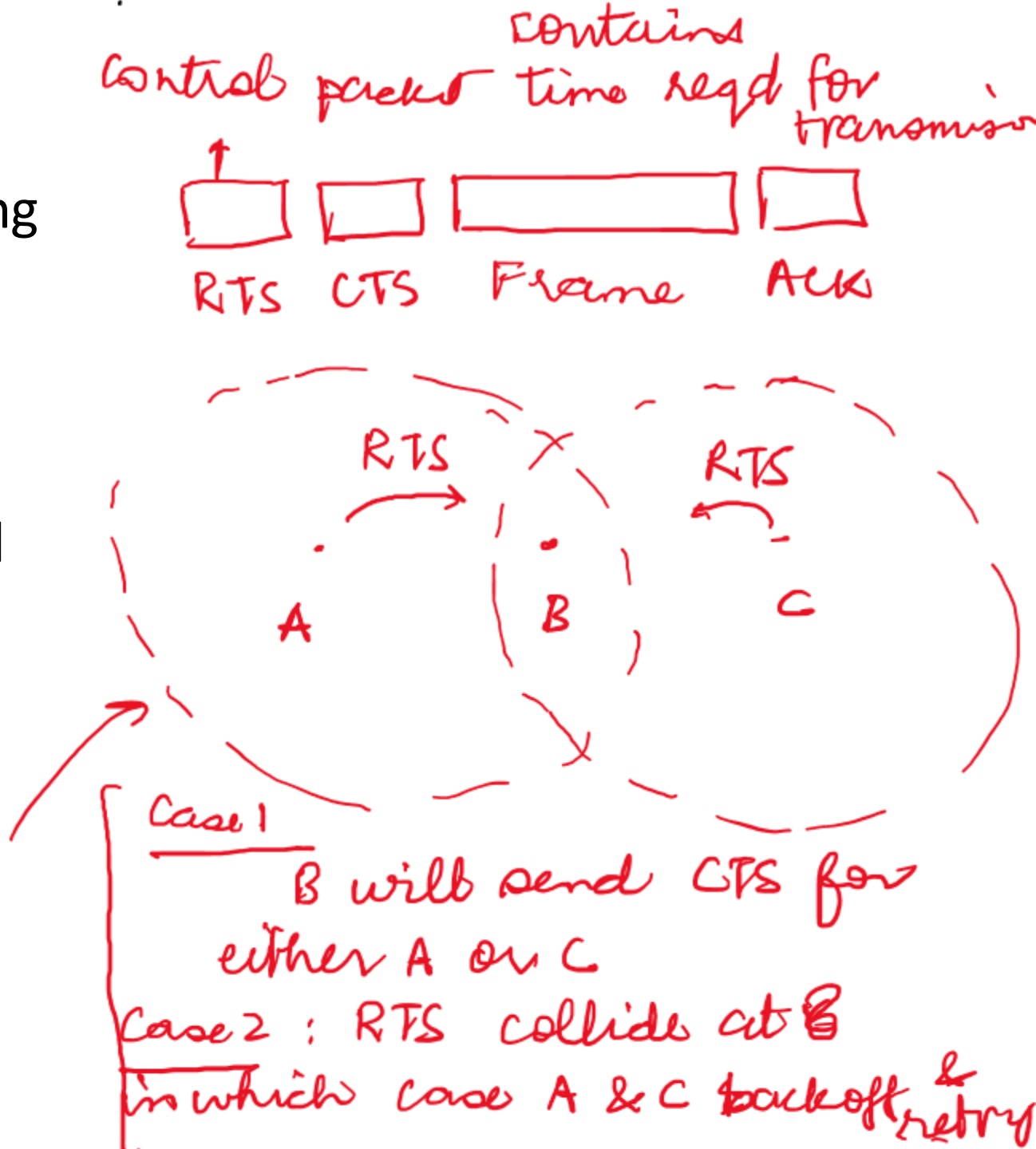
Receive Power : 10^{-9} mW



A & C won't detect each other even if they do carrier sensing

CSMA/CA

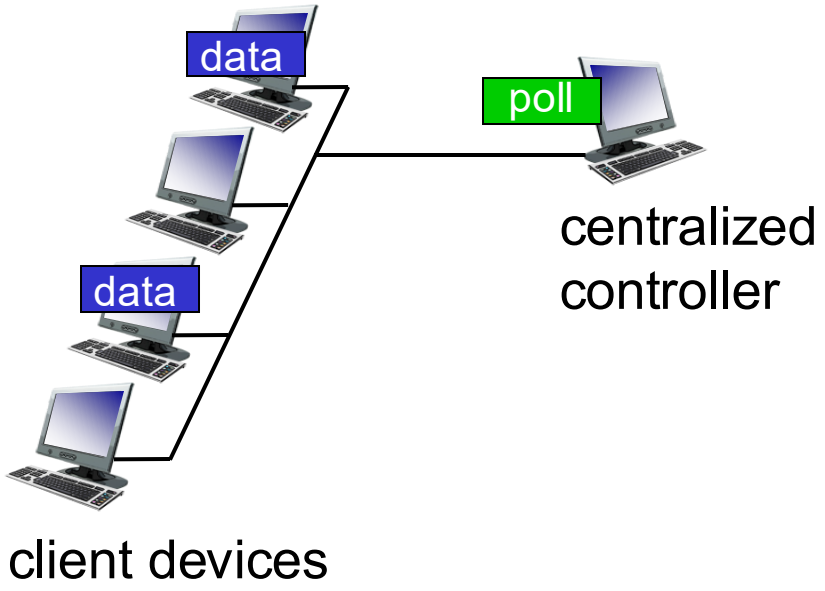
- Use control frames before sending data frames
  - Request To Send (RTS)
  - Clear To Send (CTS)
- Only transmit if CTS is received
- Any node that hears RTS/CTS will remain silent for some duration
- Duration specified in RTS/CTS frames known as **Network Allocation Vector (NAV)**
- Does this solve hidden terminal problem?



“Taking turns” MAC protocols

polling:

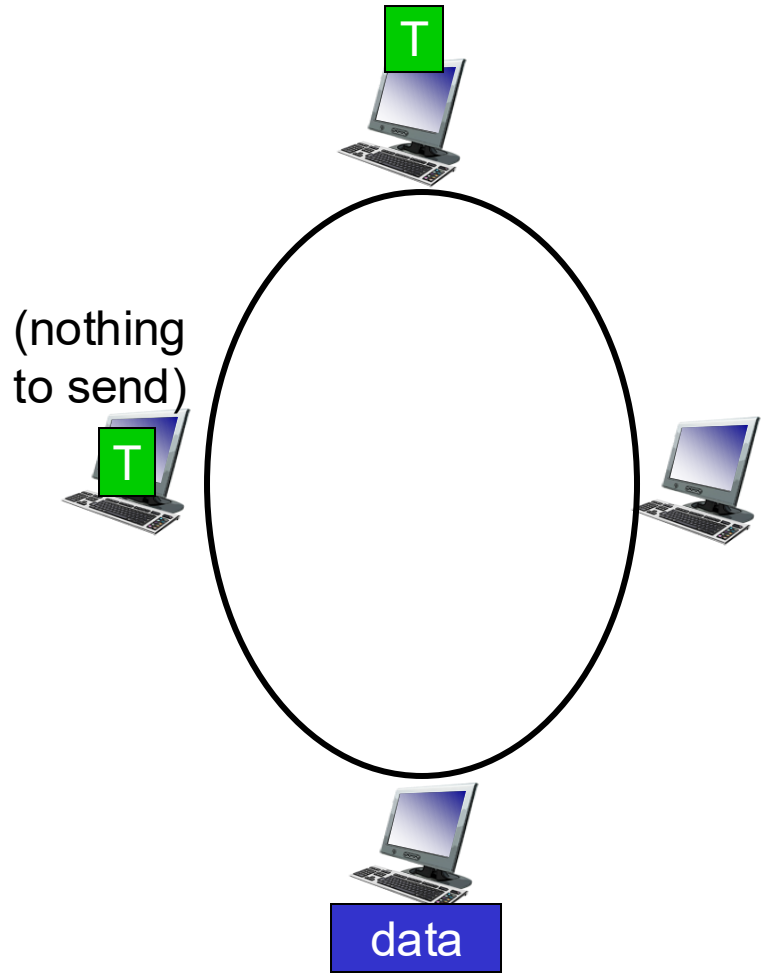
- centralized controller “invites” other nodes to transmit in turn
- typically used with “dumb” devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)
- Bluetooth uses polling



“Taking turns” MAC protocols

token passing:

- control *token* message explicitly passed from one node to next, sequentially
  - transmit while holding token
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



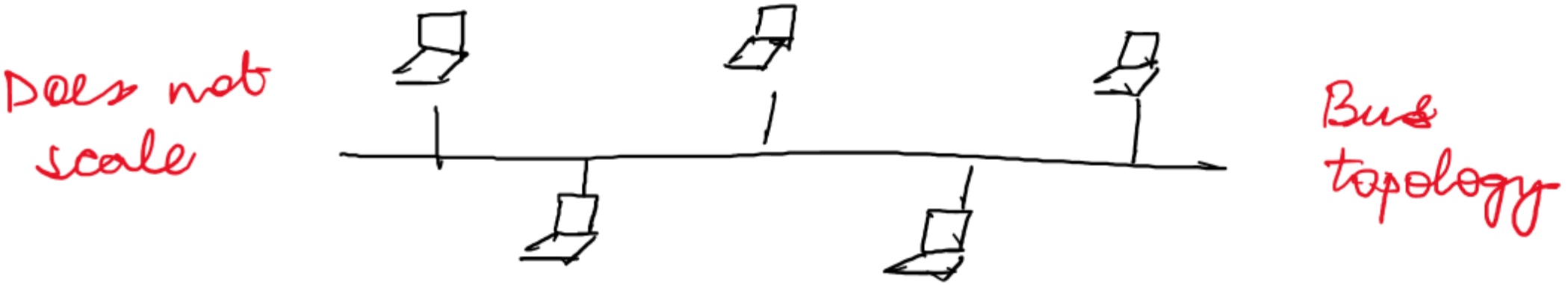
MAC protocols: summary

three broad classes:

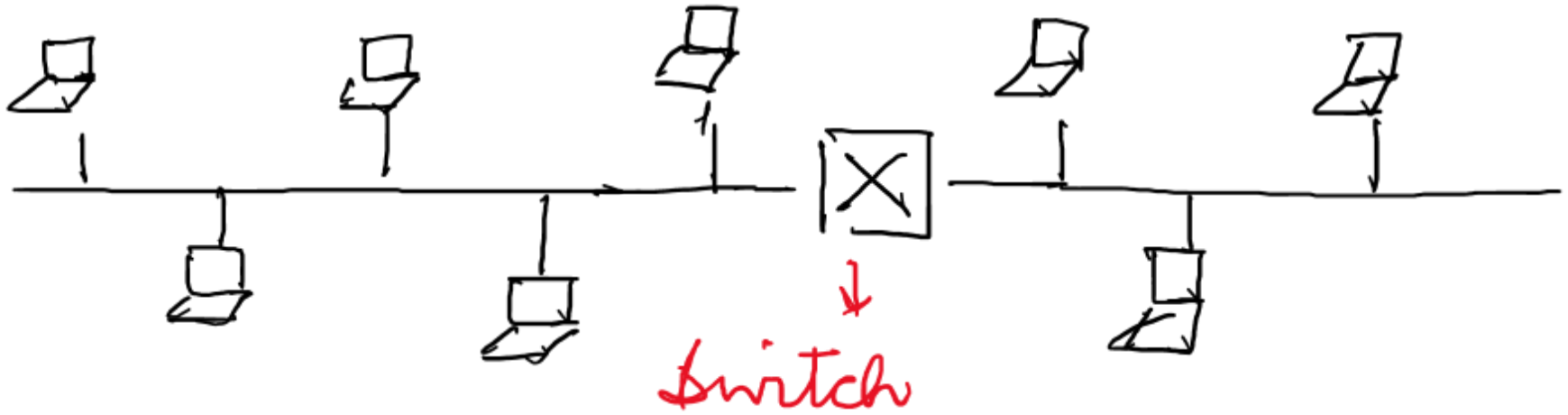
- channel partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use
- random access
  - channel not divided, allow collisions
  - “recover” from collisions
- controlled access
  - nodes take turns
  - e.g., polling, token passing

So Far..

- We know how to communicate in case of a multiple-access link

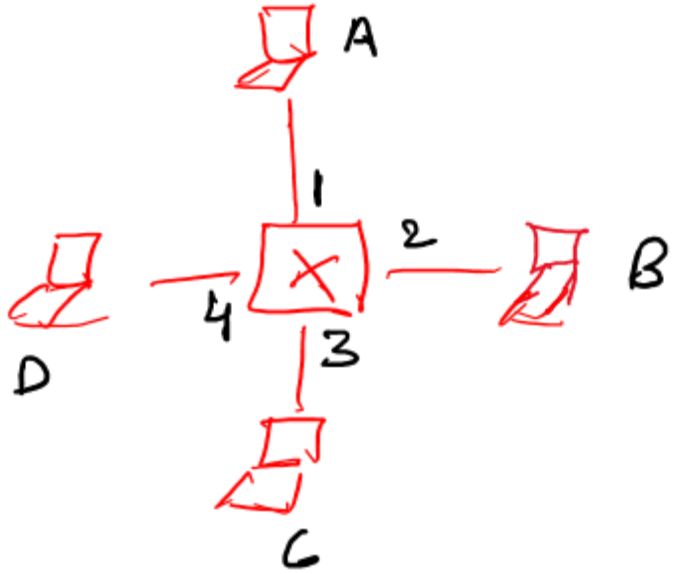


- How do we scale?



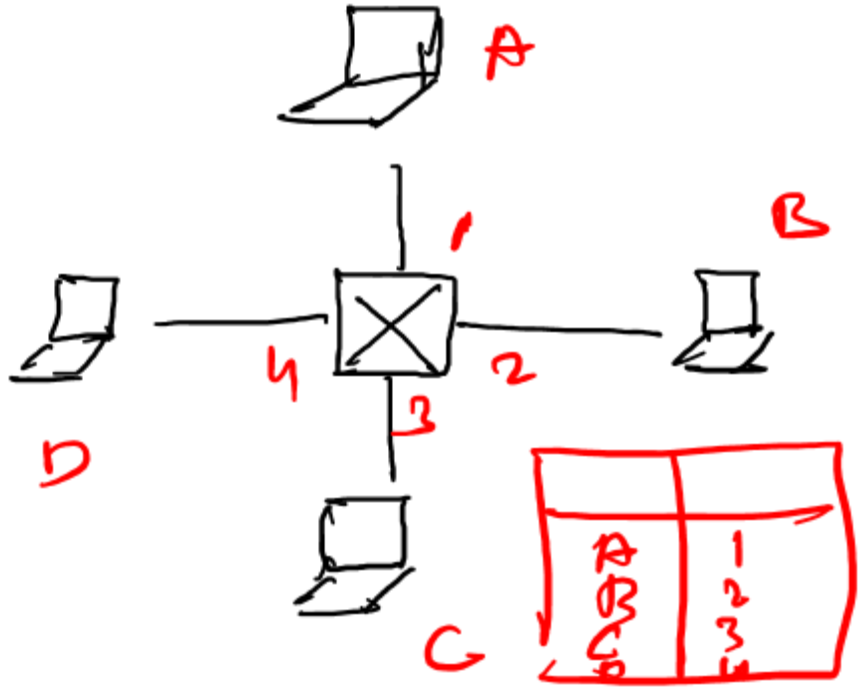
Switched Network

- Star topology
- Switches in both link layer or L2 (Ethernet protocol) and network layer or L3 (IP)
- Use store and forward approach
- L3 switches are also called routers, while L2 switches are also called bridges, ethernet switches
- **This lecture**: L2 switches or bridges or ethernet switch



How does L2 switching work?

- Multiple physical ports (indexed)
  - Input port: frame arrival
  - Output port: frame departure
- Forwarding: Data Plane
  - Moving frame from input port to the appropriate output port.
  - Forwarding table: map of destination address to output port
- Need algorithms to fill the forwarding table aka routing → Control Plane

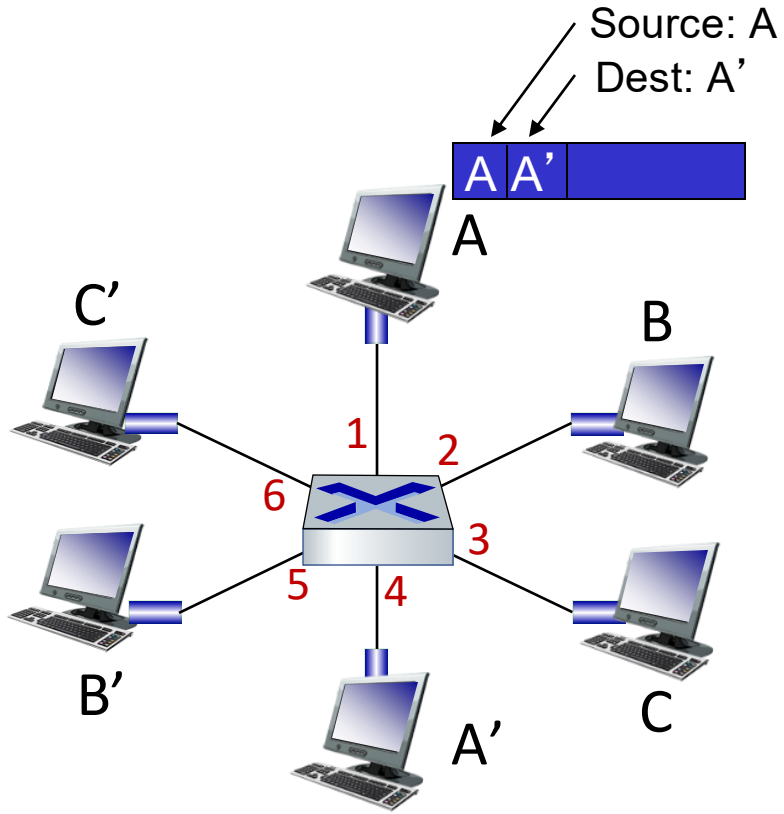


L2 Routing

- Using static tables added by network administrator
  - Difficult to support dynamic addition of nodes (end-hosts and switches)

L2 Routing: Self-learning Switch

- 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



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

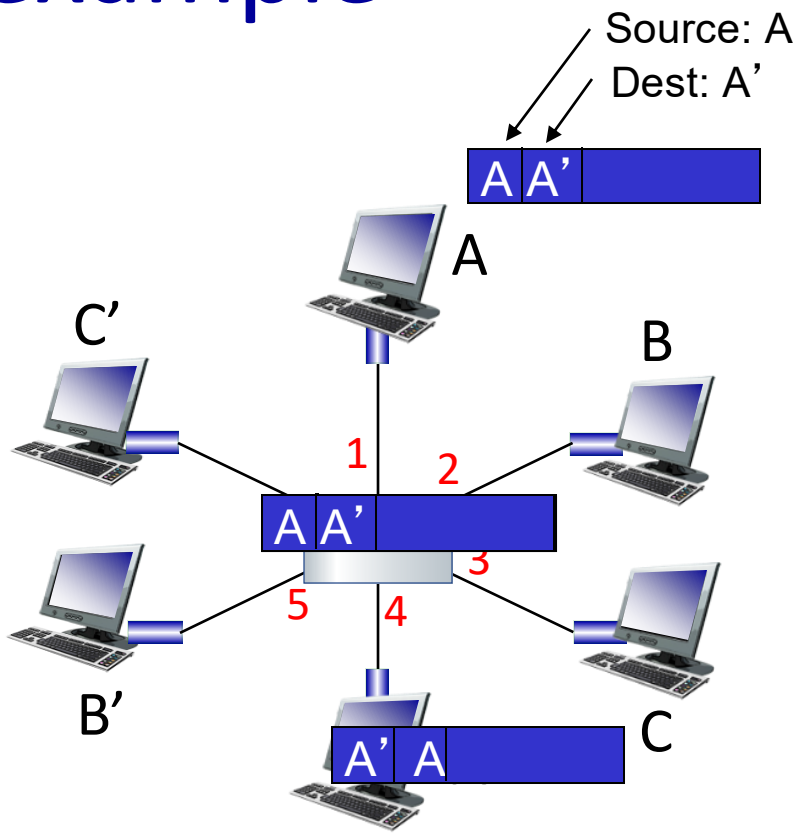
L2 Routing: Self-learning Switch

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
   then {
   if destination on segment from which frame arrived
      then drop frame
       else forward frame on interface indicated by entry
    }
   else flood /* forward on all interfaces except arriving interface */

Self-learning, Forwarding: example

- frame destination, A', location unknown: flood
- destination A location known: selectively send on just one link

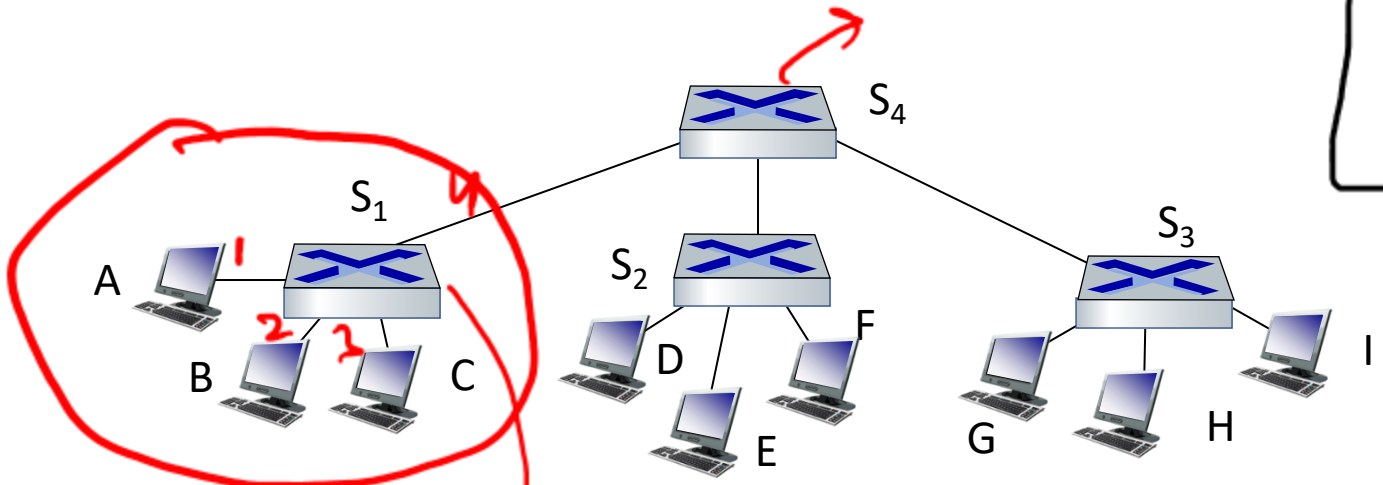


MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Interconnecting switches

self-learning switches can be connected together:



Self learning! (works exactly the same as in single-switch case!)

Q: show switch tables and packet forwarding in S_1, S_2, S_3, S_4