Link Layer

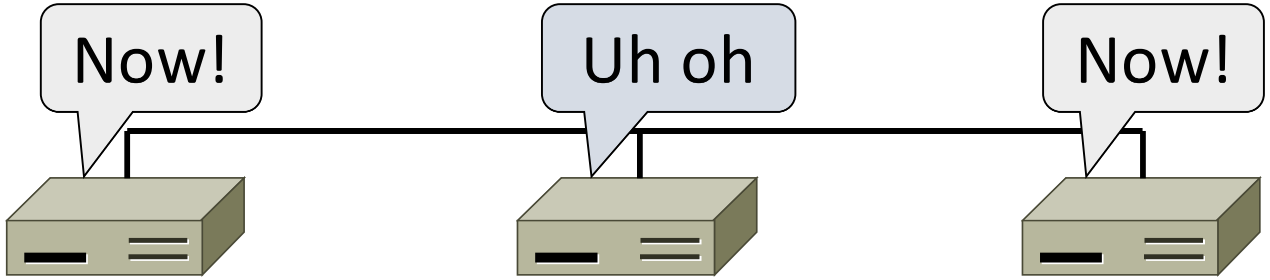
(continued)

Topics

1. Framing
   - Delimiting start/end of frames
2. Error detection and correction
   - Handling errors
3. Retransmissions
   - Handling loss
4. Multiple Access
   - 802.11, classic Ethernet
5. Switching
   - Modern Ethernet

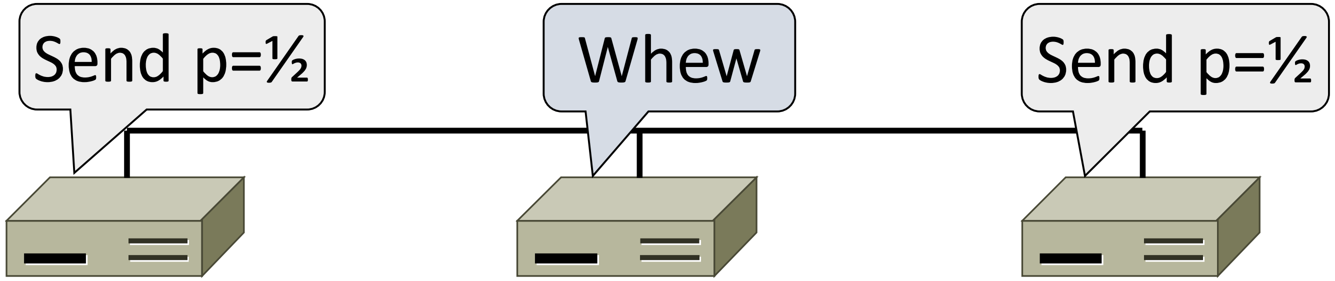
CSMA "Persistence"

- Problem is that multiple waiting nodes will queue up then collide
  - More load, more of a problem



CSMA "Persistence" (2)

- Intuition for a better solution
  - If there are N queued senders, we want each to send next with probability 1/N

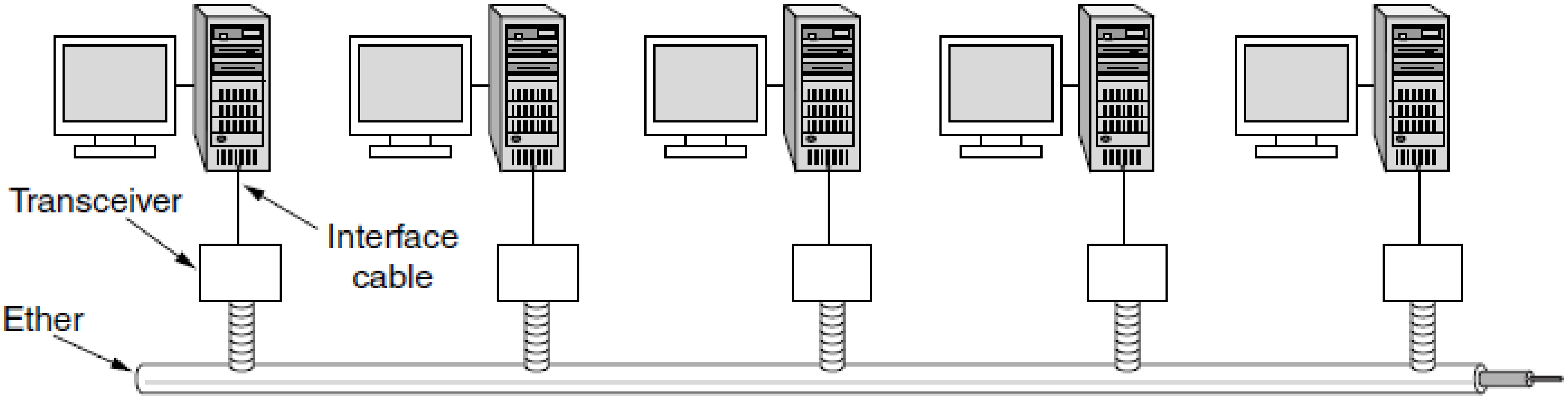


Binary Exponential Backoff (BEB)

- Cleverly estimates the probability
  - 1st collision, wait 0 or 1 frame times
  - 2nd collision, wait from 0 to 3 times
  - 3rd collision, wait from 0 to 7 times …
- BEB doubles interval for each successive collision
  - Quickly gets large enough to work
  - Very efficient in practice

Classic Ethernet, or IEEE 802.3

- Most popular LAN of the 1980s, 1990s
  - 10 Mbps over shared coaxial cable
  - Multiple access with "1-persistent CSMA/CD with BEB"



Ethernet Frame Format

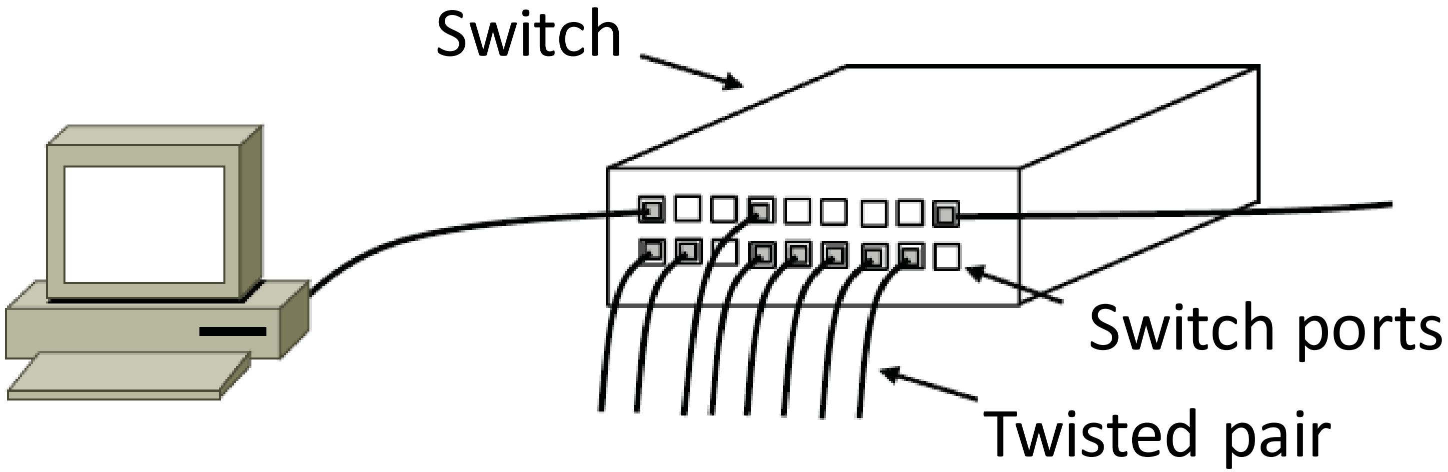
- Has addresses to identify the sender and receiver
- CRC-32 for error detection
- No ACKs
- Start of frame identified with physical layer preamble

Packet from Network layer (IP)

	Preamble	Destination address	Source address	Type	Data	Pad	Check-sum
Bytes	8	6	6	2	0-1500	0-46	4

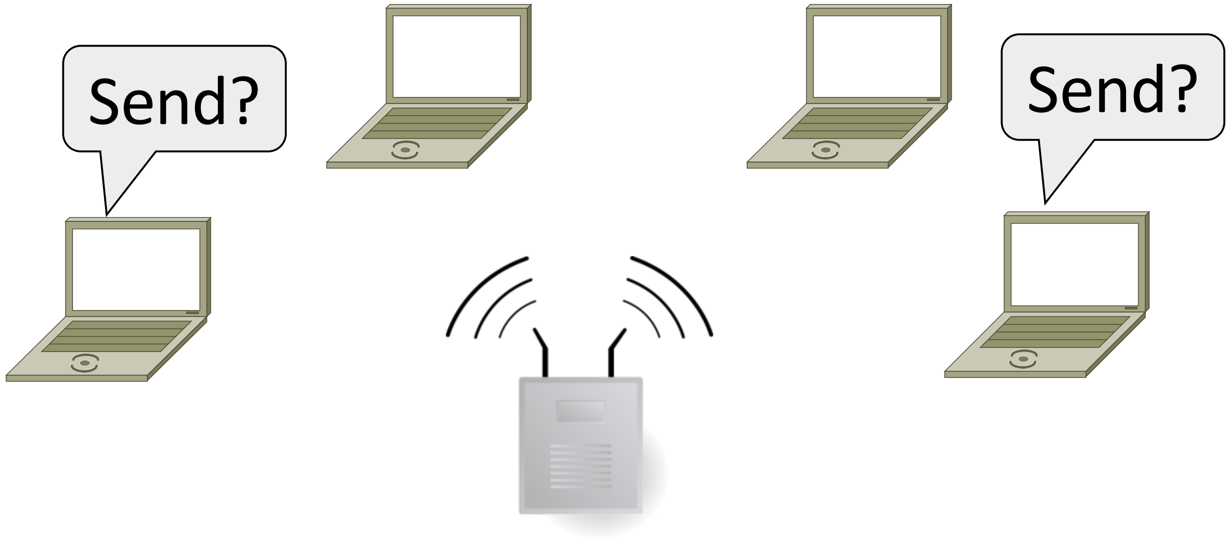
Modern Ethernet

- Based on switches, not multiple access, but still called Ethernet
  - We'll get to it in a later segment, but...
  - Why did a shared cable become unacceptable?



WiFi

- Do wireless nodes share a single link? Yes!
- Build on our simple, wired model



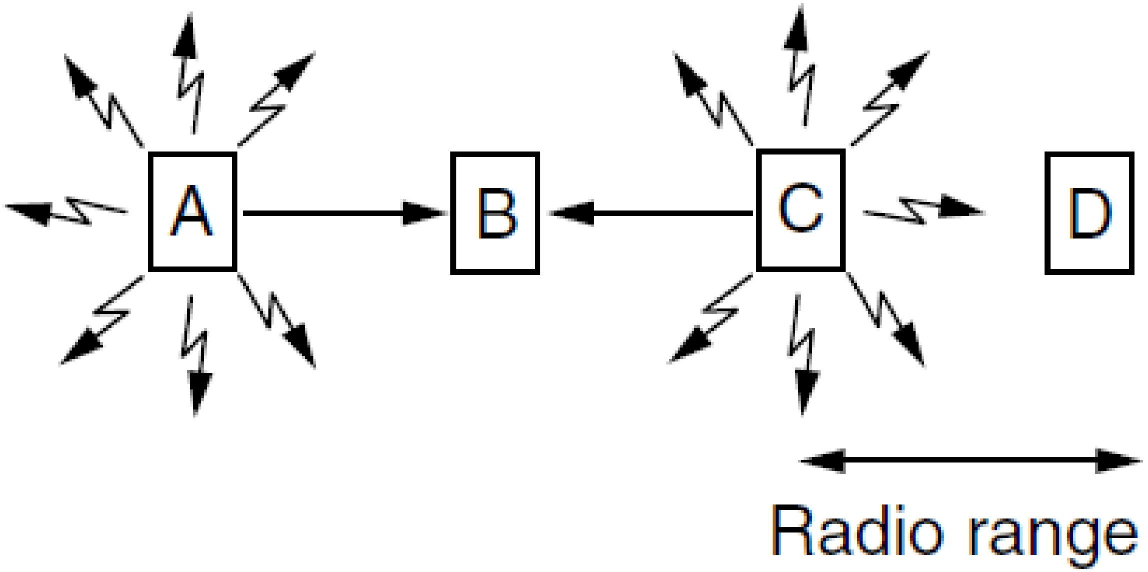
Wireless Complications

- Wireless is more complicated than the wired case (Surprise!)
    1. Nodes may have different areas of coverage – doesn't fit Carrier Sense
    2. Nodes can't hear while sending – can't Collision Detect

≠ CSMA/CD

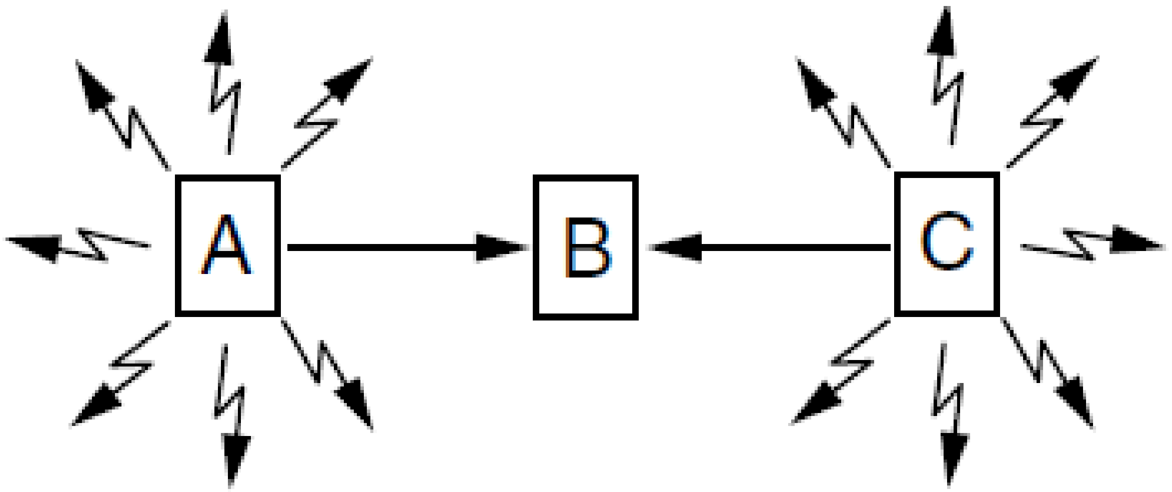
Different Coverage Areas

- Wireless signal is broadcast and received nearby, where there is sufficient SNR (signal to noise ratio)



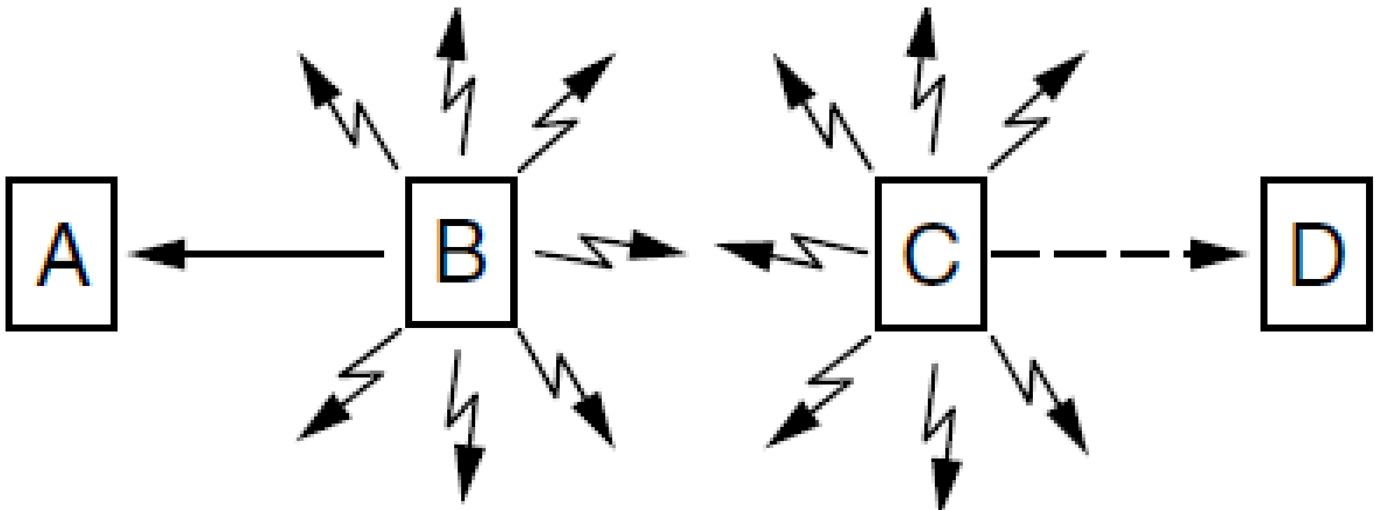
Hidden Terminals

- Nodes A and C are hidden terminals when sending to B
  - Can’t hear each other (to coordinate) yet collide at B
  - We want to avoid the inefficiency of collisions



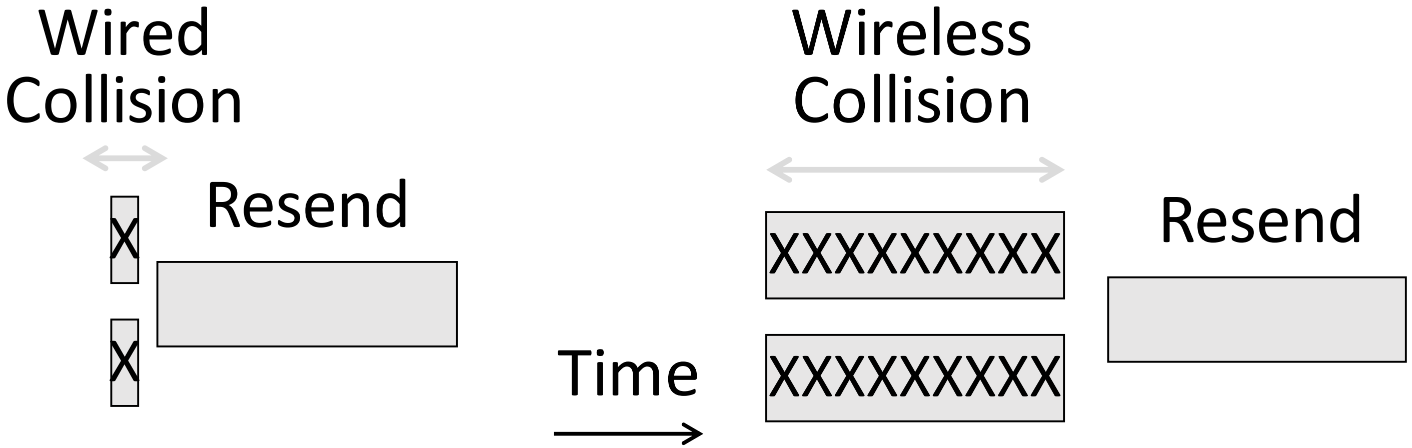
Exposed Terminals

- B and C are <u>exposed terminals</u> when sending to A and D
  - Can hear each other yet don't collide at receivers A and D
  - We want to send concurrently to increase performance



Nodes Can't Hear While Sending

- With wires, detecting collisions (and aborting) lowers their cost
- More wasted time with wireless



MACA (Multiple Access with Collision Avoidance)

- MACA uses a short handshake instead of CSMA (Karn, 1990)
  - 802.11 uses a refinement of MACA

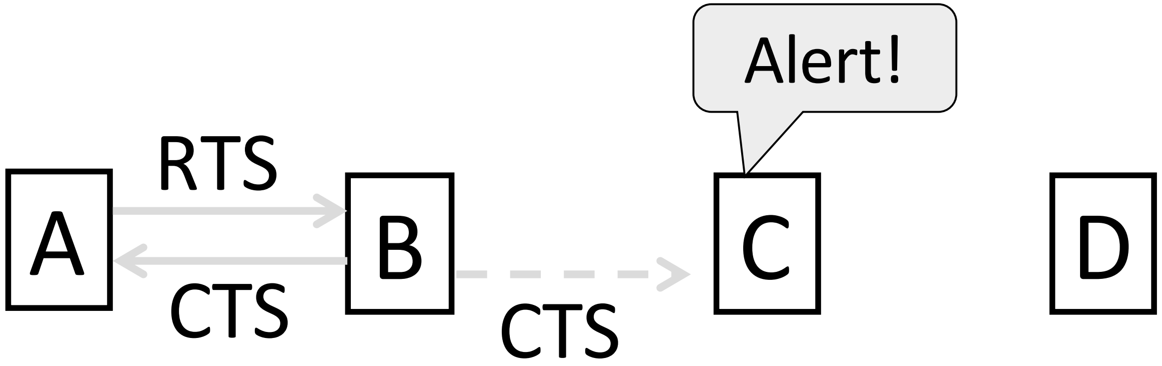
- Protocol rules:
  1. A sender node transmits a RTS (Request-To-Send, with frame length)
  2. The receiver replies with a CTS (Clear-To-Send, with frame length)
  3. Sender transmits the frame while nodes hearing the CTS stay silent
  - Collisions on the RTS/CTS are still possible, but less likely

MACA – Hidden Terminals

- A→B with hidden terminal C
  1. A sends RTS, to B

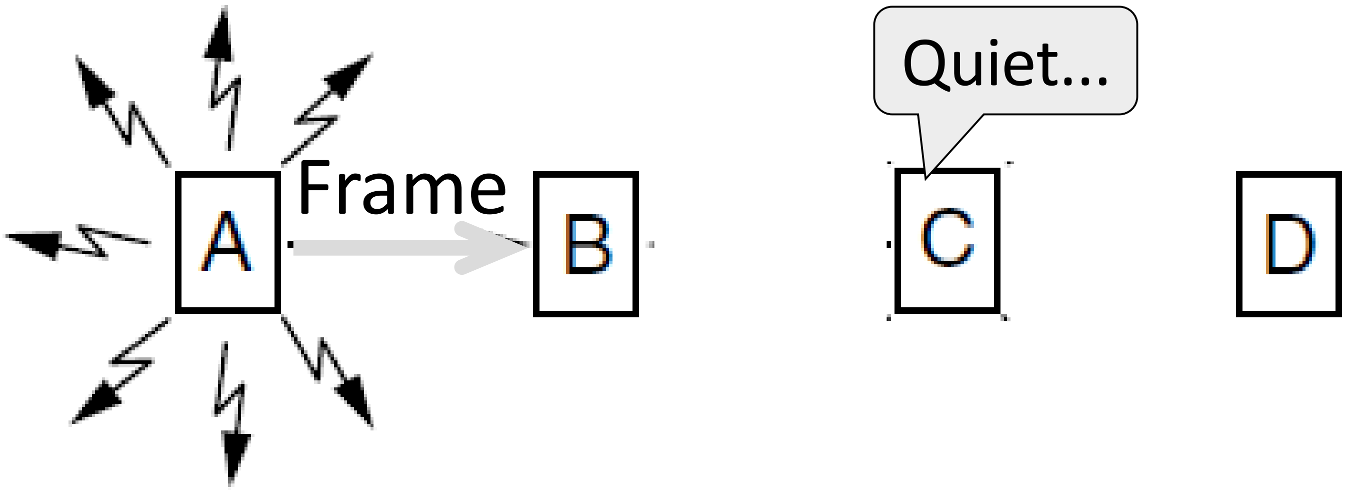
MACA – Hidden Terminals (2)

- A→B with hidden terminal C
    2. B sends CTS, to A, and C too



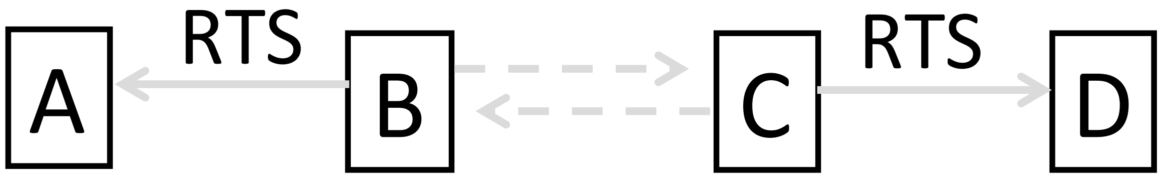
MACA – Hidden Terminals (3)

- A→B with hidden terminal C
  3. A sends frame while C defers



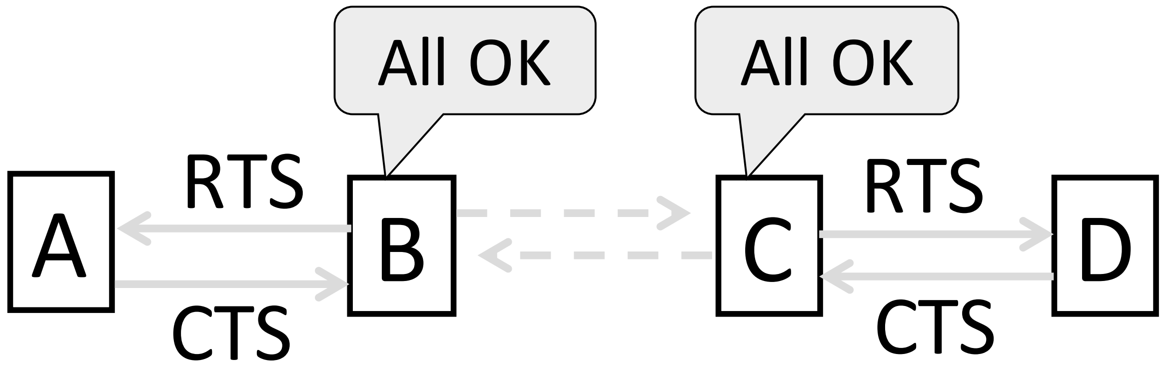
MACA – Exposed Terminals

- B→A, C→D as exposed terminals
  - B and C send RTS to A and D



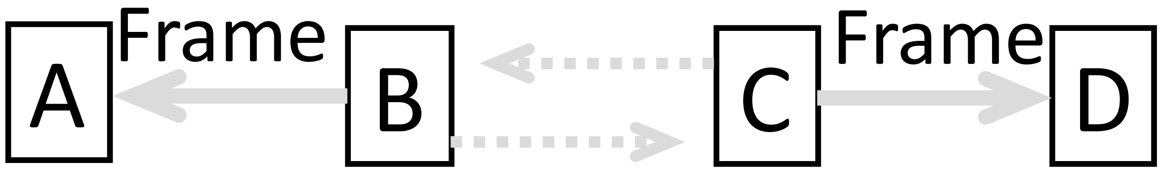
MACA – Exposed Terminals (2)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



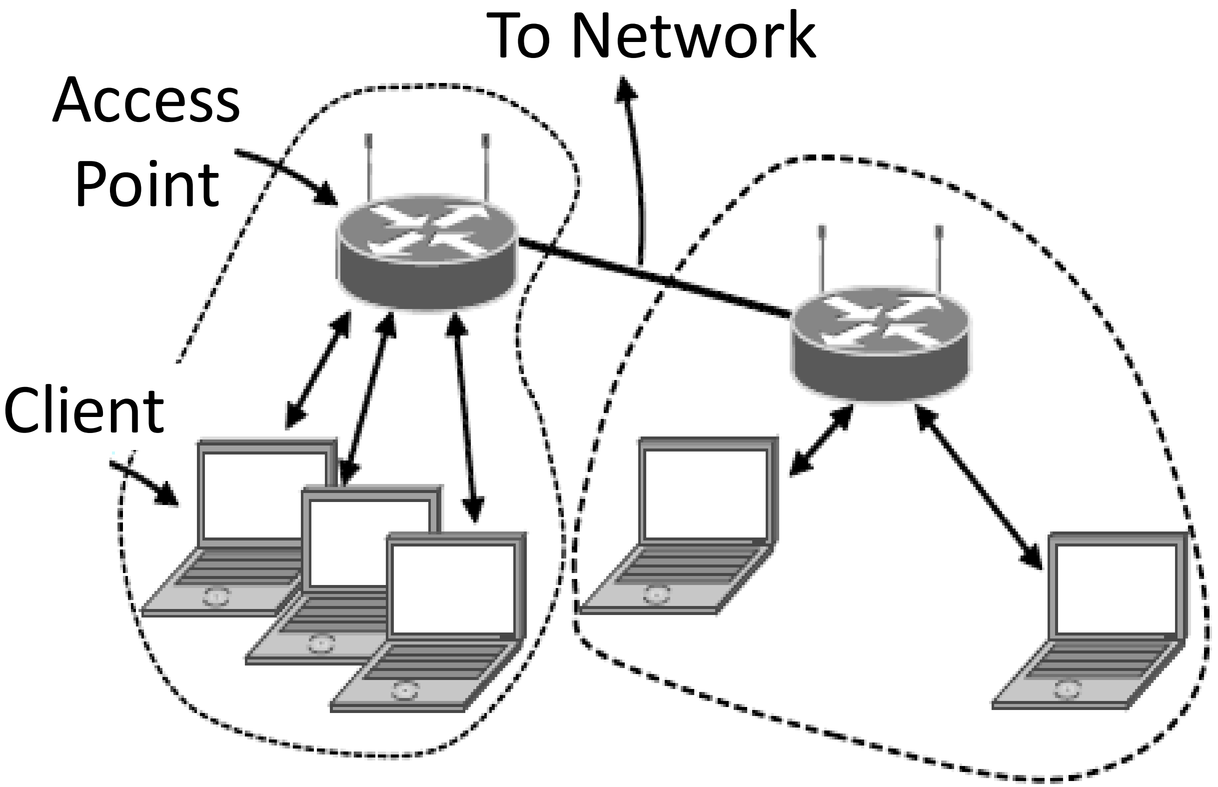
MACA – Exposed Terminals (3)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



802.11, or WiFi

- Very popular wireless LAN started in the 1990s
- Clients get connectivity from a (wired) AP (Access Point)
- It's a multi-access problem ☺
- Various flavors have been developed over time
  - Faster, more features



802.11 Link Layer

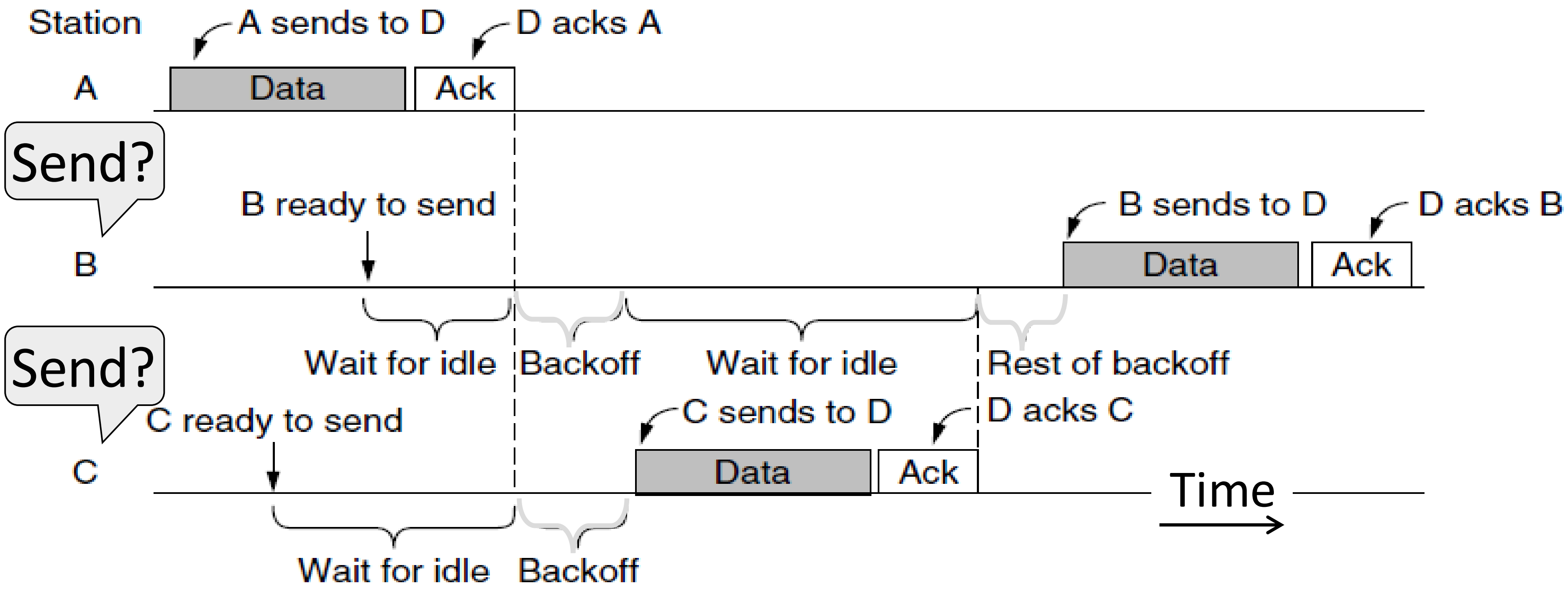
- Multiple access uses CSMA/CA (next); RTS/CTS optional
- Frames are ACKed and retransmitted with ARQ
- Funky addressing (three addresses!) due to AP (access point)
- Errors are detected with a 32-bit CRC
- Many, many features (e.g., encryption, power save)

Packet from Network layer (IP)

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

802.11 CSMA/CA for Multiple Access

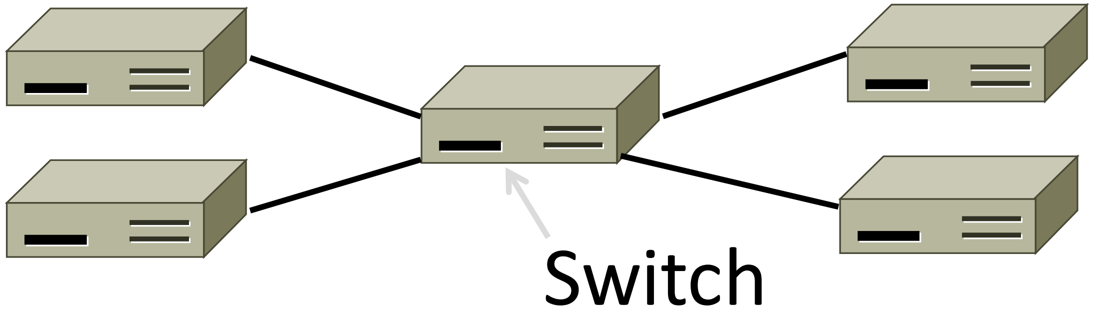
- Still using Binary Exponential Backoff!



Switching

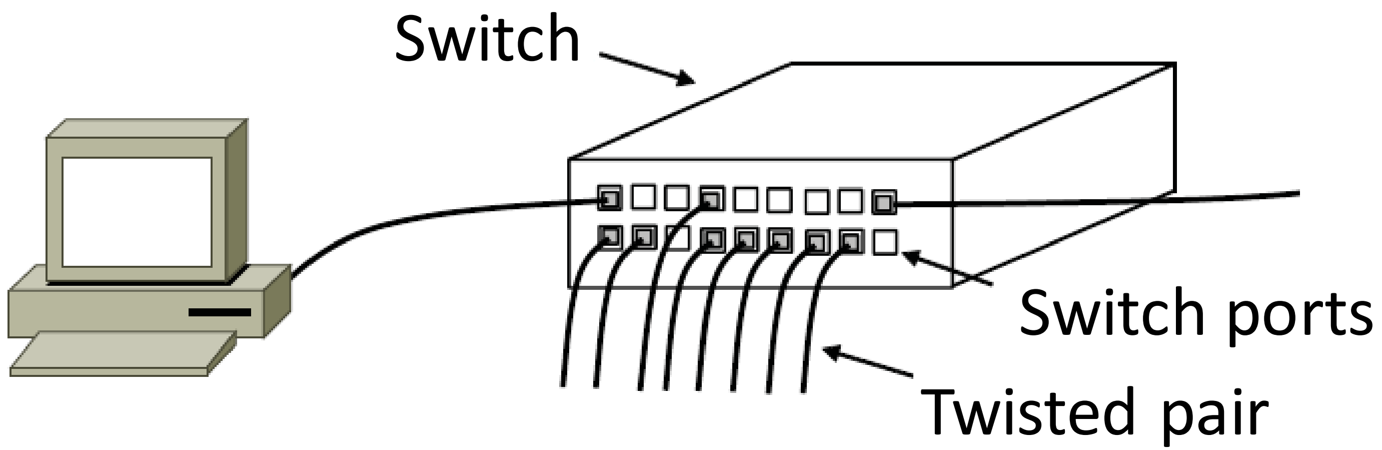
Topic

- How do we connect nodes with a switch instead of multiple access
  - Uses multiple links/wires
  - Basis of modern (switched) Ethernet
- Why do we want to?



Switched Ethernet

- Hosts are wired to Ethernet switches with twisted pair
  - Switch serves to connect the hosts
  - Wires usually run to a closet

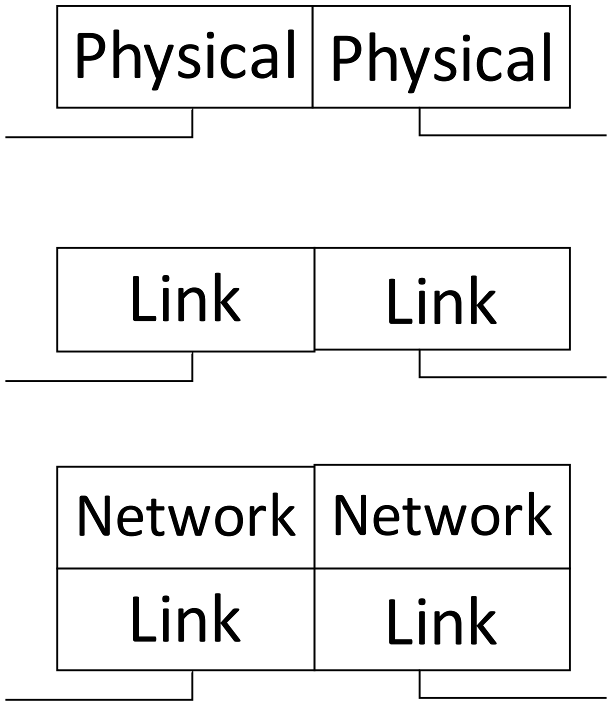


What’s in the box?

Hub, or repeater

Switch

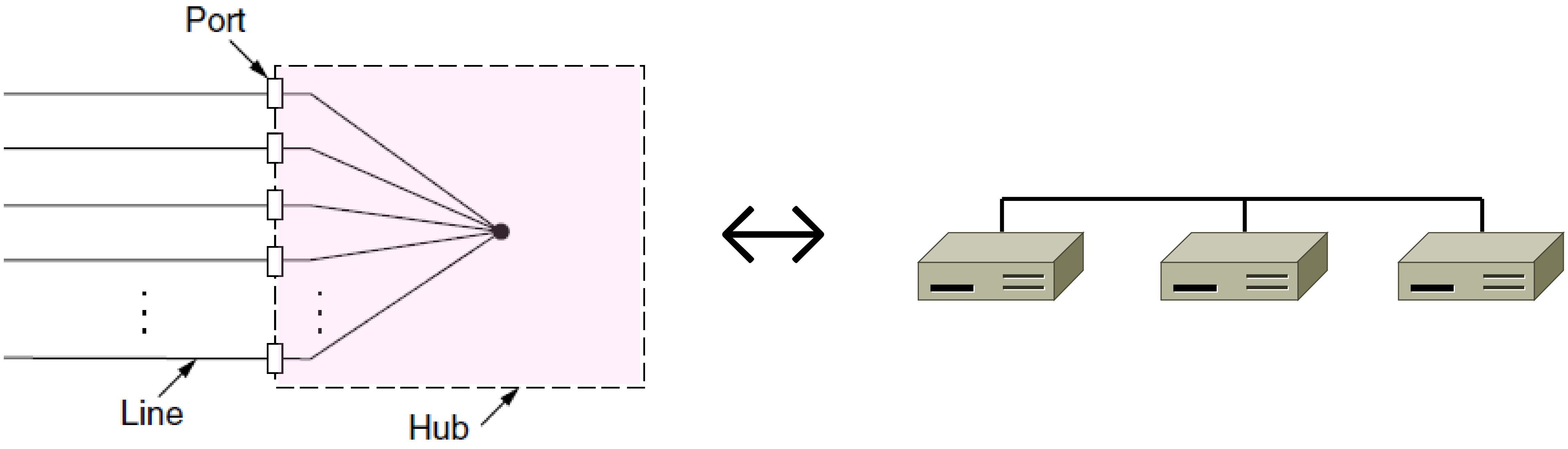
Router



All look like this:

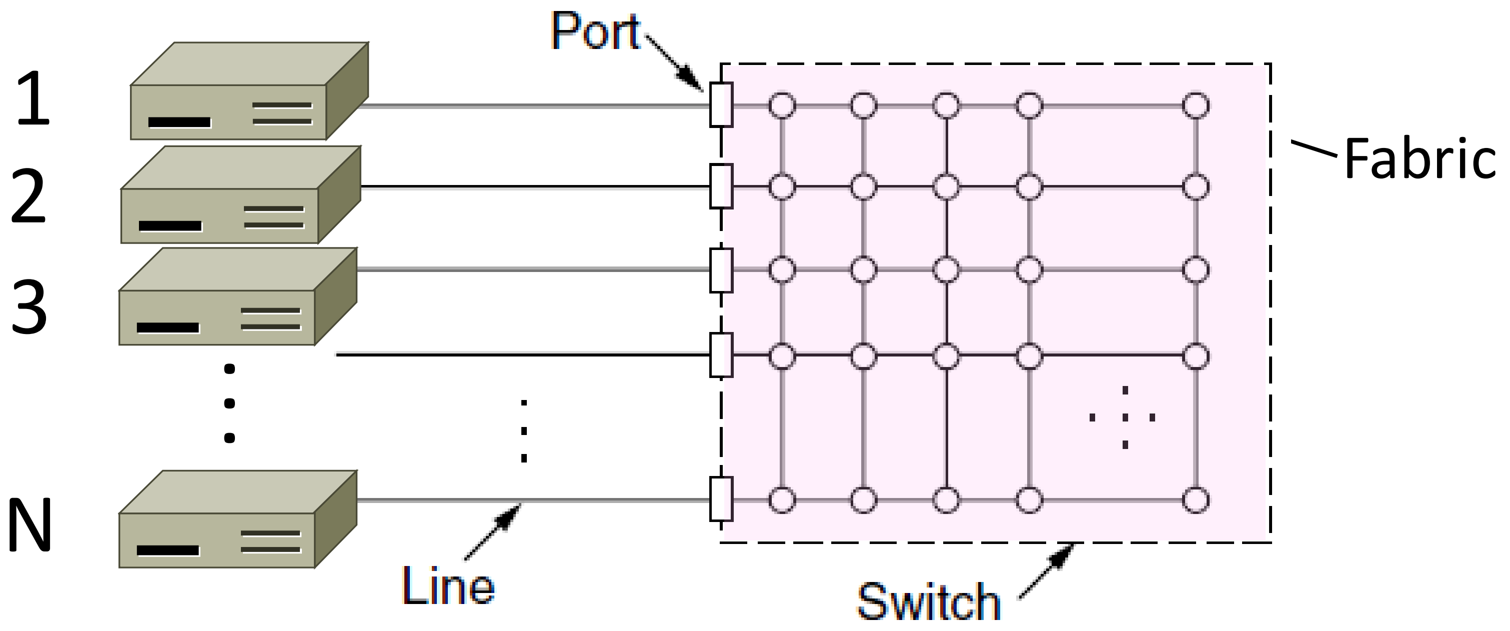
Inside a Hub (physical layer)

- All ports are wired together; more convenient and reliable than a single shared wire



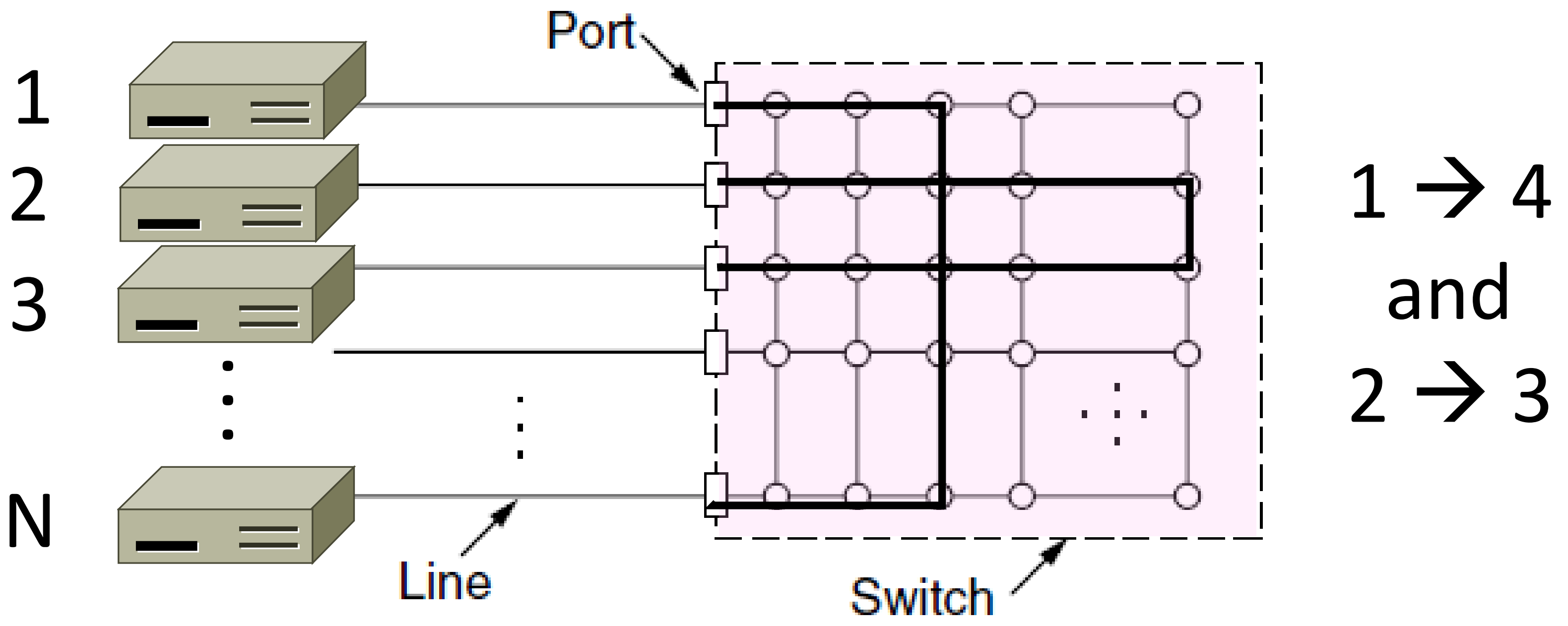
Inside a Switch

- Uses frame addresses to connect input port to the right output port; multiple frames may be switched in parallel



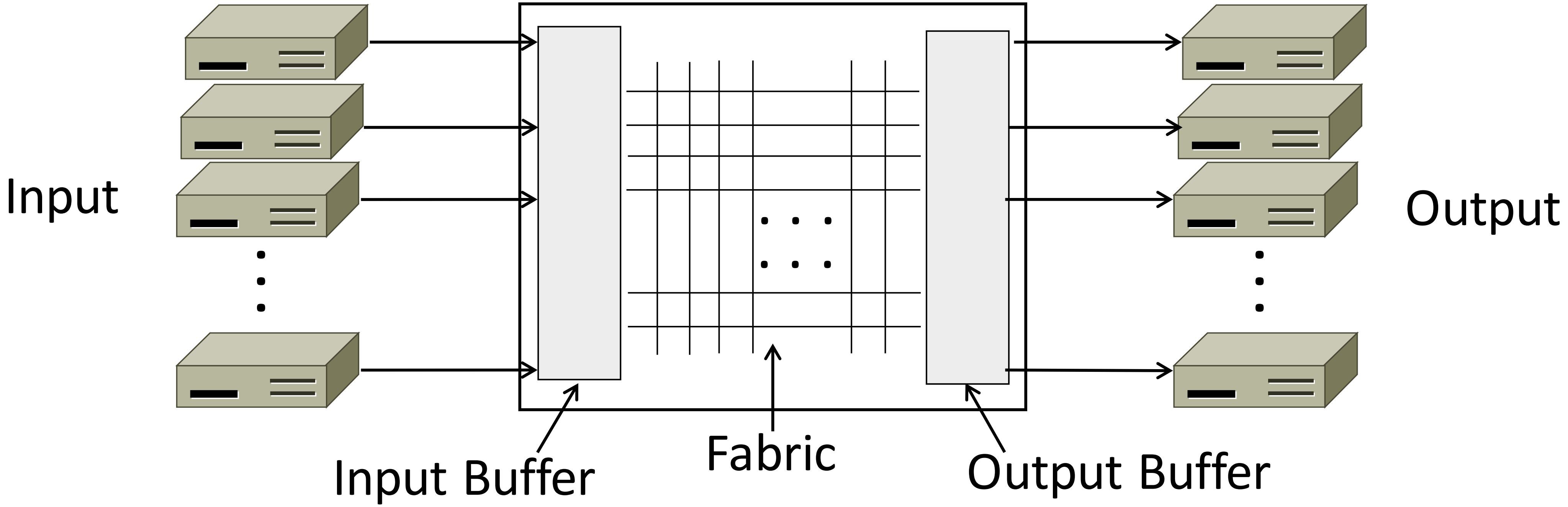
Inside a Switch

- Port may be used for both input and output (full-duplex)
  - Just send, no multiple access protocol



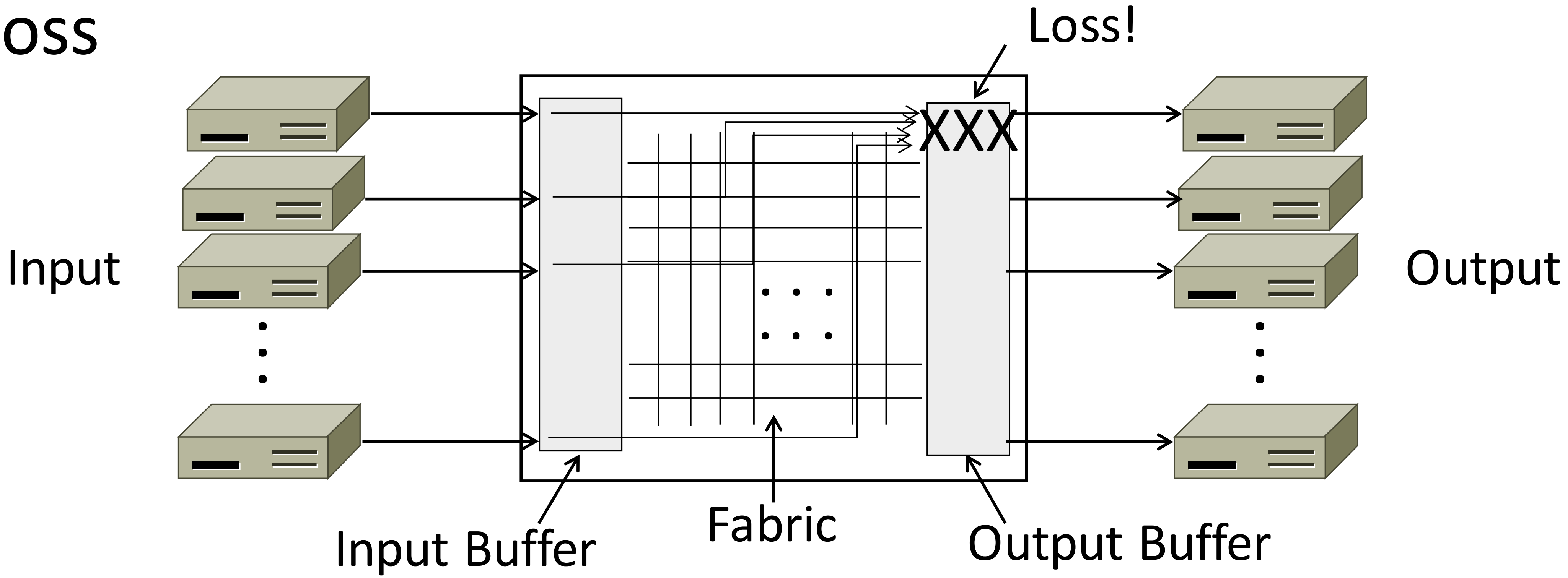
Inside a Switch

- Need buffers for multiple inputs to send to one output



Inside a Switch

- Sustained overload will fill buffer and lead to frame loss

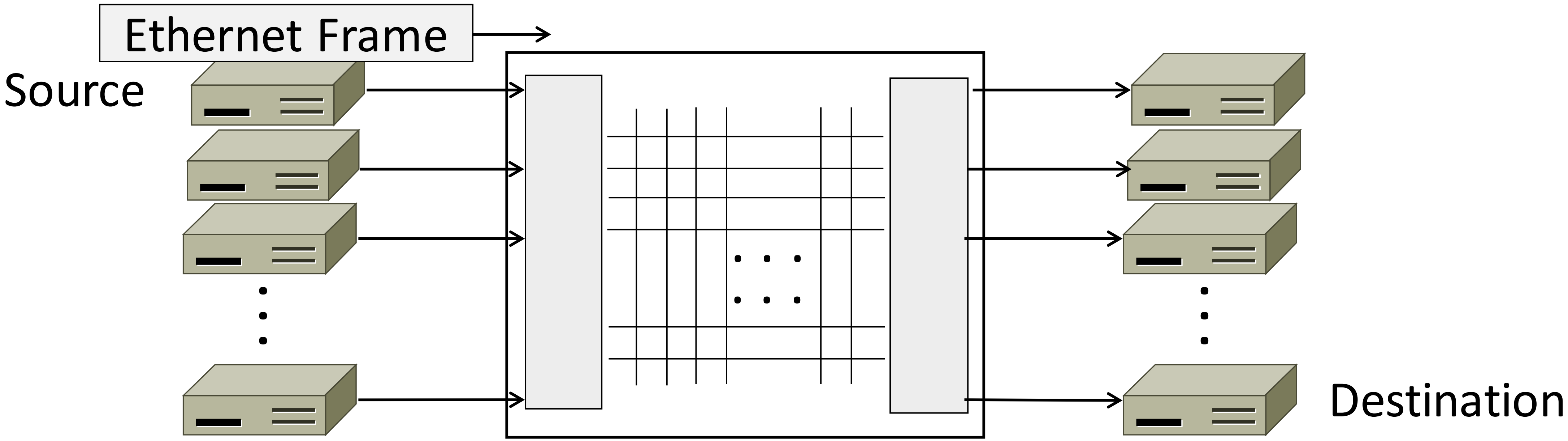


Advantages of Switches

- Switches and hubs have replaced the shared cable of classic Ethernet
  - Convenient to run wires to one location
  - More reliable; wire cut is not a single point of failure that is hard to find
- Switches offer scalable performance
  - E.g., 1000 Mbps per port instead of 1000 Mbps for all nodes of shared cable / hub
  - No collision and backoff performance losses

Switch Forwarding

- Switch needs to find the right output port for the destination address in the Ethernet frame. How?
  - Link-level, don't look at IP



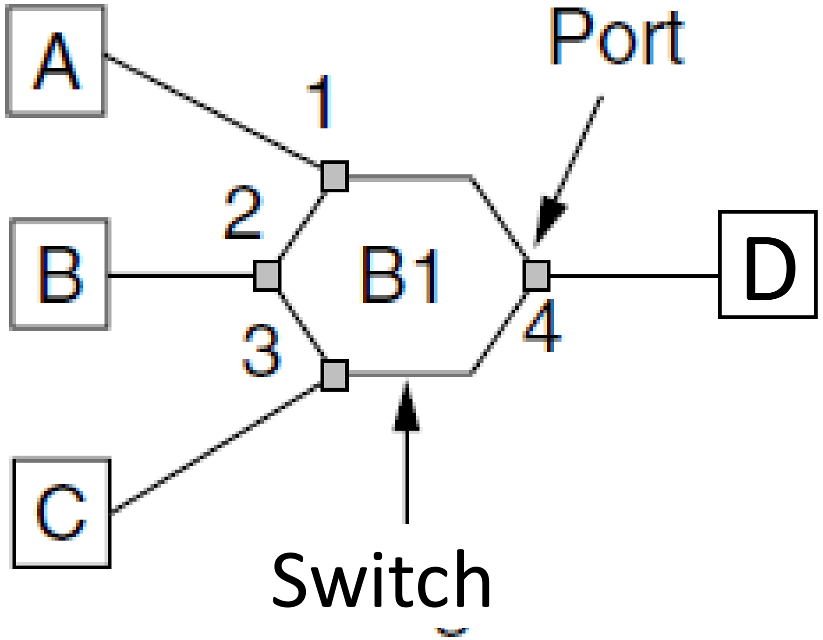
Backward Learning

- Switch forwards frames with a port/address table
  - keys are MAC addresses, values are port id

- Switch "learns" the table's contents:
  1. To fill the table, it looks at the source address of input frames
  2. To forward, it looks in the table
     1. If entry found, sends to that port
     2. otherwise, send on all ports (broadcast)

Backward Learning (2)

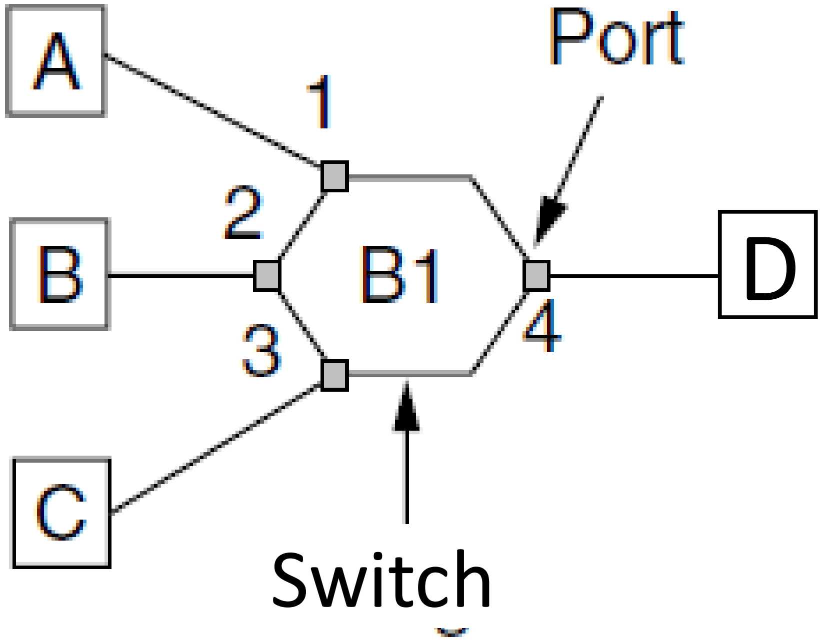
- 1: A sends to D



Address	Port
A	
B	
C	
D	

Backward Learning (3)

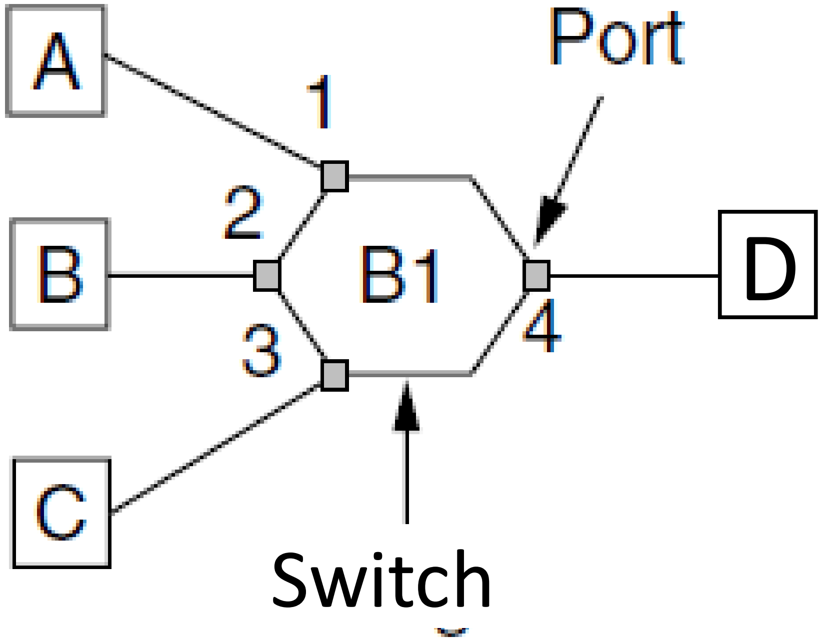
- 2: D sends to A



Address	Port
A	1
B	
C	
D	

Backward Learning (4)

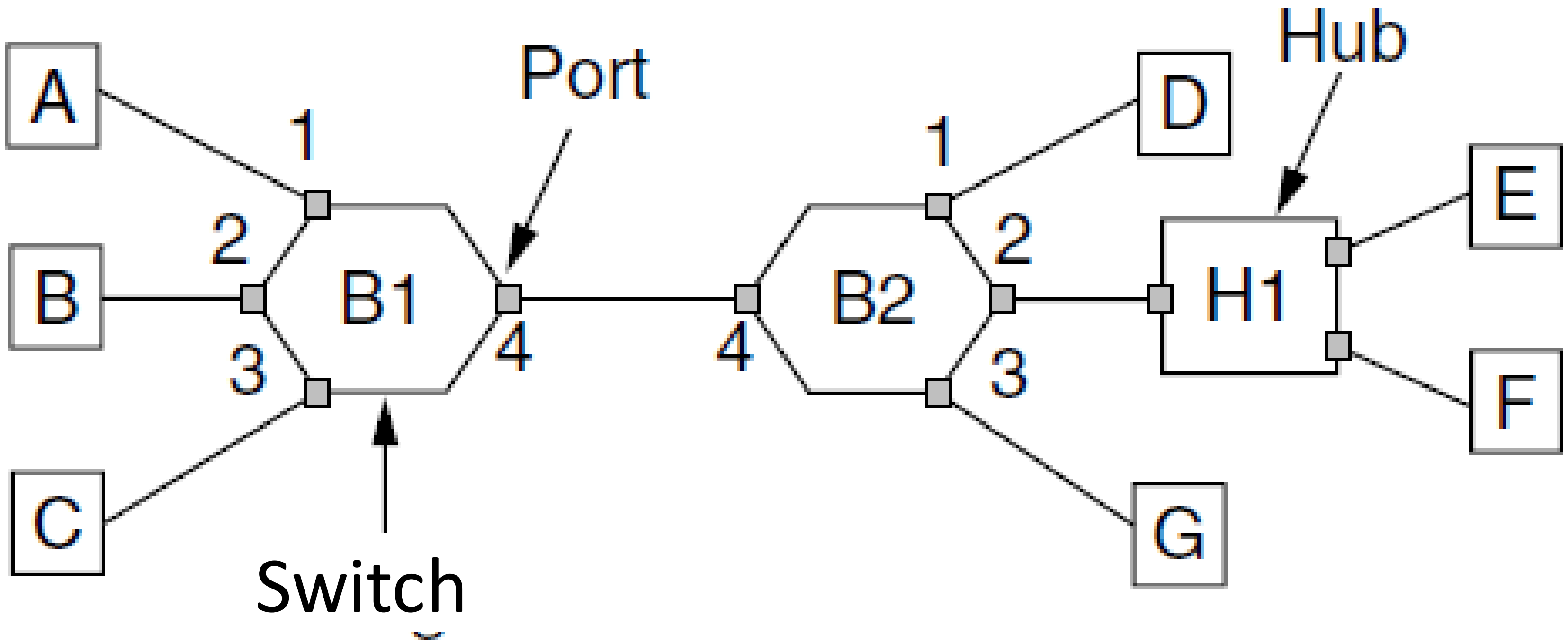
- 3: A sends to D



Address	Port
A	1
B	
C	
D	4

Learning with Multiple Switches

- Just works with multiple switches and a mix of hubs *assuming no loops*, e.g., A -> D then D -> A



Problem – Forwarding Loops

- May have a loop in the topology
  - Redundancy in case of failures
  - Or a simple mistake
- Want LAN switches to "just work"
  - Plug-and-play, no changes to hosts
  - But loops cause a problem …