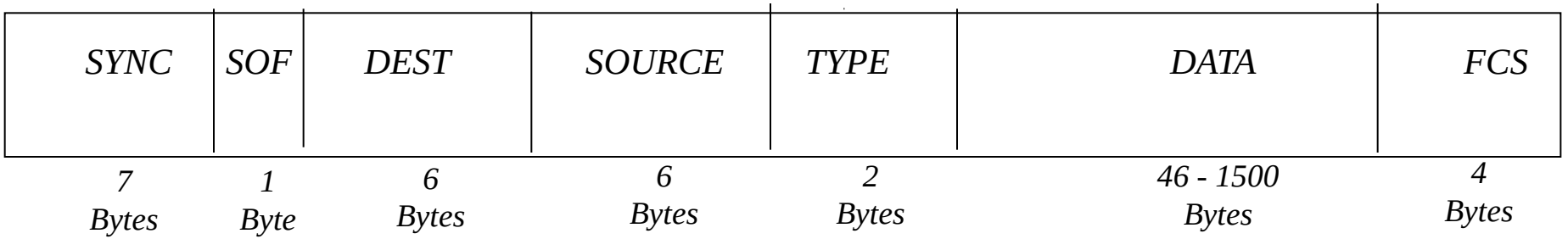
Lecture 4B

Local Area Networks and Bridges

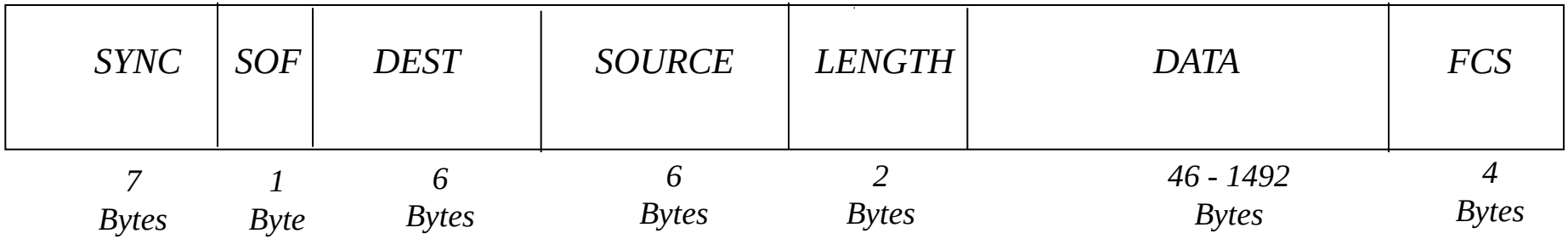
Ethernet

- Invented by Boggs and Metcalf in the 1970's at Xerox
- Local area networks were needed to connect computers, share files, etc. Thick or Thin Ethernet Cable to which computers could be connected every 2.5 meters. Now also fiber optic links, twisted pair in conjunction with switches.
- Original Ethernet and IEEE 802.3 differ in a small way
  - Original standard was 10 Mbps, 1995 100Mbps
  - 1998 – standards approved for 1 Gbps (1 GbE)
  - 2002 – standards approved for 10 GbE
    - (from 10 GbE, full duplex and no CSMA / CD)
  - 2010 – standards approved for 40GbE and 100 GbE
  - 2017 – 200 GbE and 400 GbE
  - 2024 – 800 GbE

Ethernet DIX versus Ethernet 802.3 Logical Formats



DIX: Type field or upper layer protocol; for example, 0x0800 means data is an IPv4 packet, 0x0806 is ARP, 0x86DD is IPv6. Note type values ≥ 0x0600



802.3: Length of data field; the first 8 bytes of the data is the LLC/SNAP protocol which is 8 bytes and includes the protocol type field. Note length values < 0x0600.

LLC/SNAP Header: 1 byte DSAP and SSAP (0xAA), 1 byte Control (usually set to 03), 3 bytes OSI (typically set to 0), 2 bytes protocol type.

Ethernet Addressing

- EUI-48 and EUI-64 (48 bits and 64 bits respectively)
- 48 bits
  - Written as 12 hex digits, with bytes separated by dashes or colons
    - A4-81-42-5B-D2-BB
    - 6 bytes with high order byte (most significant) on the left. Thus in the example, the high order byte is A4
- First 3 bytes are assigned to the vendor (OUI) and the next 3 bytes are the IAB assigned by vendor.
- Transmission order online:
  - Byte order, this means highest or most significant byte is sent first (big endian order)
  - Within the byte, the least significant bit is sent first.
- Special bits
  - First bit transmitted is 0 for unicast and 1 for multicast
  - Second bit transmitted is 0 for global address, for local address

Original Ethernet Specs

- 10 Mbits / sec
- Maximum distance between two stations of 2500 meters
- Maximum one-way propagation time of 22.5 μsecs
- Acquisition time (twice propagation time) of 45 μsecs
- CSMA / CD with truncated binary exponential backoff
- Slot time is 51.2 μsecs
  - In 51.2 μsecs can transmit $51.2 \times 10^{-6} \times 10 \times 10^{6}$ bits = 512 bits = 64 bytes

Transmission Algorithm

- Start transmitting whenever the ethernet (channel) is in an idle state as determined by the carrier sensing mechanism. Continue to transmit as long as no collision is noted
- If the carrier sensing indicates someone else is transmitting, wait until the ethernet is idle and then transmit
- If there is a collision, then transmit a *jamming signal* for a short time (3.2 μsecs) and then use the truncated binary exponential backoff algorithm to decide when next to try to transmit
- Once station has been transmitting for the *acquisition time*, no one will interfere as all stations will be *deferring*
- Hardware automatically filters out short packets and a minimum packet size is defined of 512 bits (51.2 μsecs)
  or 64 bytes
- Station waits a short time after channel is free before assuming the ethernet is idle – interframe spacing between 9.6 μsecs and 10.6 μsecs.

Truncated Binary Exponential Backoff

- Retransmission strategy after a collision
  - Keep track of the number of collisions
  - For the Nth retransmission attempt, delay for an integral multiple *r* times the slot time (51.2 μsecs) where r is a uniformly distributed integer $0 \leq r < 2^k$ where k = min (N, 10)
  - After 16 collisions, give up and report an error.

Example

Station 1: starts transmitting at 0 μ secs

Station 2: starts transmitting at 3 μ secs

Station 3: starts transmitting at 10 μ secs

Ethernet Performance Analysis (High Load)

- Assume n stations are each trying to transmit with probability $p = 1/n$ in each Ethernet slot time where n is large.
  - We know that the number of expected contention slots is e (since probability of max success is 1/e by the binomial theorem).
- The slot time is $2\tau = 51.2$ μsecs
- Assume that the expected time to transmit a frame is P seconds.
- Then we have that:

  Channel Efficiency = $P/(P + 2\tau e)$
- For a 10 Mb/s Ethernet, the slot time 2τ is 512 bit times or 64 byte times. For 1536 byte packets, we have the maximum efficiency as:

  Channel efficiency = $P/(P + 2\tau e) = 1/(1 + 64\ e/1536) = 1/(1 + e/24) \approx 0.90$
- For the same assumptions, and 64 byte packets, we have:

  Channel efficiency = $1/(1 + e) \approx 0.27$

- Important design parameters
  - Bandwidth
    - Propagation + transmission delay: limits the minimum frame size.
- Physical medium
  - thin cable/thick cable/twisted pair/fiber

10Base5 500 meters thick coax (10 mm cable bus) Ethernet
100 nodes/seg

10Base2 200 meters thin coax (5mm cable bus) 30 nodes/seg

10BaseT 100 meters twisted pair star (to hub) 1024 nodes/seg

10BaseF 2000 meters fiber optics (point to point) 1024 nodes/seg

10Base5/10Base2, cable connected to each machine

10BaseT -- connecting to a hub

10BaseF – connecting between buildings

- Multiple segments can be connected through the repeaters (hubs).
- All segments connected by the repeaters are in the same *collision domain*.
  - constraint: two transceivers may be up to 2.5km apart and separated by 4 repeaters.
  - frame format

| Preamble | Start| Dst Addr | Src Addr | length | Data | Pad | Checksum|

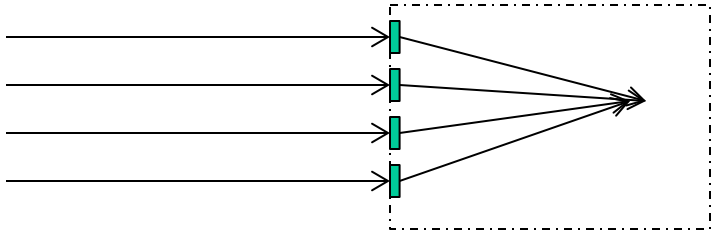
7	1	6	6	2	0-1500	0-46	4

  - Header: 14 bytes, CRC: 4 bytes
  - Minimum frame length: 46 data bytes + 18 bytes header / trailer = 64 bytes (excludes preamble & SD)
  - Maximum frame length: 1500 date bytes + 18 bytes header/ trailer = 1518 bytes (excludes preamble & SD)

Evolution to Switched Ethernet

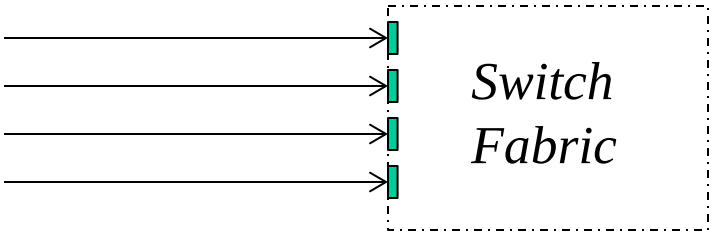
- Evolution of Ethernet first saw a change in the wiring pattern where dedicated cable (wires) from a station went to a central *hub*.

All lines go to a port on the hub. They are then connected inside the hub to essentially be equivalent to one Ethernet. All the stations are in a single collision domain.



- Next step was to move to using a *switch*.

Use a switch fabric that sends a frame incoming on one port to the exact destination port. Each port is in its own collision domain. Stations can also send simultaneously with no collision.



- Ethernet Switch: Increase the bandwidth, segments connected by switch have different collision domain.
  - Ethernet switch: data link layer device
  - Ethernet hub (repeater): physical layer device

– Fast Ethernet
  - Keep everything as in Ethernet, make the clock faster 100Mbps.
  - Cable
    - » 100Base-T4 100m category 3 UTP, 4 lines.
    - » 100Base-T 100m category 5 (2 twisted pair)
    - » 100Base-F 2000m Fiber optic

- What are the problems?
  - Cable
  - CSMA/CD?
    - minimum frame size = 64bytes = 512 bits,
    - 5.12µs are needed to transmit using 100Mbps transmission rate.
    - What can you do about this?
      - Increase the minimum frame size (if cable length is the same).
      - Reduce cable length
    - Solution
      - » Reduce the cable length by a factor of 10, maximum length = 200 meters (100-Base-T, 100 meter cable).
    - Also can do the following: use full duplex mode: point to point connection
      - no contention. No CSMA/CD needed, can have longer cable.

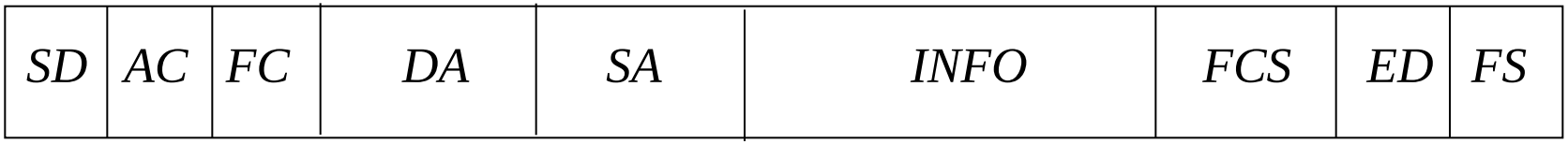
- Gigabit Ethernet: make it even faster at 1Gbps.
  - Cable: mainly fiber optics.
  - CSMA/CD domain
    - Shortening the cable? 20 meters
  - Alternative: increase the minimum frame size to 512 bytes, CSMA/CD domain 200 meters (not much error margin)
    - Experimental studies say that typical frame size are 200 - 300 bytes.
  - backward compatibility:
    - carrier extension -- short packet, stuff extra bits to make to 512 bytes
  - improve performance: packet bursting -- transmit a burst of small frames, only the first one need carrier extension.
  - The actual coding and signaling technology has changed.
  - Jumbo Frames (allow frames up to 9 KB)

- Note that hubs at higher rates are obsolete and switches must be used

Token Ring Obsolete

- Complexity: Handling the token
  - Token is a distinctive bit pattern that “circulates” in the ring
  - Normally called free token
  - When a station transmits, it converts the free token to a “busy token” and sends its data frame (token really is part of data frame)
  - Each station checks the circulating frame, destination station copies to local buffer and modifies some bits in frame and continues the frame on the ring
  - Originating station is responsible for removing the frame and reinserting a free token into the ring
  - Variations: single token vs multiple token
  - Sending station can insert free token immediately after transmitting frame, after header returns, or after entire frame returns

Logical Data Frame Format



FC: 1 byte frame control

DA: 6 bytes destination address

SA: 6 bytes source address

INFO: information field

FCS: 4 bytes frame check sequence

FS: 1 byte frame status

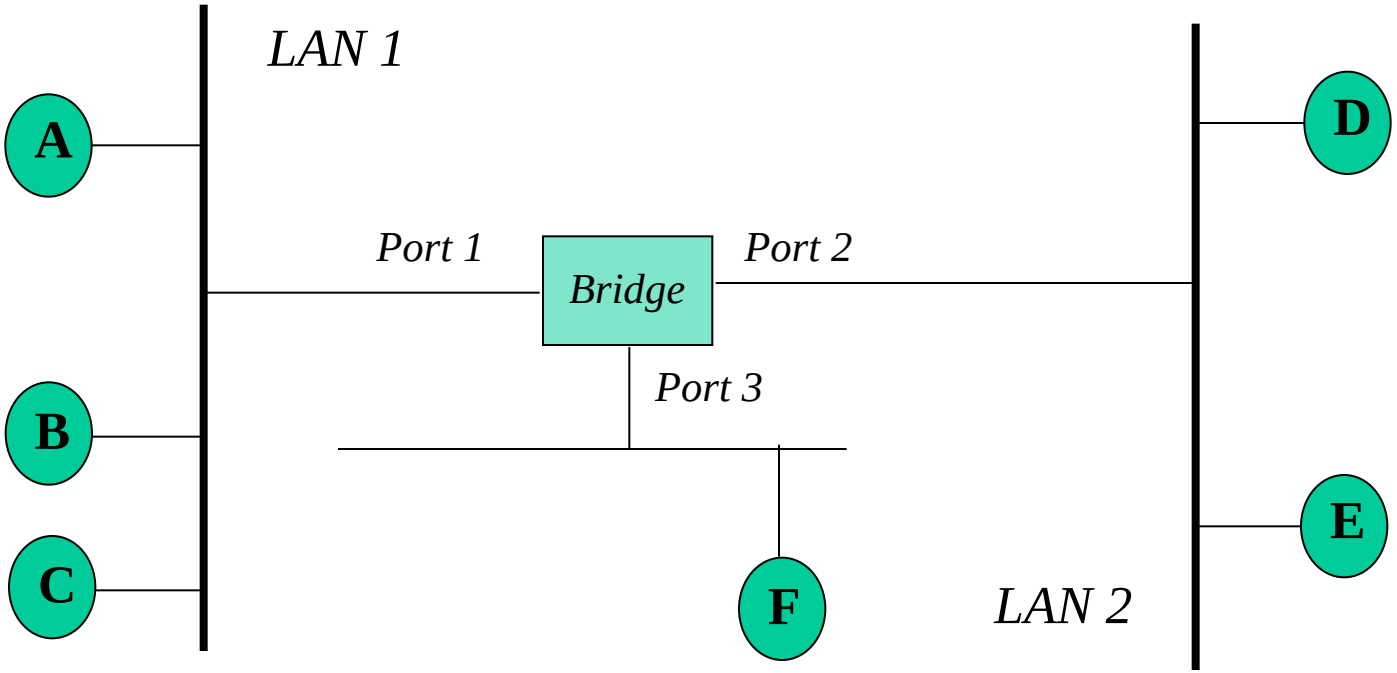
FDDI (obsolete because of 100M /1G Ethernet)

- Fiber distributed data interface
- Optical fiber links at 100 Mbps in a ring
- Spans up to 200 kilometers and supports up to 500 stations
- Was used as a campus backbone network
- Dual ring arrangement often used
- Frame format and operation fairly similar to IEEE 802.5 token ring
- Token insertion strategy is immediately after frame transmission
- Option to have more than 1 token circulating in ring
- Can handle synchronous traffic such as voice and video in addition to asynchronous traffic
  - TTRT: target token rotation time (targeted time agreed by all stations)
  - TRT: token rotation time as measured by each station
  - THT = TTRT – TRT: token holding time to indicate activity on the ring
  - TTRT is sufficient to allow each station to have some dedicated access time S_i. Thus, by calculating THT, the station can determine whether it can transmit only in its own dedicated time or at other times also.

LAN Bridges

- Reasons for bridges
  - Limited number of stations on a LAN segment or ring
  - Limited distance for executing CSMA / CD algorithm or distance one wants a token traveling on a ring
  - Limited traffic on a single LAN: available bandwidth must be shared by all stations
- Interconnecting networks
  - Networks connected at the physical layer are connected by a repeater
  - Networks connected at the MAC or link layer are connected by **bridges**
  - Networks connected at the network layer are connected by **routers**
  - Higher layer interconnection devices that perhaps execute additional functions such as protocol conversion are often called **gateways**
- Bridges
  - Devices for gluing together LANs so that packets can be forwarded from one LAN to the other

An example of a LAN bridge



- Bridge can prevent packets transmitted on LAN 1 from going to LAN 2 or LAN 3 – this is called filtering
- Bridge needs to have packets originating from A to able to get to D. This requires the packet sent over LAN 1 to first be received by the bridge and then *retransmitted* onto LAN 2
- Traffic can concurrently be localized on all LANs
- The bridge operates at the MAC layer because it needs to make decisions based on the MAC address

Simple ideas for bridges

- The no frills bridge : simply transmit all traffic from one LAN segment onto all the other segments
  - Advantages: two stations can be transmitting at the same time. Bridge will buffer a packet until it can transmit on a LAN
  - Disadvantages: total bandwidth that can be safely utilized is still the minimum bandwidth of each LAN segment
- Keeping a database of all stations on each LAN segment
  - Manually enter addresses in such a database
  - Partition addresses into ranges on each LAN
    - Eg. LAN 1 has 1-50, LAN 2 has 51-100, LAN 3 has 101-150
  - Have the MAC address be hierarchically divided into a LAN address and a station address (like the IP address)
  - None of these solutions are really used
- Better solution: the transparent learning bridge
  - Learn on which segment a station resides
  - Transmit a packet only onto the correct segment

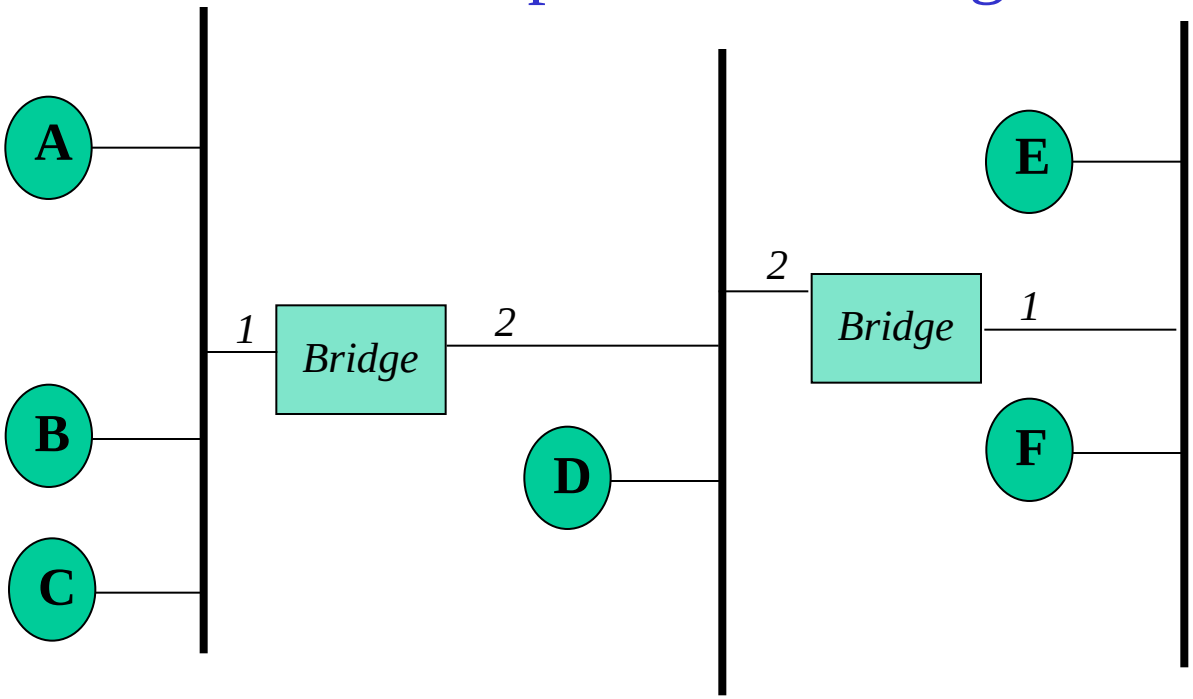
The Transparent or Learning Bridge (backward learning algorithm)

- Maintain a forwarding database or cache of station MAC addresses and the bridge port that the stations are on
- Promiscuously listen to packets arriving on any port
- For each packet arriving at the bridge:
  - Store the stations source address and arriving port in the cache (if an entry already exists for an address, update if different)
  - determine if the destination address is in the cache
    - If entry then forward only on the appropriate port unless the port is the same as the arrival port
    - If no such entry then forward packet on all segments except the one the packet was received on.
  - Age each entry in the cache and delete after an appropriate time

The complete forwarding table of the example

Address	Port
A	1
B	1
C	1
D	2
E	2
F	3

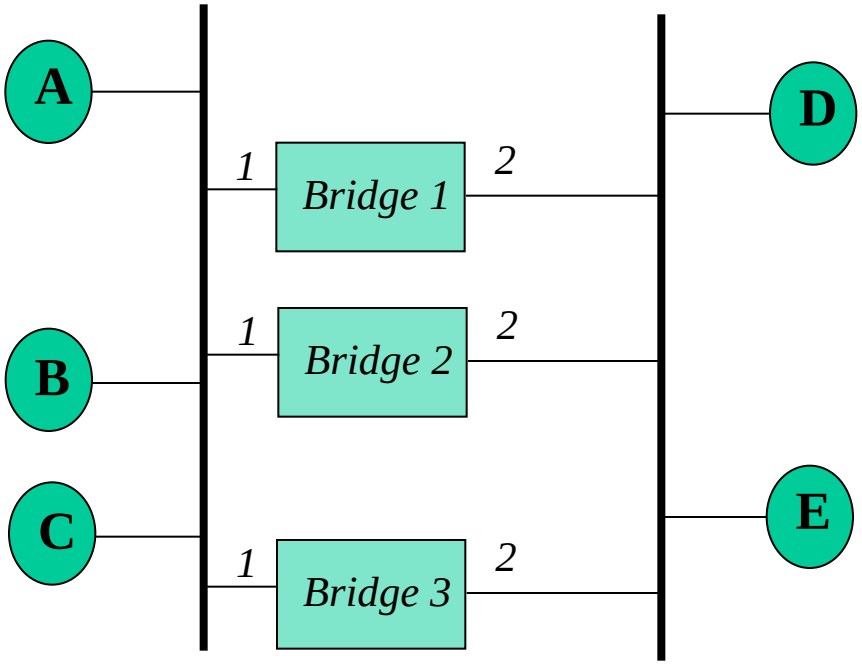
Another example at some stage of learning



Address	Port
A	1
F	2
C	1

Address	Port
A	2
F	1

Interconnection problems: routing loops



What happens when A transmits a packet to E?

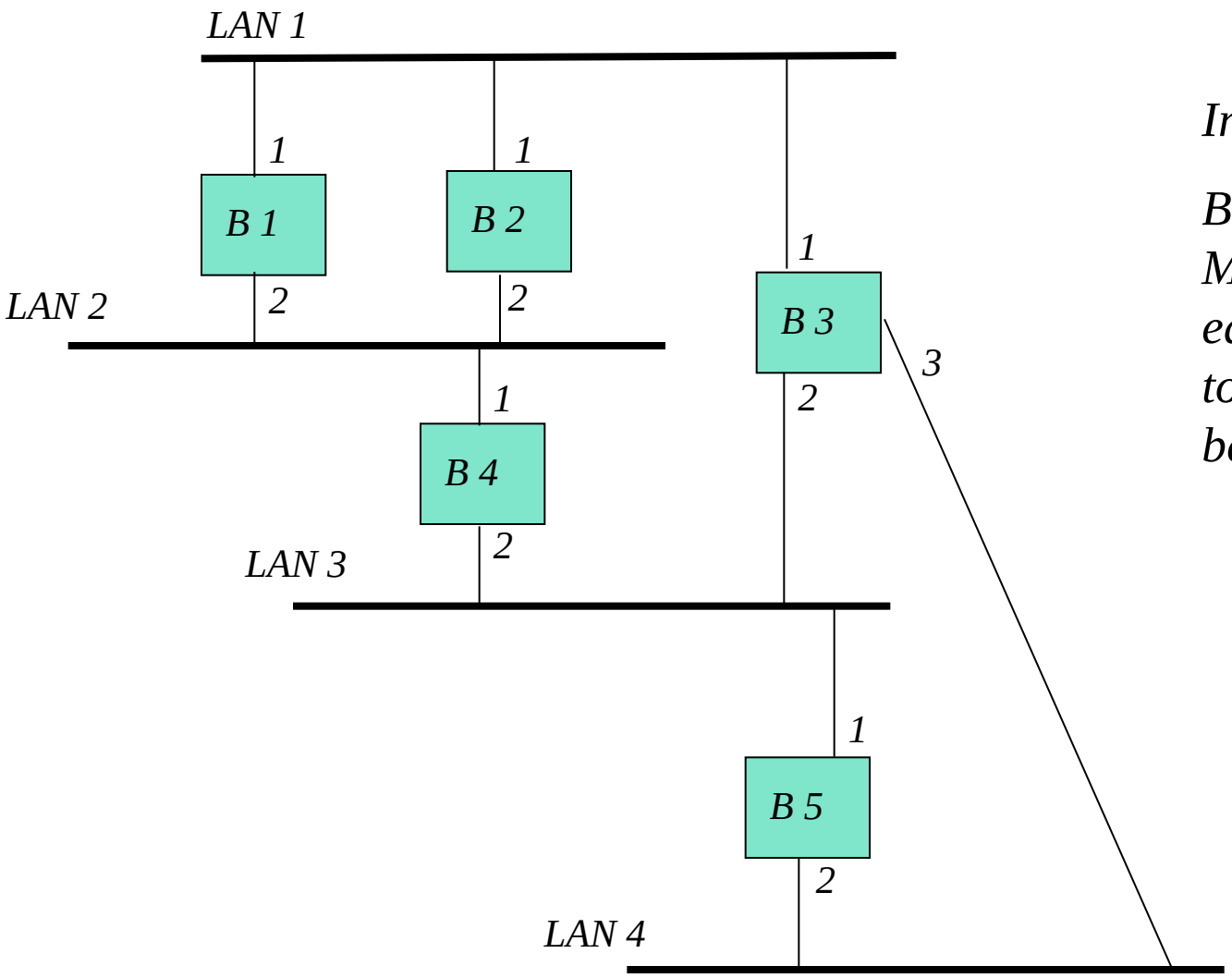
A broadcast storm!

How to fix problem?

Spanning tree algorithm that bridges run that prunes the topology into a loop-free subset so that any two stations are connected but the connection graph is a tree

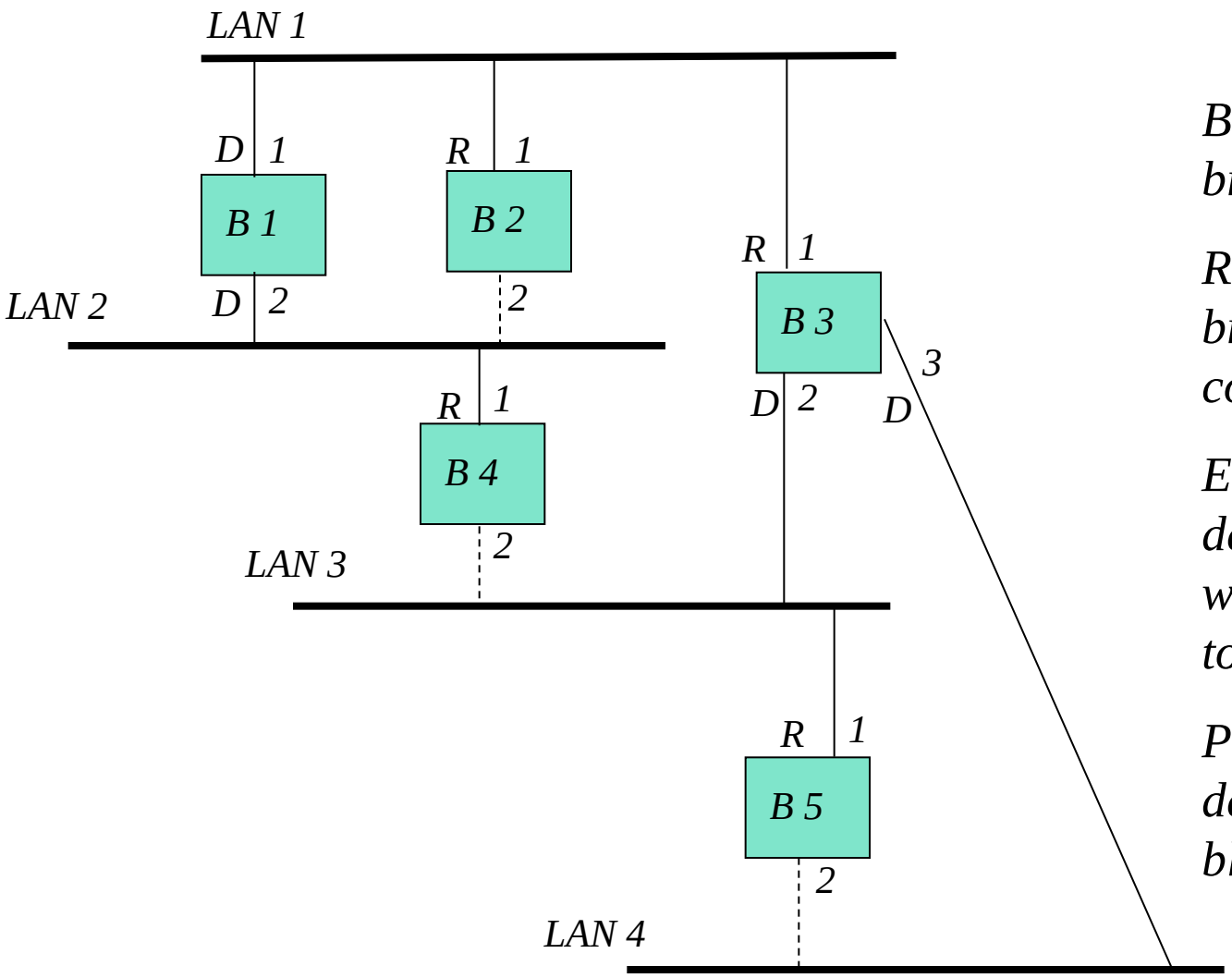
The spanning tree algorithm

- Elect a single bridge among all bridges as the root bridge. The algorithm will select the root bridge as the one with the lowest bridge id. Id are usually a concatenation of a locally defined priority for that bridge and its MAC address.
- Each bridge determines the root bridge and the least cost path (shortest path with respect to some metric, say hops or LAN costs) from itself to the root bridge through each of its ports. The port with least cost to the root is the root port for that bridge. In case of ties use the smallest port id.
- Elect a designated bridge for each LAN from the bridges directly connecting to that LAN. The designated bridge is the one closest to the root bridge. In case of ties it is the one with the lowest bridge id. The port that connects the designated bridge and the LAN is the designated port for that LAN.
- Ports in the final spanning tree are all root ports and designated ports. Other ports are in the blocking state.
- Data traffic is forwarded to and received from ports in the spanning tree only.



Initial Configuration

Bridges have 48 bit MAC addresses on each port connecting to a LAN. Lowest can be the bridge ID

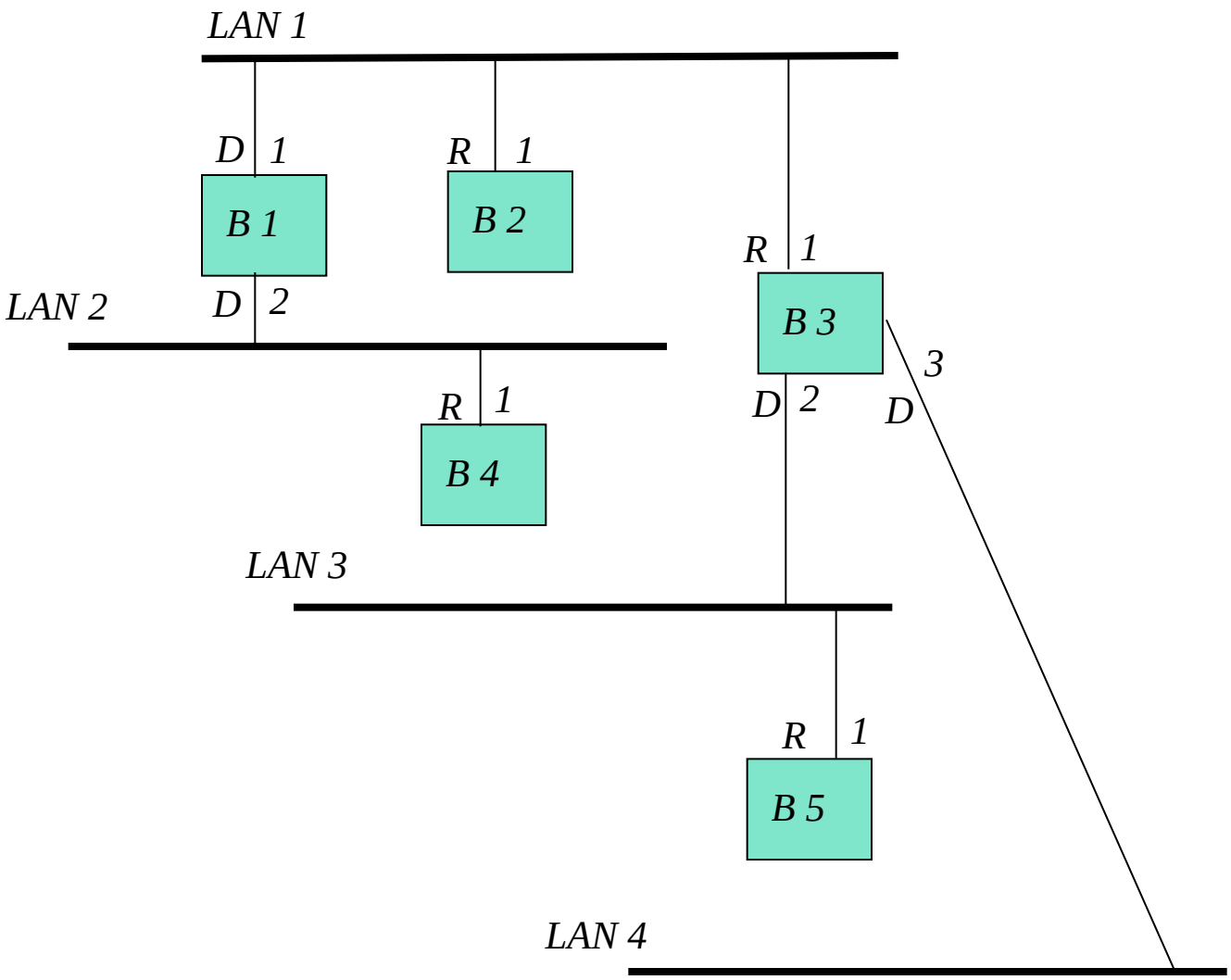


B1 becomes the root bridge.

Root ports for each bridge are R's. (least cost path to root)

Each LAN has a designated port. (one whose bridge is closed to root bridge)

Ports without R or D designations are blocked



B1 is the root bridge.

The set of bridges and LANS are the nodes and the active port connections are the edges in the spanning tree.

Note that there is a unique path from each bridge to any other bridge

D ports face away from the root and forward control and data packets

R ports face root and forward data packets only

How does the ST algorithm work

- Bridges exchange bridge protocol data units (BPDUs). These have *configuration* messages consisting of:
- < Rid, Cost, TBid, Port>
  - TBid: id of the transmitting bridge that sends the configuration message
  - Port: port through which the message is sent
  - Rid: the bridge that TB assumes is the root bridge
  - Least cost path to the root known to the TB.
- All bridges start by transmitting on all ports with the following configuration message:
- < TBid, 0, TBid, Port>
  - I think I am root at this point
  - The cost to get to me is 0
  - I am TBid
  - This is the port the message is being sent on

- Bridges compare messages received. How are messages compared?
  - First compare Rid, lower is better
  - If tie, compare Cost, lower is better
  - If tie, compare TBid, lower is better
  - If tie, compare Port, lower is better
- A bridge receives configuration messages from neighbor bridges on its ports. It saves the best configuration messages received or sent (by itself) on each port. Configuration messages received are stored as received. Using these best configuration messages, it can calculate the best known root and path known so far and the BPDU that it would send. It can also decide whether or not to continue transmitting BPDU on a port.

- Suppose a bridge with Bid 18 has the following best messages on its 4 ports:
  - Port1 <12, 86, 14, 6>
  - Port2 <12, 85, 47, 3>
  - Port3 <81, 0, 81, 3>
  - Port4 <15, 31, 27, 5>
- Bridge 18 determines that 12 is the root, and best cost is 85.
- Bridge 18 (adds LAN cost on that port, say1, to the cost for comparisons) and determines its new BPDU is <12, 86, 18, Port> over Port. It also determines that its Root Port is 2.
- Bridge 18 stops transmitting over a port if the saved best configuration is better than what it would transmit over that port.
- Note also that 18 is the designated bridge on ports 3 and 4, and that port 1 is blocked since the saved configuration is better that its own (but not the root port. Also, 18 would transmit its new BPDU over ports 3 and 4 and replace the previous values with its own BPDU.

- Eventually only the root bridge is transmitting, and a spanning tree has been determined.