Access Technologies

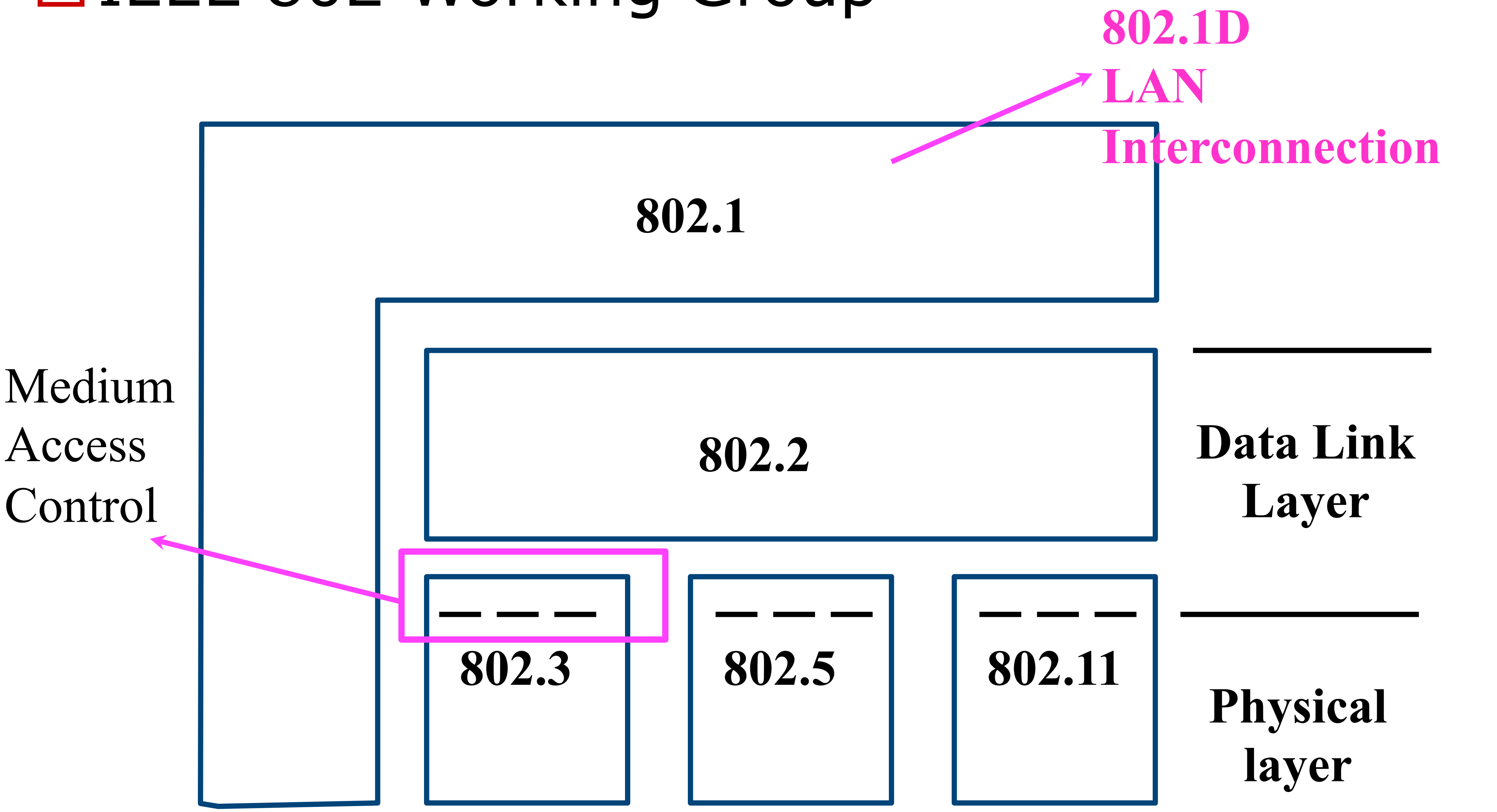
Fabio Martignon

LAN Ethernet - IEEE 802.3

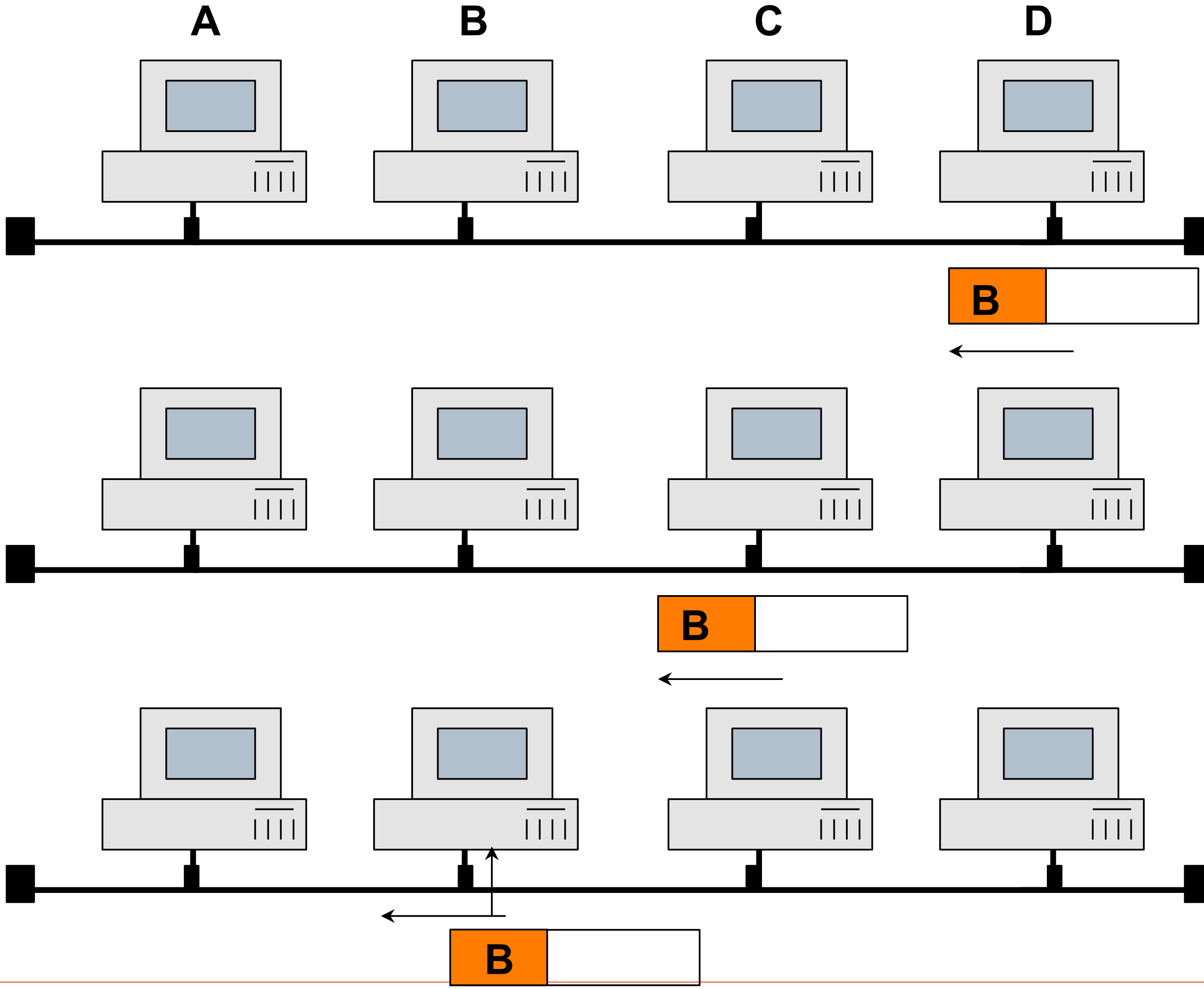
- Broadcast Bus Capacity=10 Mb/s
- Xerox-Intel-Digital inventors
- Standardized at the beginning of the 80s as IEEE 802.3
- Big Success and Several Extensions

Local Area Networks

- IEEE 802 Working Group

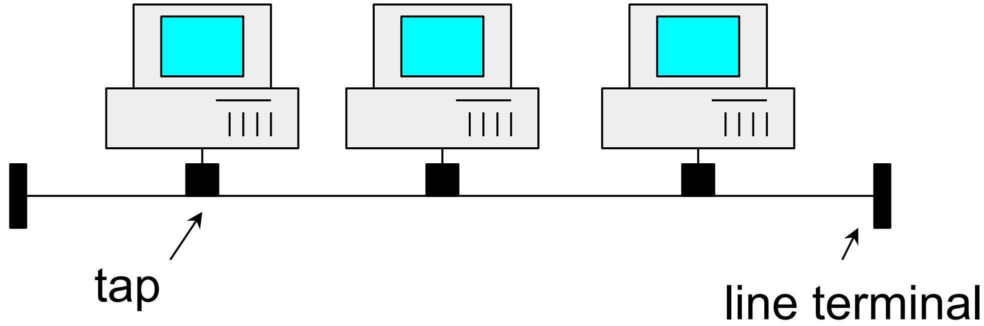


Addressing (on a broadcast medium)

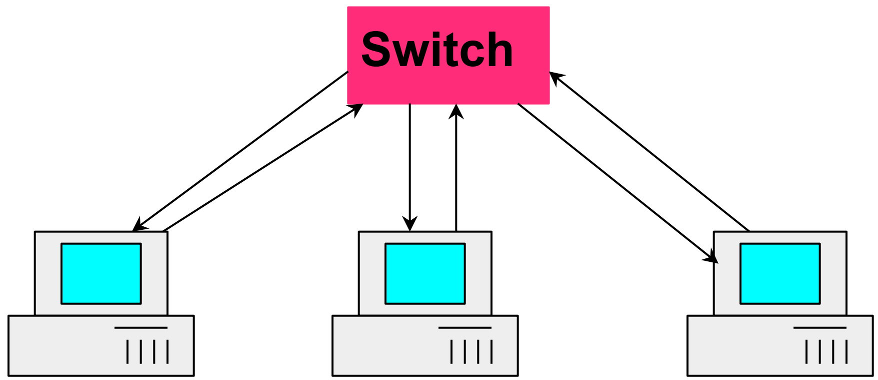


Topology

Bus Topology (historical)



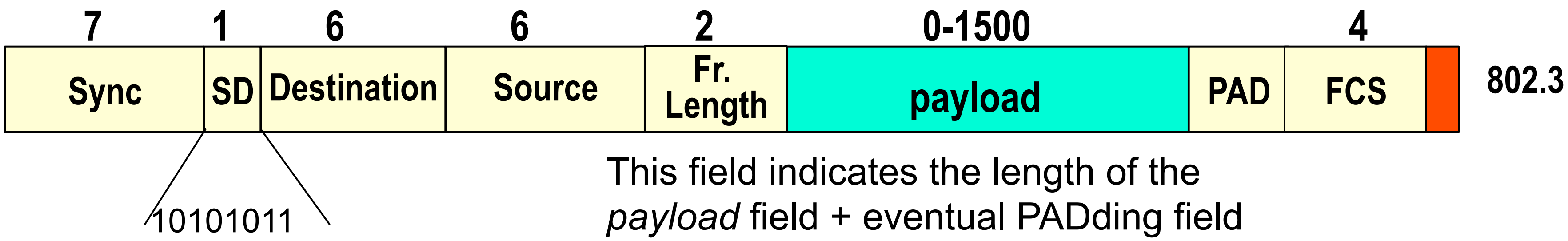
Star Topology



- The *switch* is like a repeater: whenever a frame is received from a station, it is transmitted towards *all other* stations

IEEE 802.3 Frame

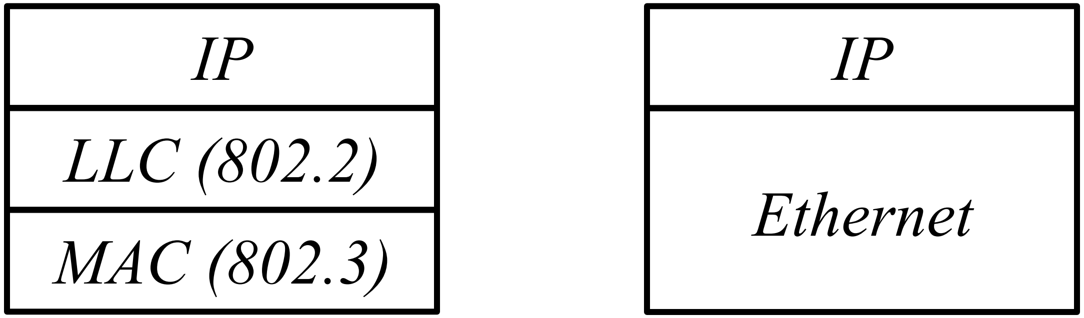
- Ethernet Frame



- Frame minimum length = 512 bit (1 slot) equivalent to 51.2 us
- Propagation speed 2 x 10^8 m/s (5 us/Km)
- Maximum LAN diameter = 2.5 Km

802.3 vs Ethernet

- They're not the same:
  - 802.3 has a LLC (802.2)
  - ethernet is directly connected with the network layer
- E.g.:



802.3 vs Ethernet

- The *protocol* field in Ethernet is used to identify the network SAP
- In many LANs Ethernet and 802.3 coexist.
  - The *Frame Length Field* can be in the range 0-1500
  - The *protocol* field is >1500 *(to be precise, the standard says ">1536", decimal notation, which means 0600 in hexadecimal)*

7	1	6	6	2	0-1500		4		
Sync	SD	Destination	Source	Frame Lenght	payload	PAD	FCS		802.3

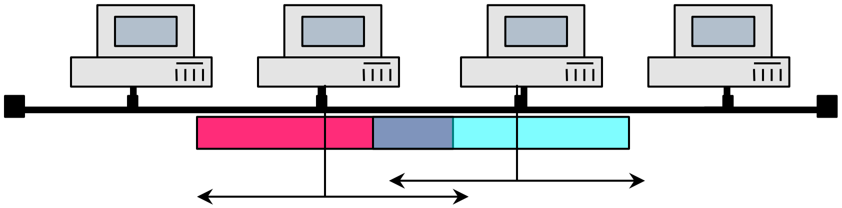
7	1	6	6	2	0-1500		4		
Sync	SD	Destination	Source	Protocol	payload	PAD	FCS		Ethernet

Note: in this latter case, the standard MAC says that it is the MAC client protocol (e.g., IP or the upper level that uses Ethernet) that must operate correctly in case Padding is introduced at the MAC layer (in other words, the correct functioning is demanded at the upper layer)

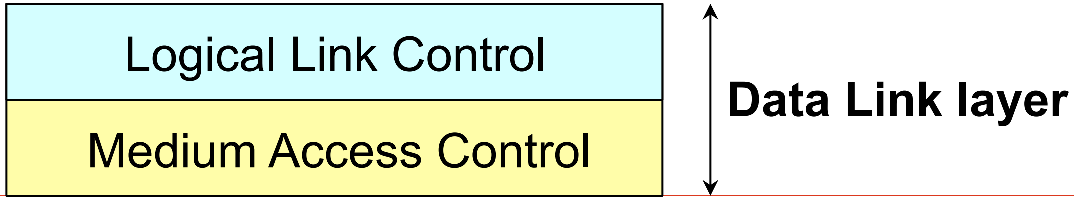
Access Protocols

- **<u>Problem:</u>**

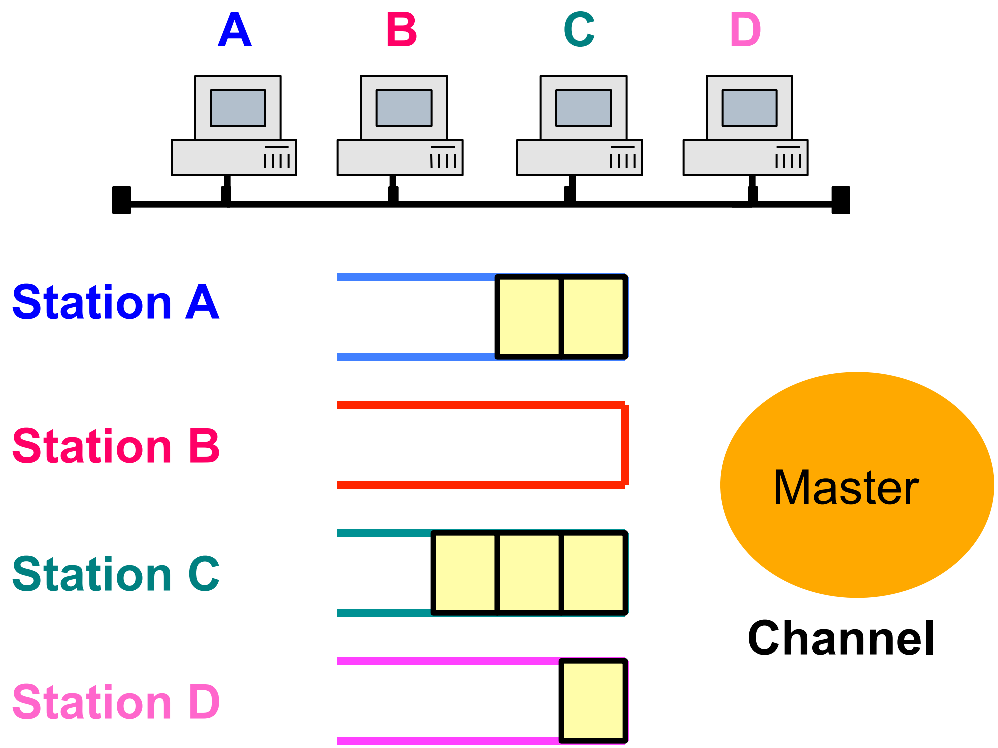
If two (or more) stations try to transmit at the same time, we have a **collision** → the signal (hence, the frames) is not received correctly



- We need *protocols* (rules) to control the access to the broadcast medium, in order to avoid (or at least, to limit) collisions
- If collision occur, they must be correctly *<u>individuated</u>* by all stations, in order to re-send the collided frames
- This function is performed in Ethernet at the <u>MAC</u> (Medium Access Control) sublayer



Conceptual Model of Multiple Access

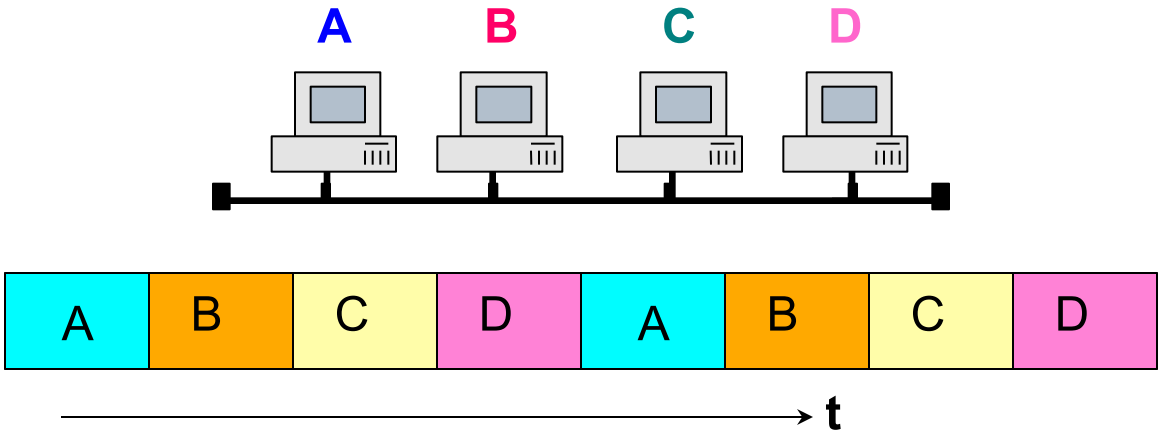


- The Master does not know *if* and *how many* packets are present in each queue (i.e., if and how many packets are produced by each station)
- Each station does not know the state of other stations' queue

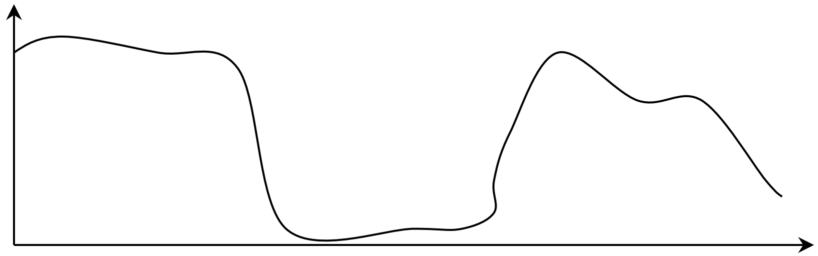
Multiple Access Techniques Classification

- We have two types of access techniques
  - Ordered Access
    - TDMA
    - Round Robin
    - Polling
      - Roll Call Polling
      - Hub Polling
  - Random Access
    - CSMA/CD (Ethernet)
    - CSMA/CA (IEEE 802.11, WiFi)

Example: TDMA



In LANs, taffic is **bursty**, and we have **several stations**



TDMA is <u>inefficient</u>: **high delays, low throughput**

Example: Round Robin

- Each station has, in each round, the <u>opportunity</u> to transmit
- When it is the turn of the station:

<u>if she has no packet to transmit:</u>

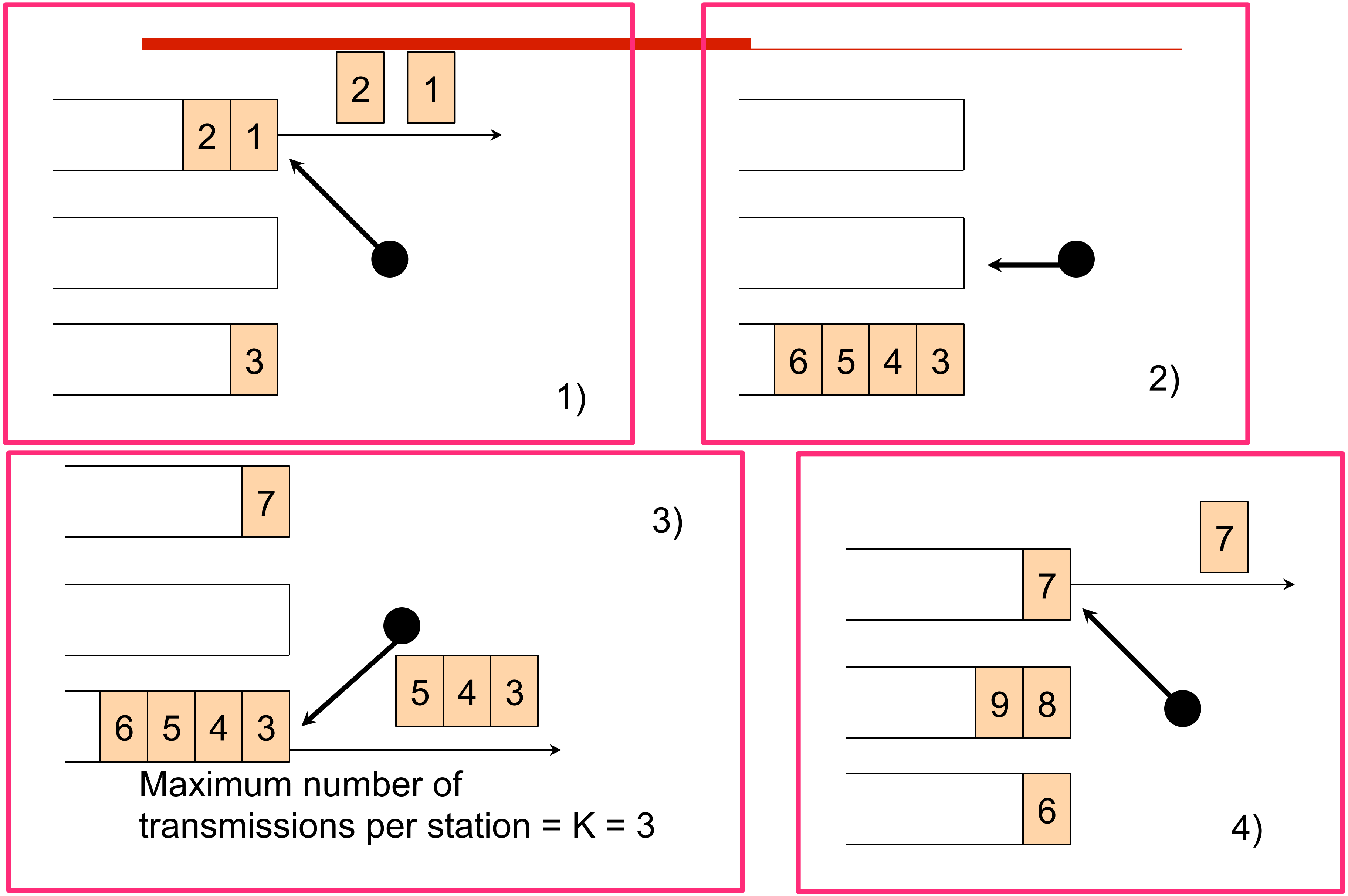
she declines the transmission opportunity, which will be given to the subsequent station

<u>if she does have packets to transmit:</u>

she transmits her packets up to a maximum number (K), defined by the protocol itself

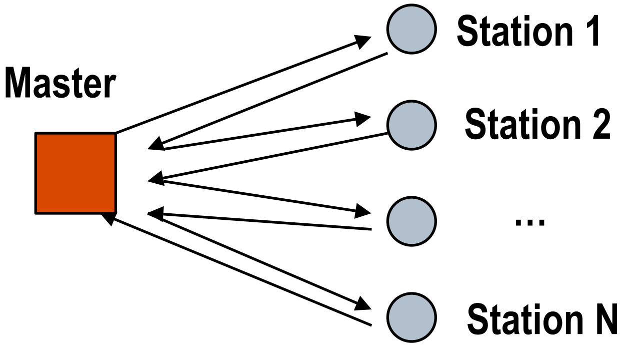
Then, the transmission opportunity (the right to transmit) is sent to the subsequent station

Example: Round Robin



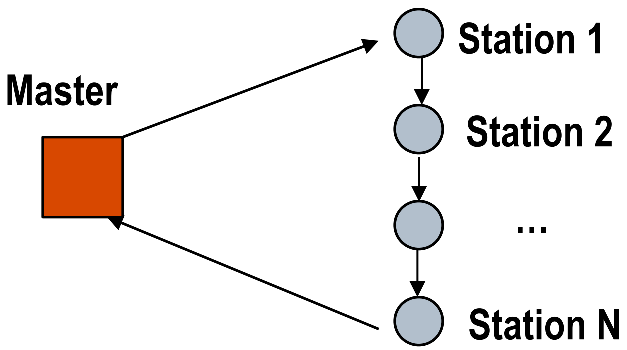
Ordered Access: Polling

- Roll-call polling



◆The token (the control packet which guarantees the transmission opportunity) is always sent back to the Master station

- Hub polling



◆The token comes back to the Master only at the end of the cycle

Random Access Protocols

- Random access protocols do not have an explicit coordination among stations…

- … hence, collisions may occur

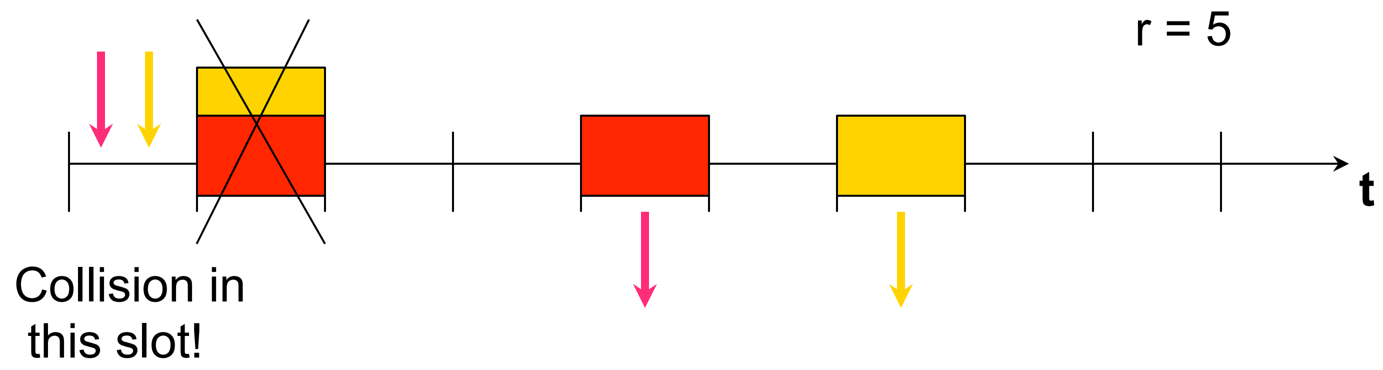
- They differ in *how* they resolve collisions …

- … and also in the feed-back from the channel (i.e., the information that derives from listening to the channel)

- Collisions are overcome by introducing a random mechanism

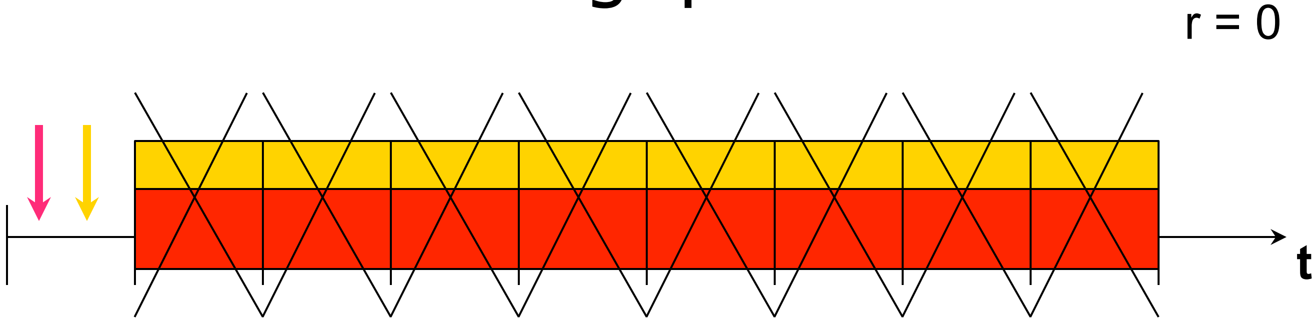
Random Access Protocol Example: Slotted Aloha

- Slotted channel (time is divided into slots)
- When a packet arrives at a station, she tries to send it in the first available slot
- If a collision occurs, the station tries to re-send it after a random number of slots …
- … such random number is chosen uniformly at random in an interval [0,r]



Slotted Aloha: Collision Resolution

- If r = 0 → collision repeats infinite times!
  → throughput = 0



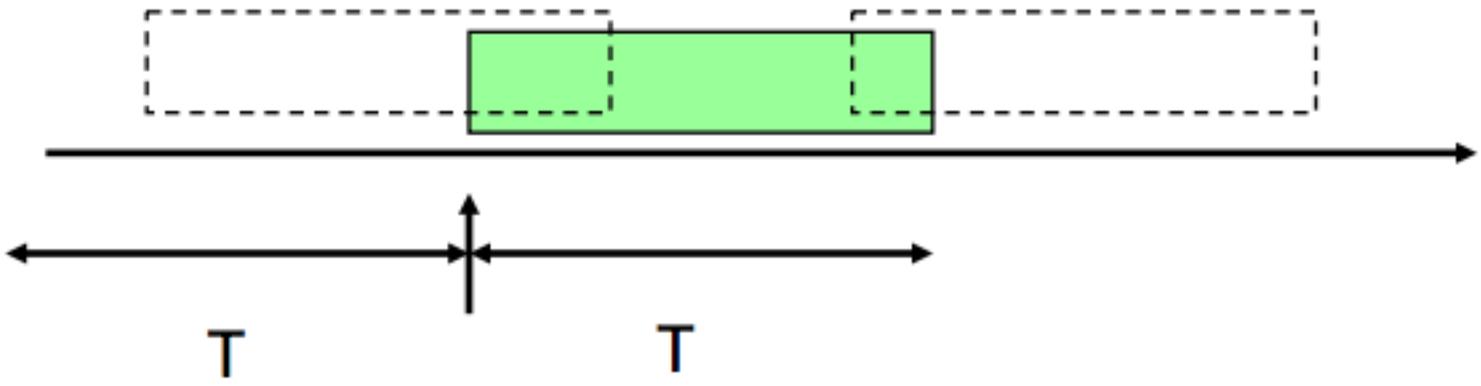
- If the offered traffic is high, we need a high *r* value to avoid instability

To summarize: we would like to have small *r* values when the network is empty and large *r* values when the network is congested !!!

Aloha: Performance analysis

In Aloha, the access mechanism is very simple:

- When there is a packet to be transmitted, just transmit it.
- If transmission fails, wait for a random time and retransmit



Aloha: Performance analysis

- Let us assume the transmission starting times on channel are a Poisson process with rate λ
- Let us consider the normalized rate $G=\lambda T$
- The success probability is given by the probability that there is no other transmission in a $2T$ interval

$$P_s = e^{-2G}$$

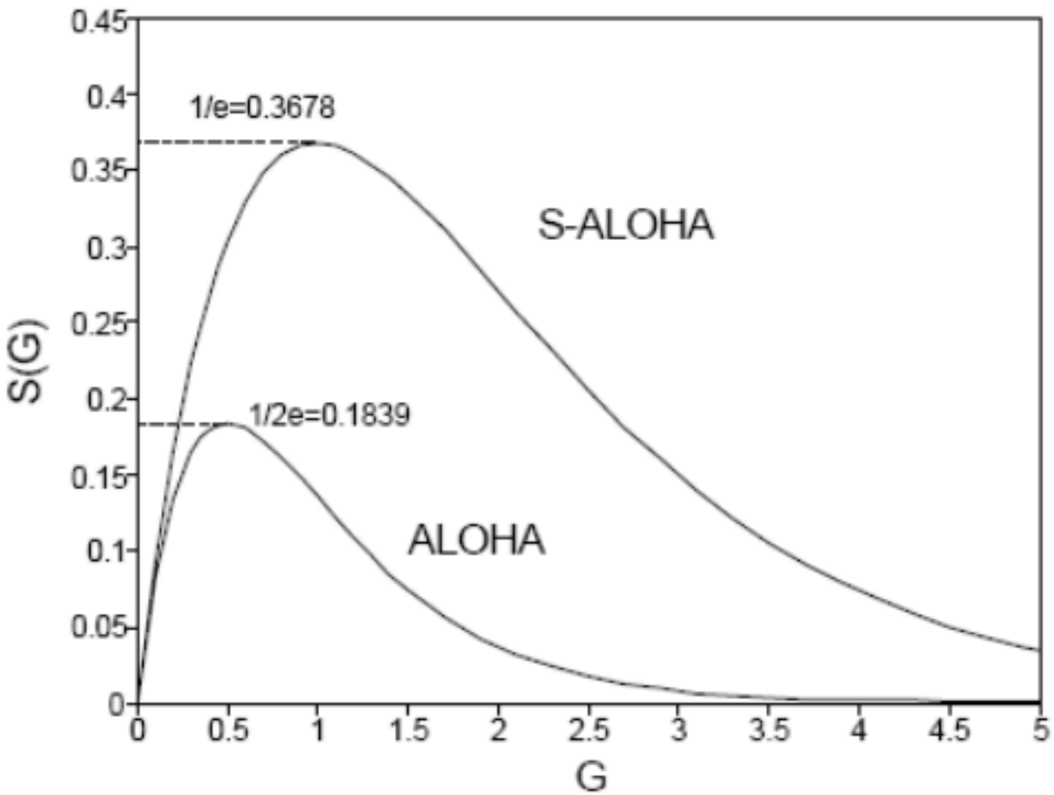
- The normalized throughput S is therefore given by:

$$S = Ge^{-2G}$$

Aloha and Slotted Aloha: Performance analysis

- If transmissions are somehow synchronized (slotted Aloha) the vulnerability period reduces to T and therefore

$$S = Ge^{-G}$$



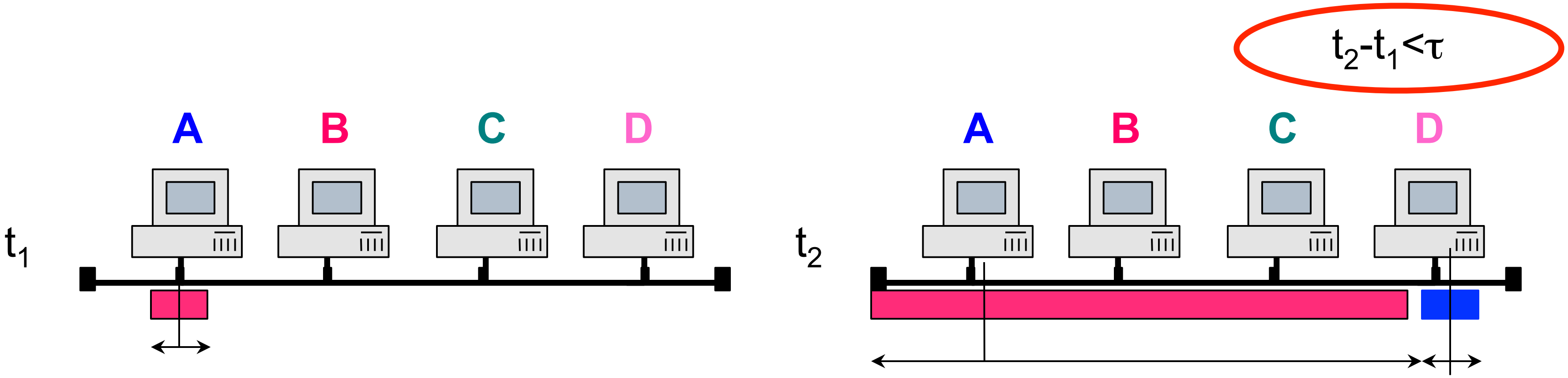
Infinite population model

Aloha and Slotted Aloha: Performance analysis

- Unfortunately, the traffic on the channel is the combination of new transmissions and retransmissions, and it can therefore increase if throughput reduces
- To evaluate the dynamic behavior of Aloha, it is necessary to consider enhanced models

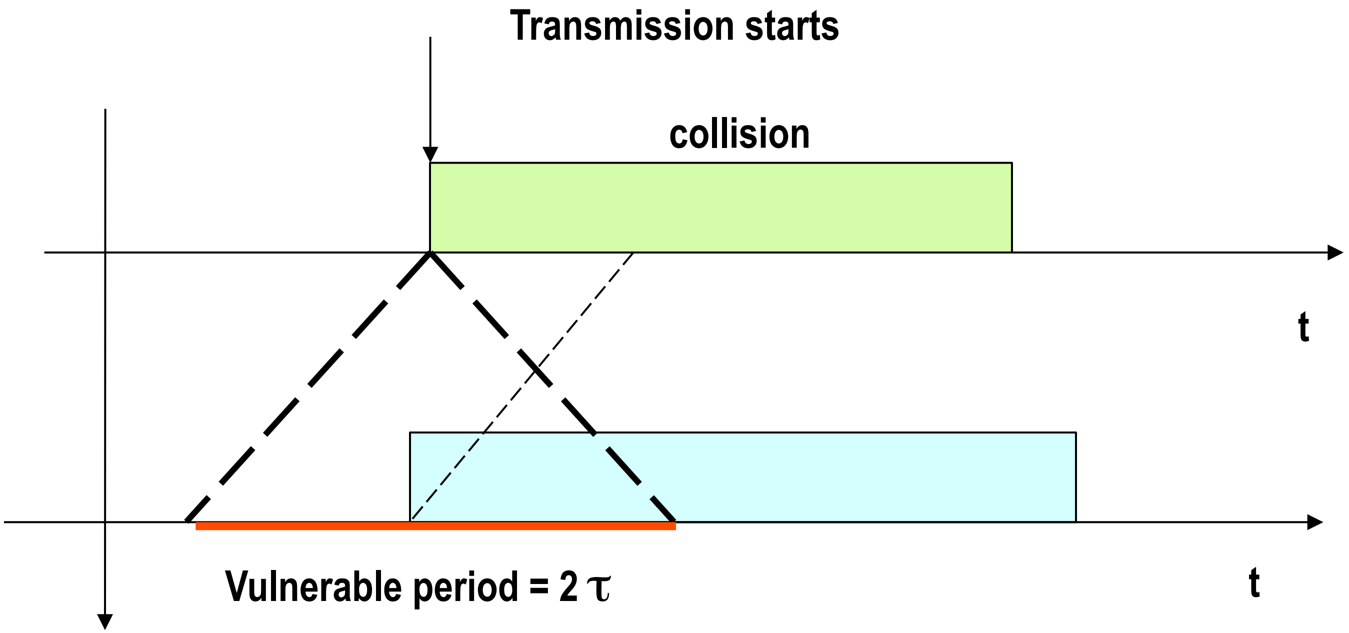
Carrier Sense Multiple Access

- CSMA has been created for systems in which the station can *listen* to the channel (Carrier Sense)
- Transmission is possible only if the channel is sensed free (*listening before transmitting*)
- Collisions are still possible due to the so called *vulnerable period*



t_1, t_2 : instants at which the fartherst stations (A and D) start transmitting a frame after having seen that the channel is (apparently) free

Vulnerable Period



τ : propagation time between the two farthest stations (A and D)

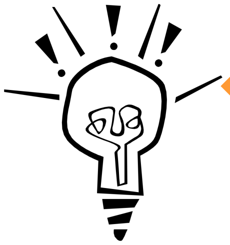
T : frame duration, must be larger than 2τ

Variations to Carrier Sense

- If a station senses the channel and finds it already active (i.e., not free):
  - the transmission is postponed after a random time (just as if a collision occurred) (***non persistent***)
  - the transmission starts immediately when the channel becomes free again (***persistent***)
  - with probability *p* the station uses the persistent approach, with 1-*p* the non persistent one (***p-persistent***)

CSMA- Collision Detect (CSMA/CD)

- In some channels (e.g., wired ones) it is possible for a station to discover if a collision occurred
- The time necessary for <u>all</u> stations on the bus to see that a collision really occurred depends on the propagation time (which is smaller than the frame transmission time in LANs)



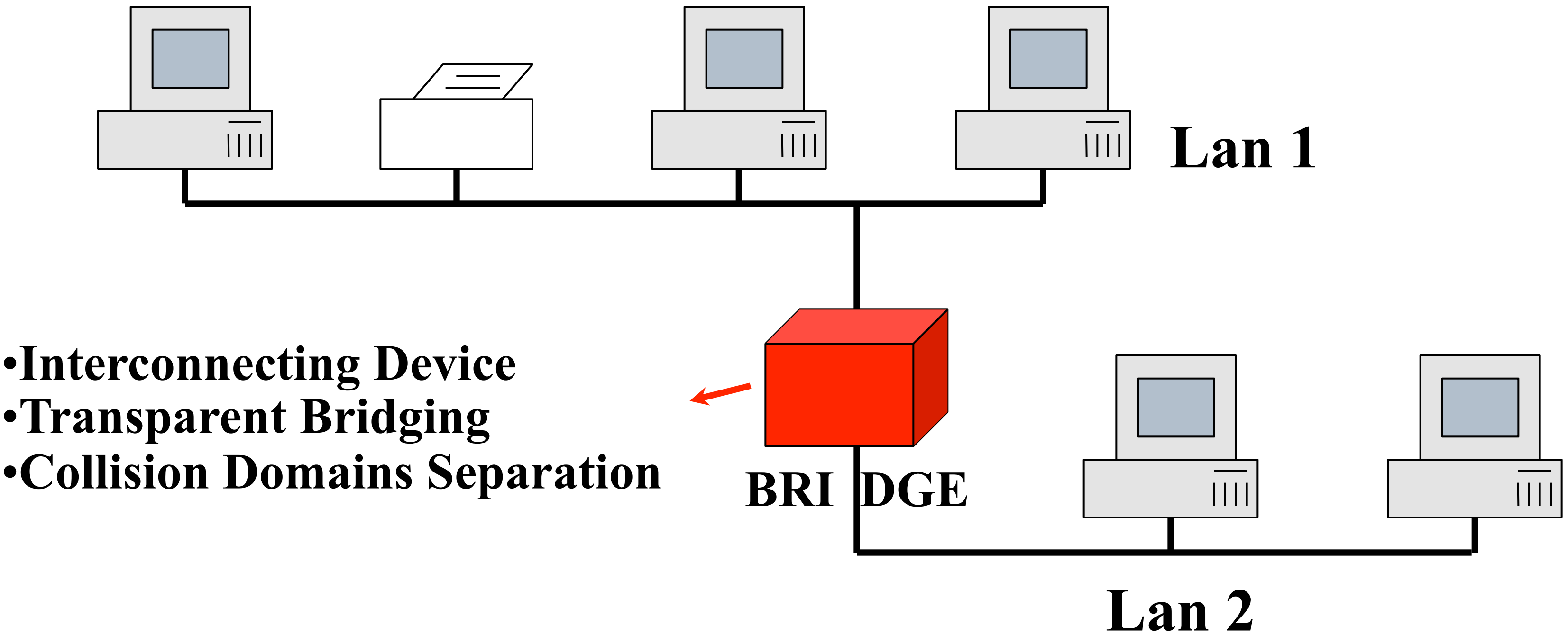
- Why continue transmitting after the station knows that the frame experienced a collision?
- Idea: whener the station knows that a collision occurred, she stops immediately sending the rest of the frame

→ CSMA-CD

Ethernet - IEEE 802.3 Protocol (CSMA/CD)

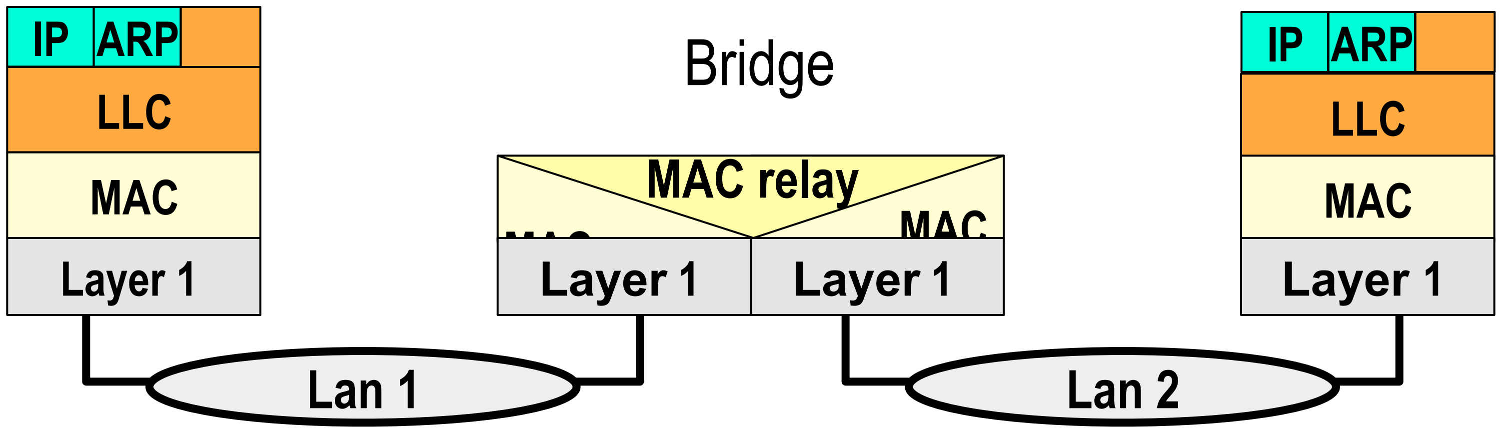
- If the channel is sensed free, the transmission is performed
- If the channel is busy, transmission is refrained; transmission happens as soon as the channel is free again (*persistent*)
- If a *collision* happens, transmission is aborted after transmitting 32 more bits of *jamming sequence*
- After a collision, the next transmission is attempted after X time slots
- X is randomly chosen between 0 and $2^{\min(K,10)}$ K number of consecutive collisions, K<=16 *(exponential binary backoff)*
- After 16 failed attempts the frame is dropped

Interconnecting Local Networks



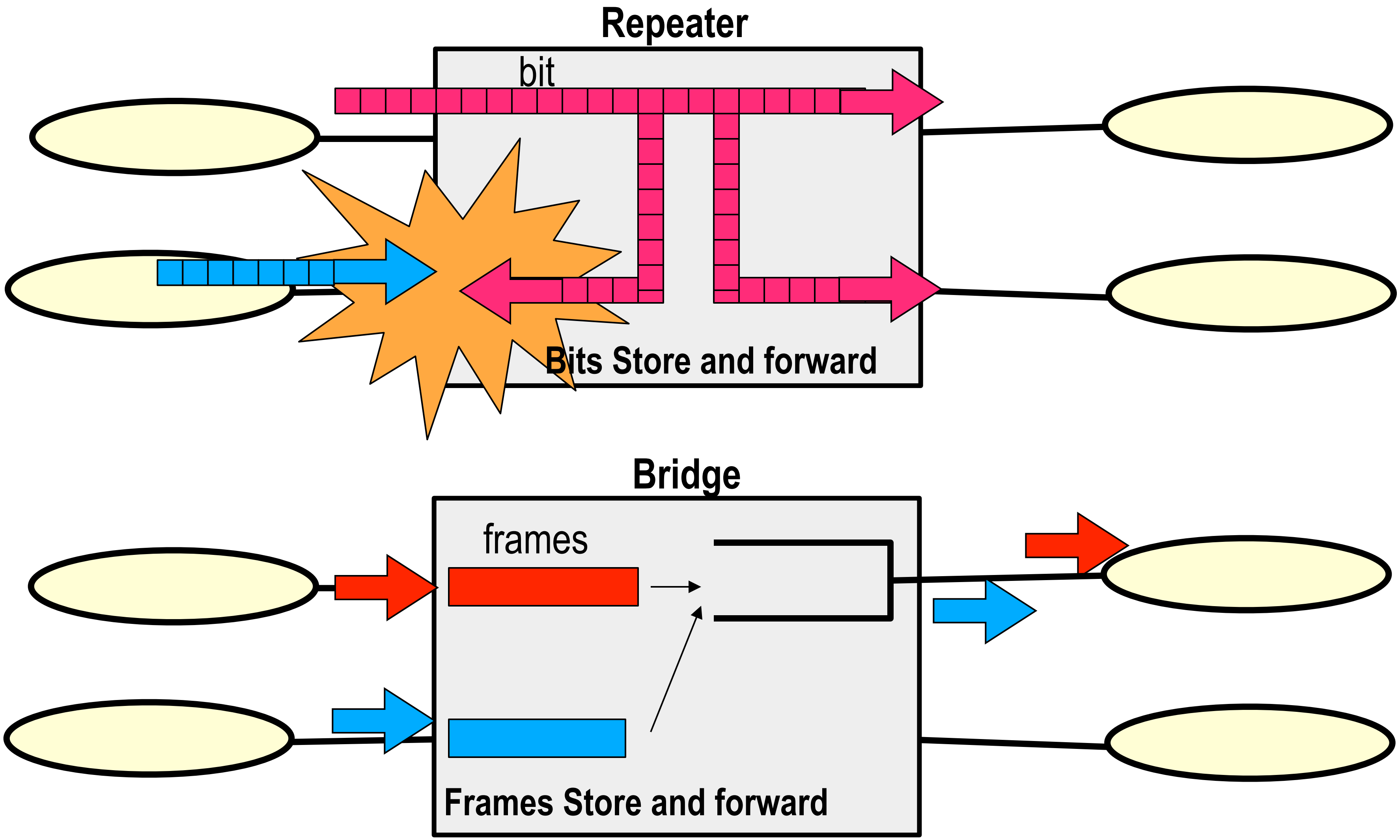
Single Broadcast Domain, **Different** Collisions Domains

Bridge



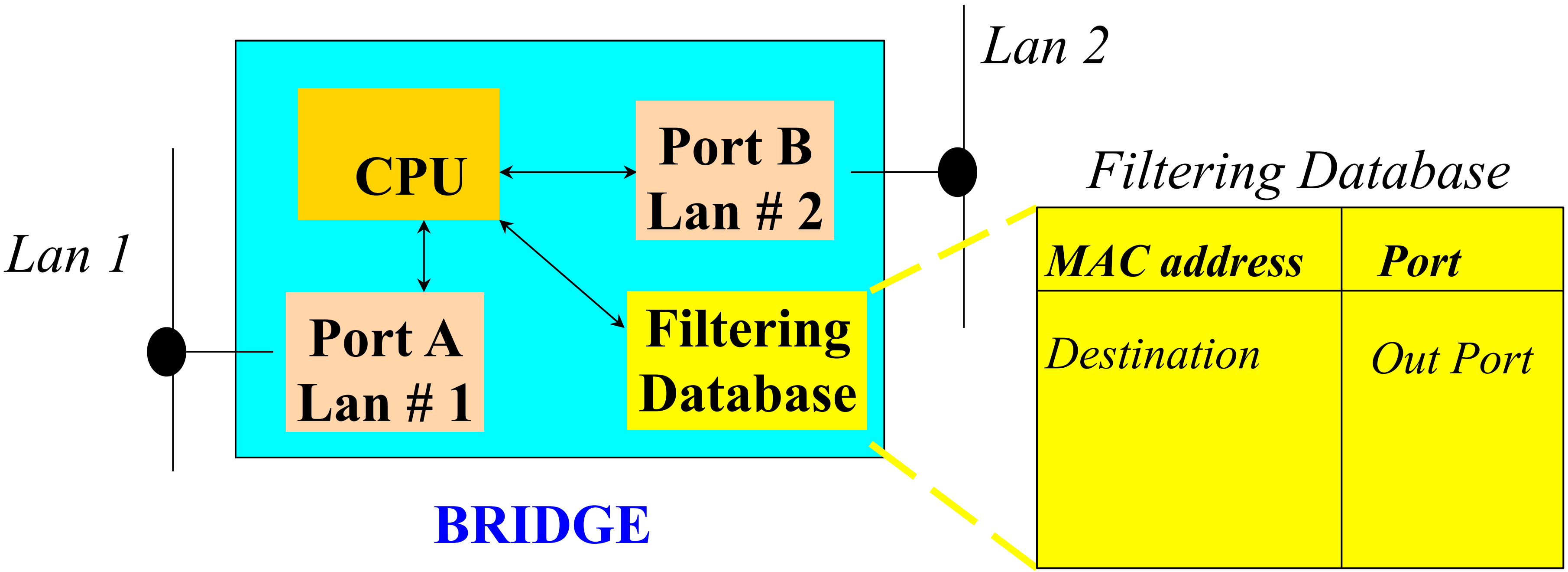
- Functionalities:
  - *Filtering:* if a frame generated within LAN 1 is destined to LAN 1 it remains confined within LAN 1
  - *Relaying:* if a frame originated within LAN 1 is destined to LAN 2 it is relayed by the bridge (possible MAC translation)

Repeaters & Bridges

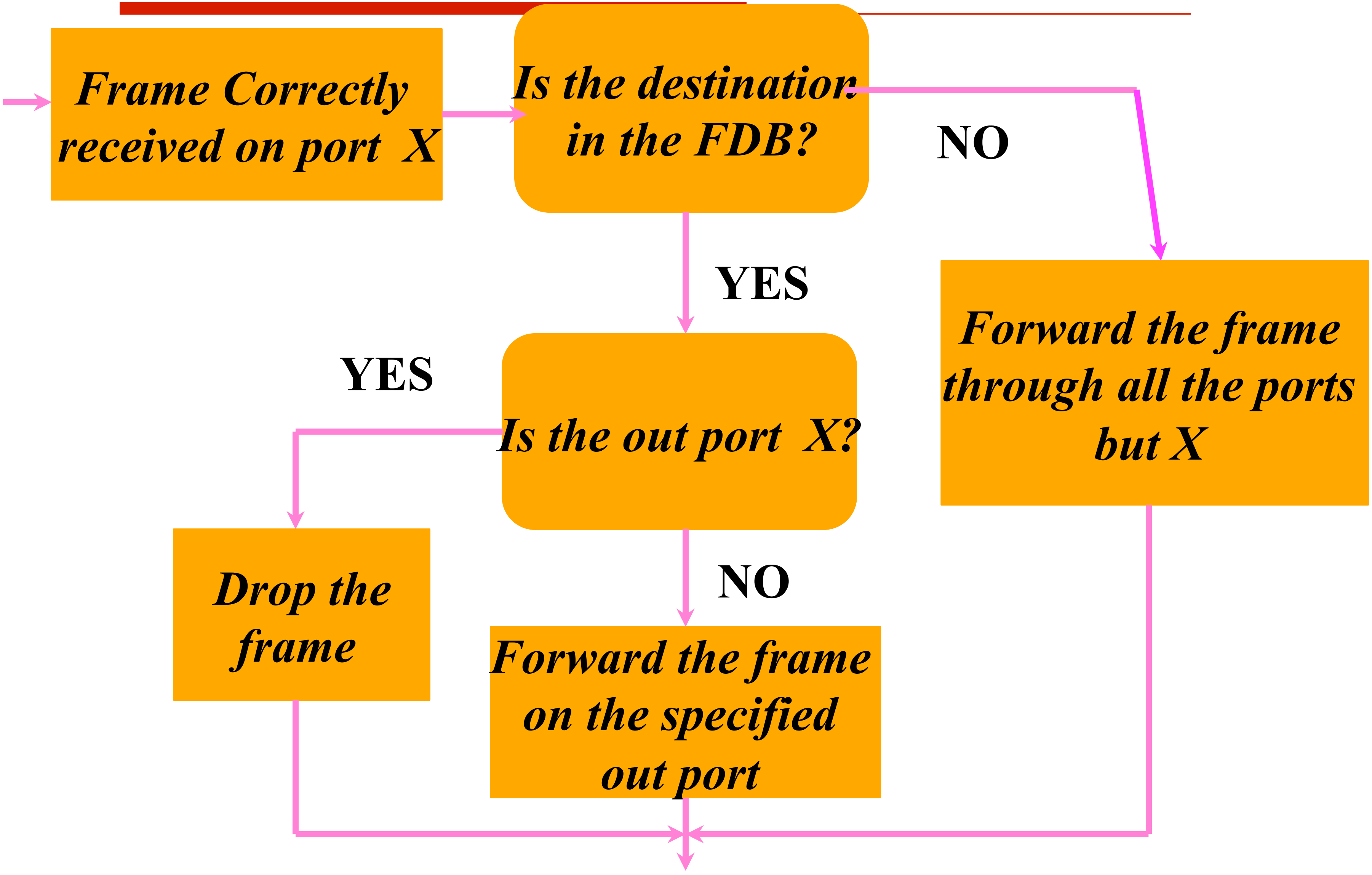


Bridge Architecture

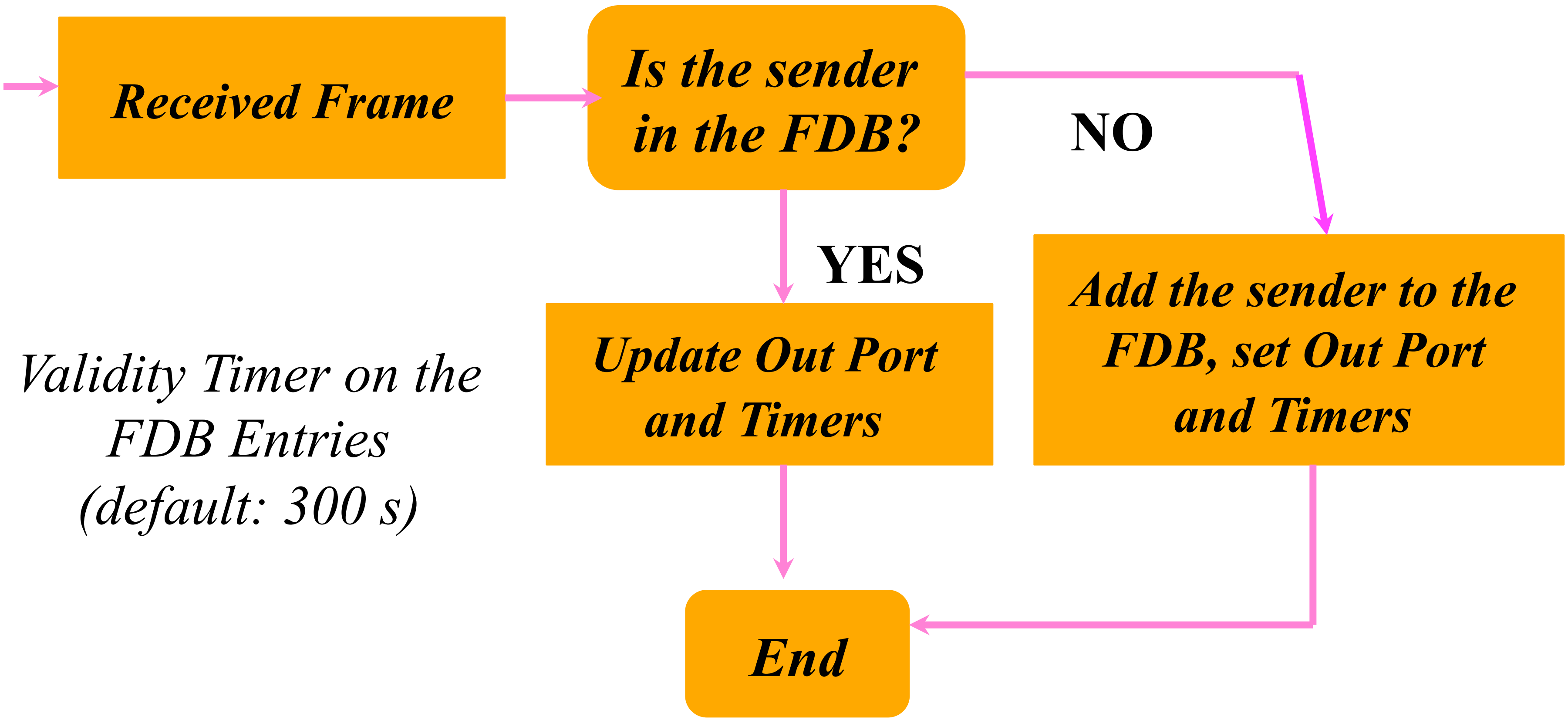
- Filtering and Relaying are performed according to a local *Forwarding Data Base* (or *FDB*)



Bridge Forwarding

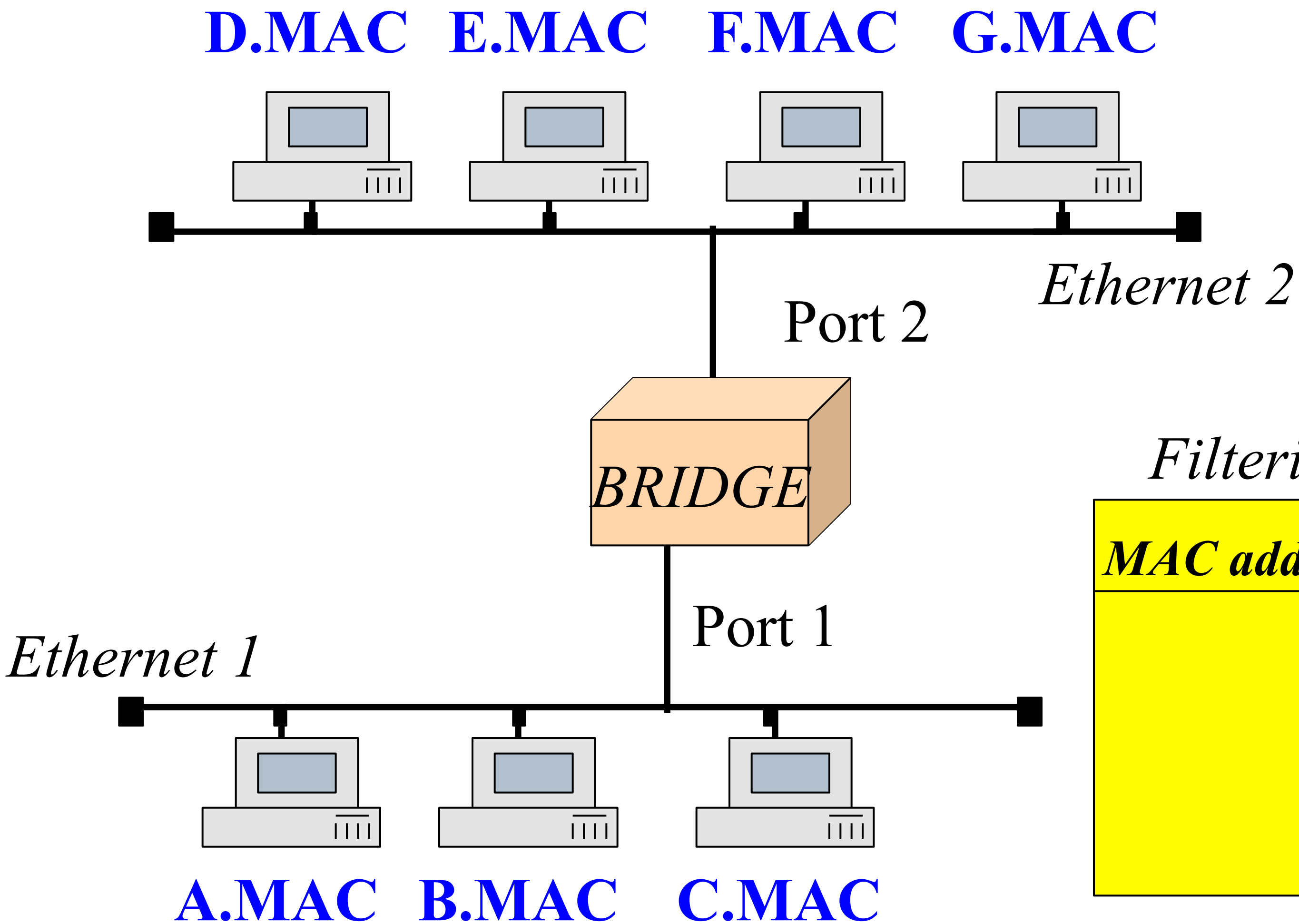


How to Fill/Update the FDB: Backward Learning



Validity Timer on the FDB Entries (default: 300 s)

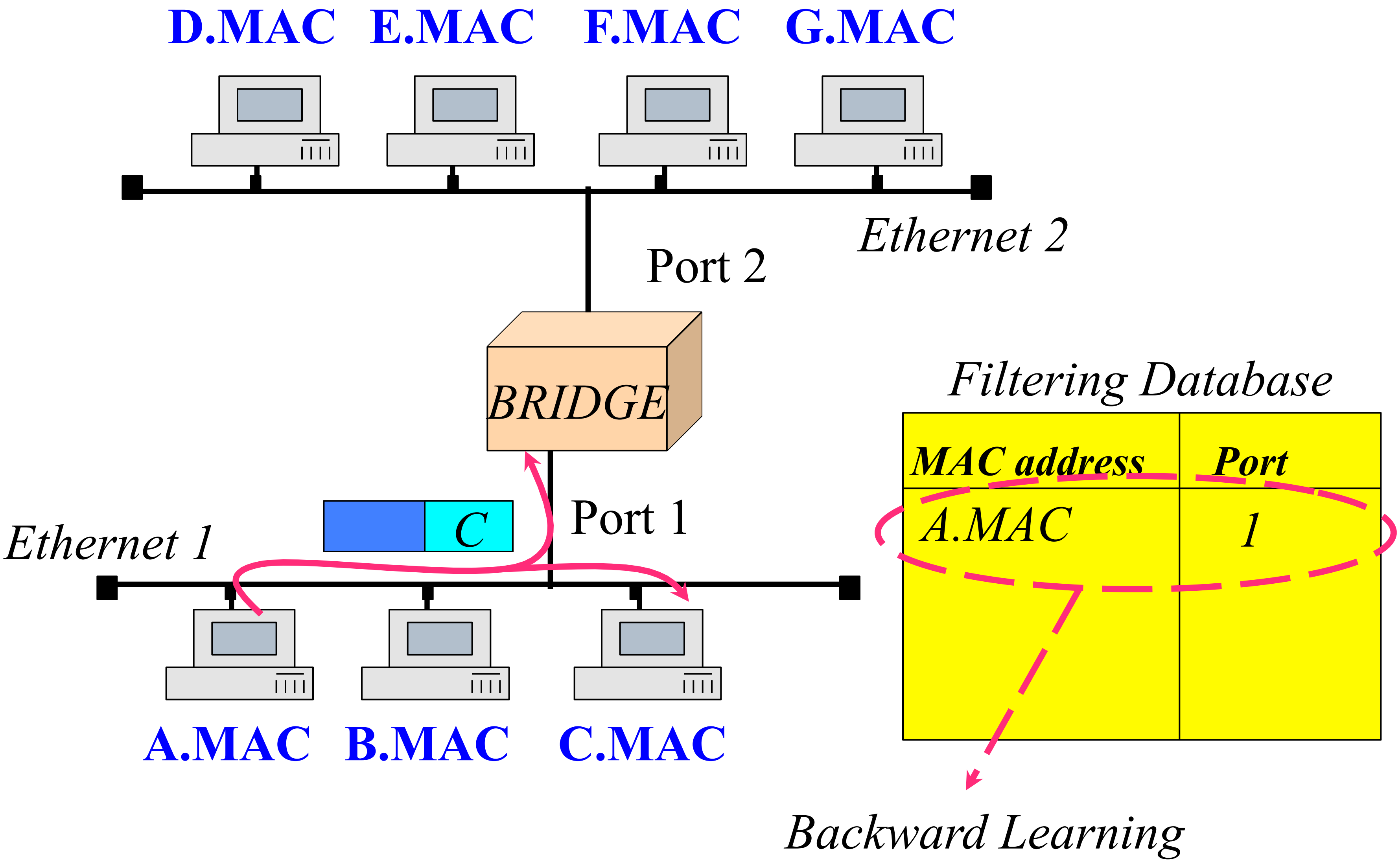
An Example



Filtering Database

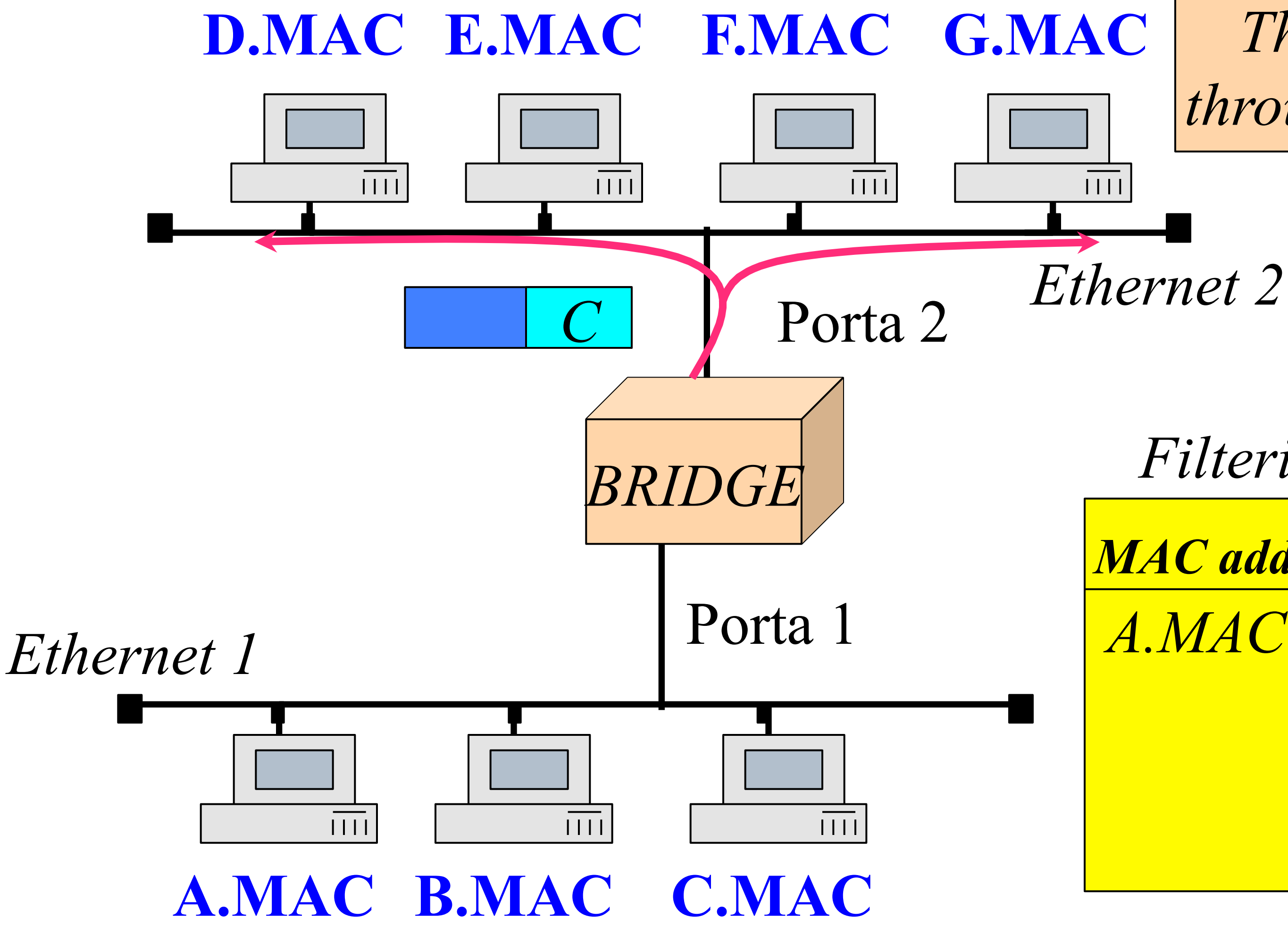
MAC address	***Port***

An Example


D.MAC
E.MAC
F.MAC
G.MAC
Ethernet 2
Port 2
BRIDGE
Port 1
C
Ethernet 1
A.MAC
B.MAC
C.MAC
Filtering Database
MAC address
Port
A.MAC
1
Backward Learning

An Example

Since C is not in The FDB
The frame goes through all the ports

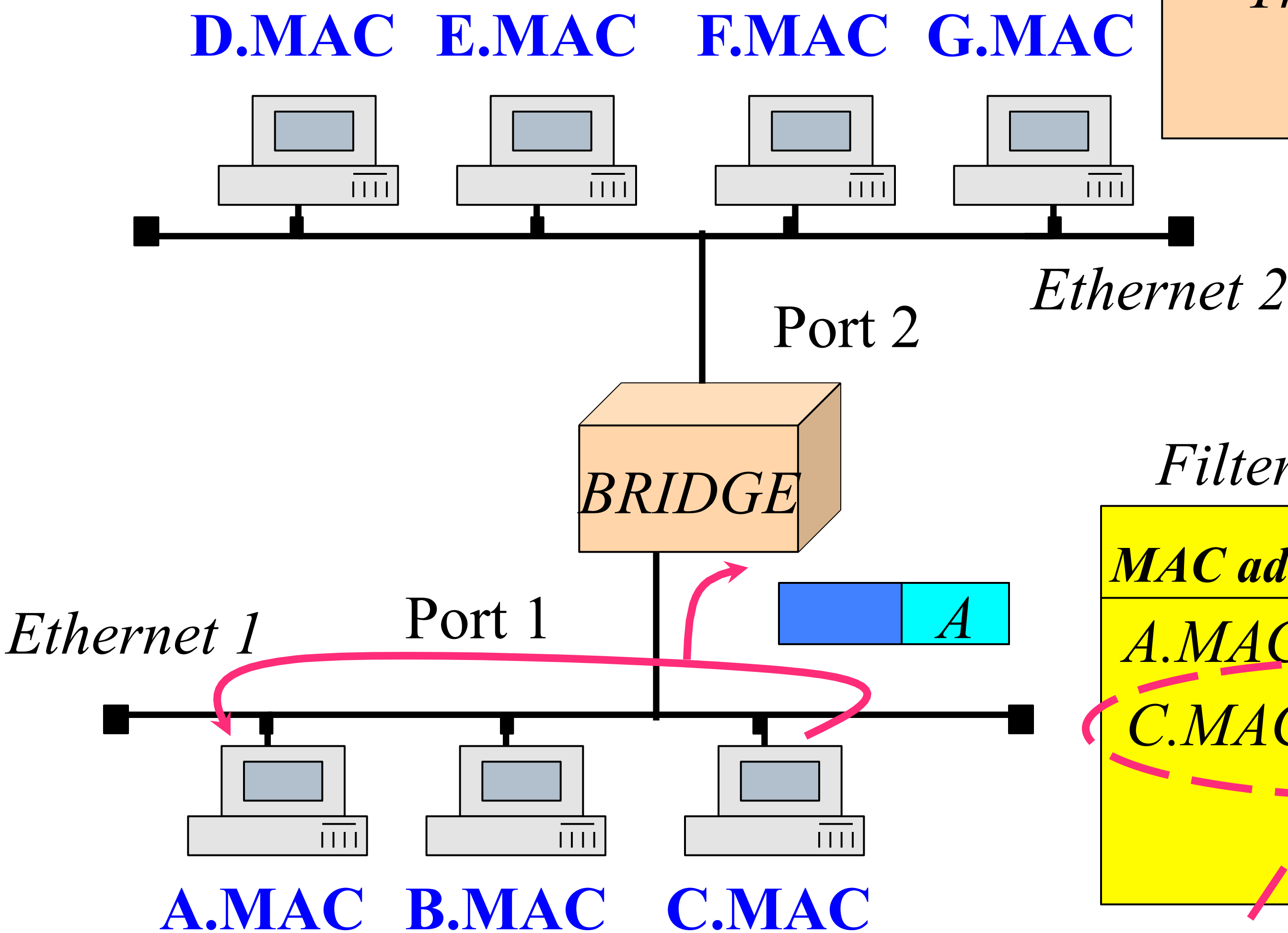


Filtering Database

MAC address	***Port***
A.MAC	*1*

An Example

A is reachable through The same port 1: filtering

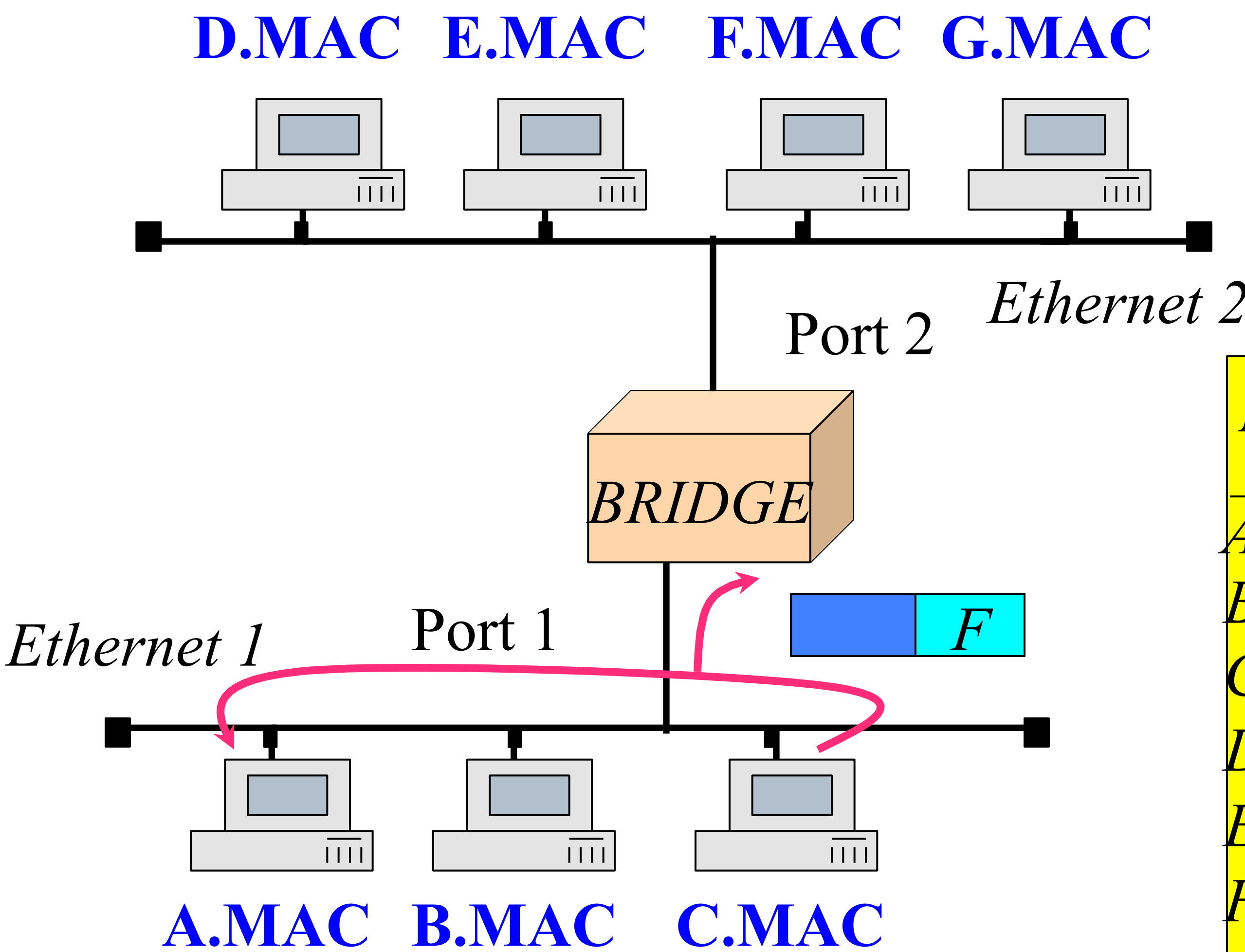


Filtering Database

MAC address	***Port***
A.MAC	*1*
C.MAC	*1*

Backward Learning

Complete FDB

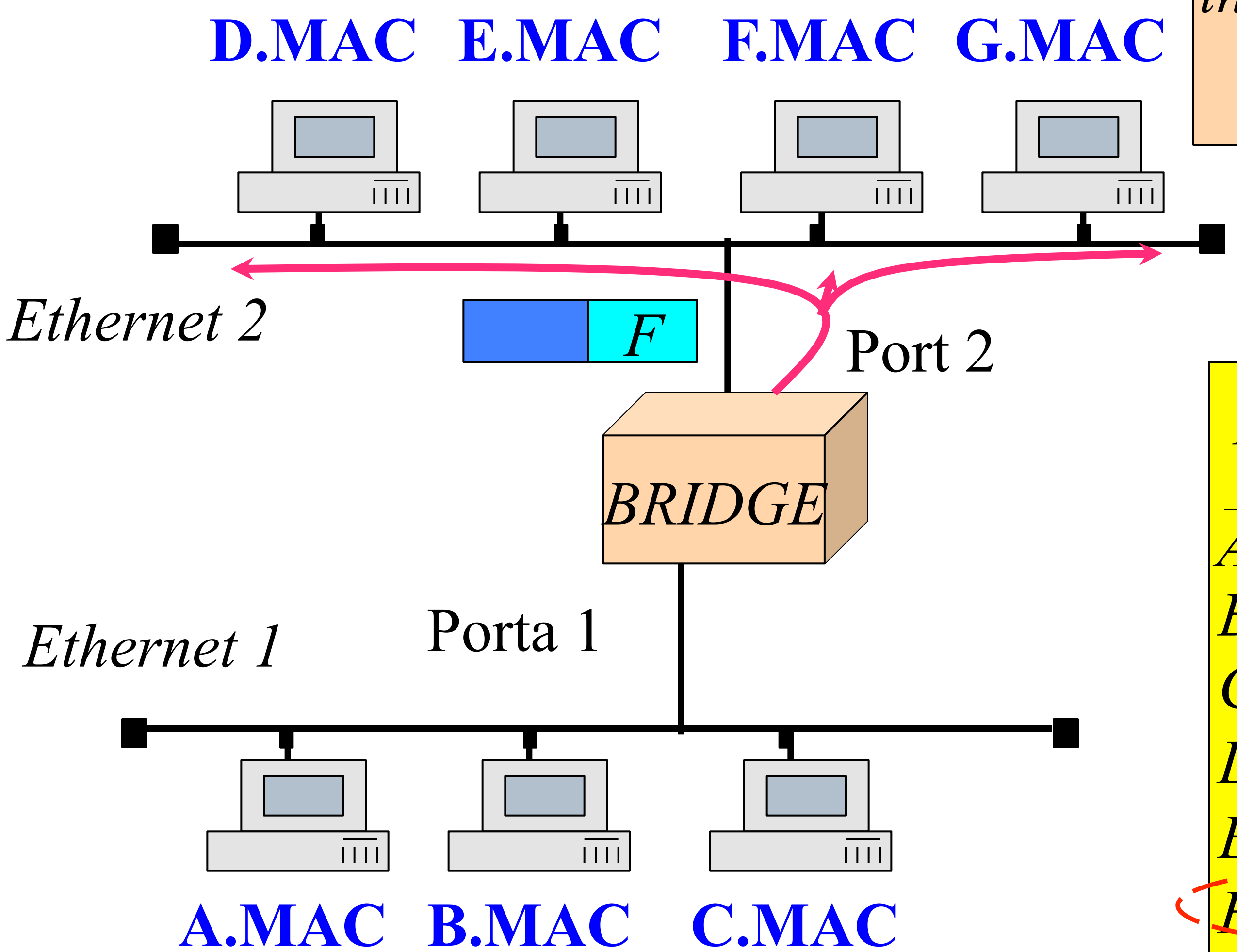


Filtering Database

MAC address	***Port***
A.MAC	*1*
B.MAC	*1*
C.MAC	*1*
D.MAC	*2*
E.MAC	*2*
F.MAC	*2*
G.MAC	*2*

An Example

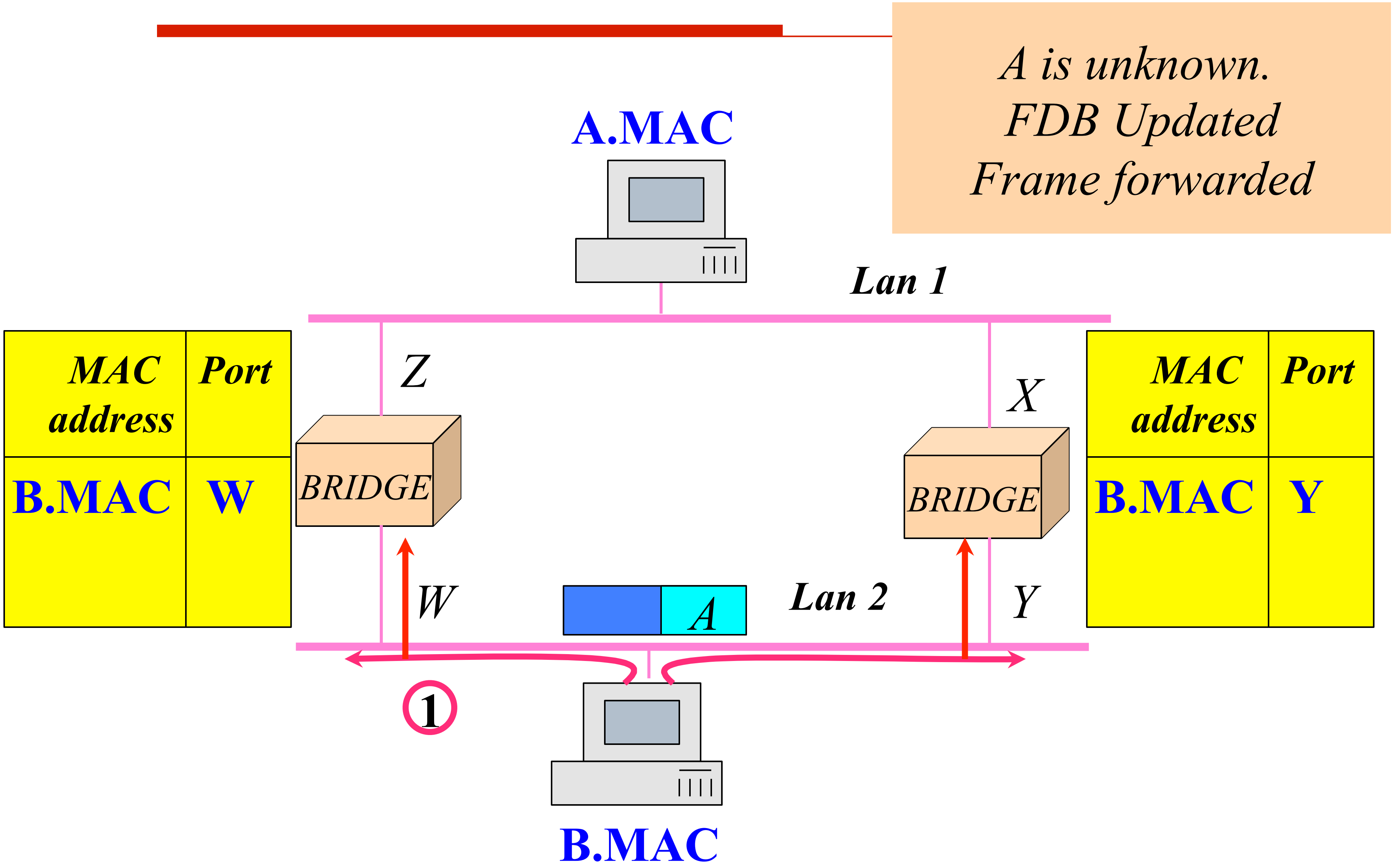
F is not reachbale through the same port 1 forwarding



Filtering Database

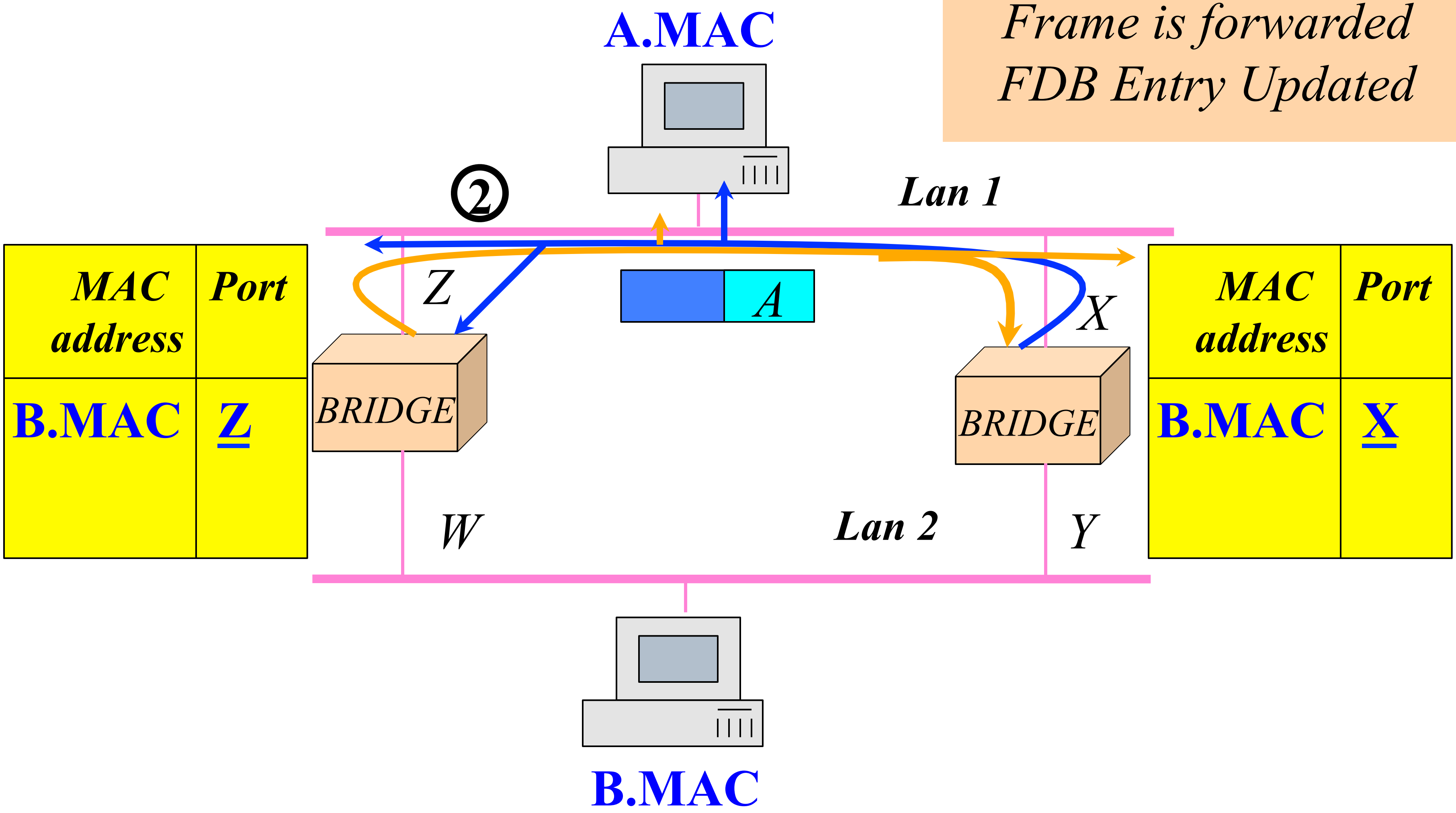
MAC address	***Port***
A.MAC	*1*
B.MAC	*1*
C.MAC	*1*
D.MAC	*2*
E.MAC	*2*
F.MAC	*2*
G.MAC	*2*

Broadcast Storm



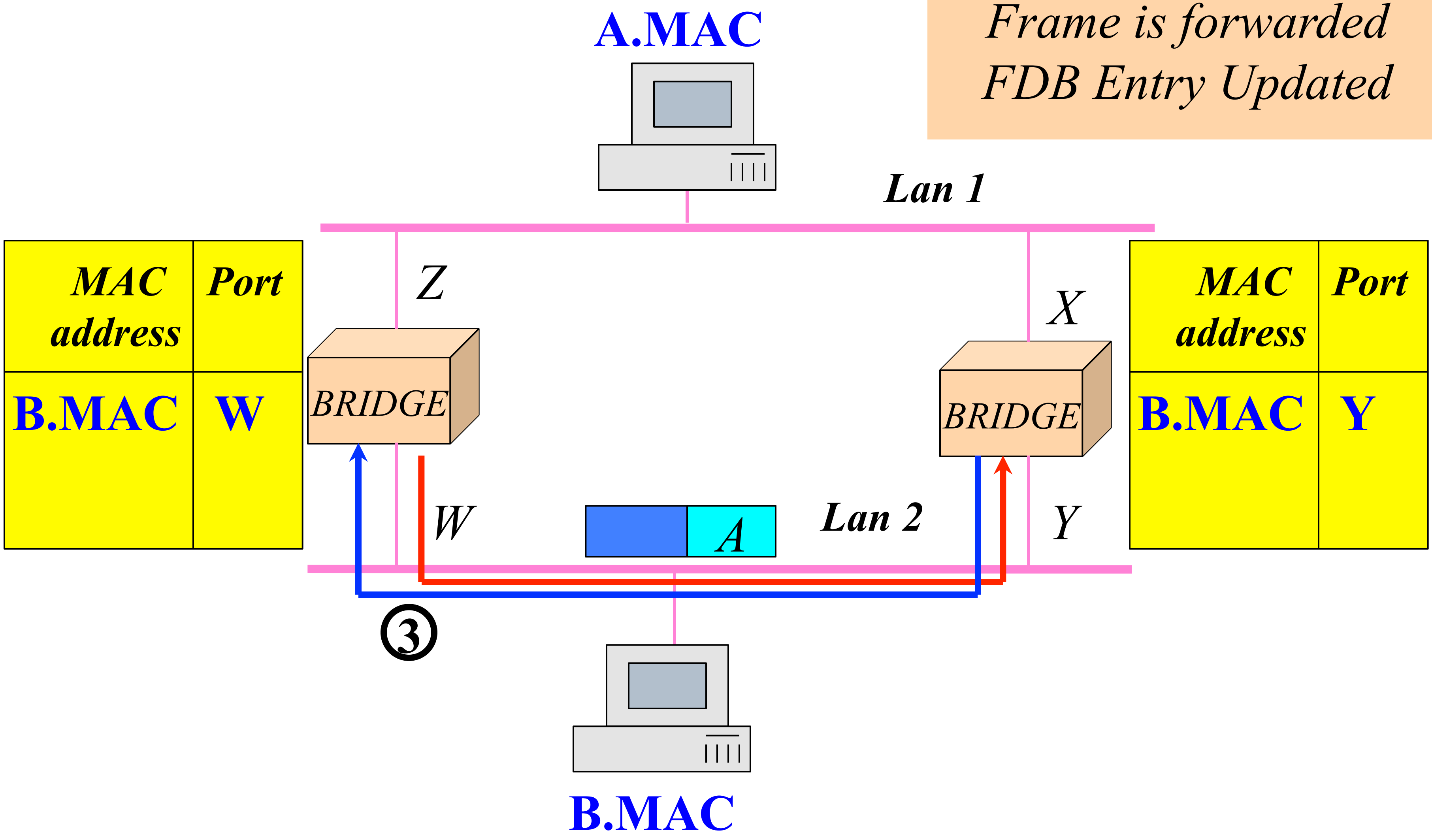
Broadcast Storm

A is not in FDB
Frame is forwarded
FDB Entry Updated



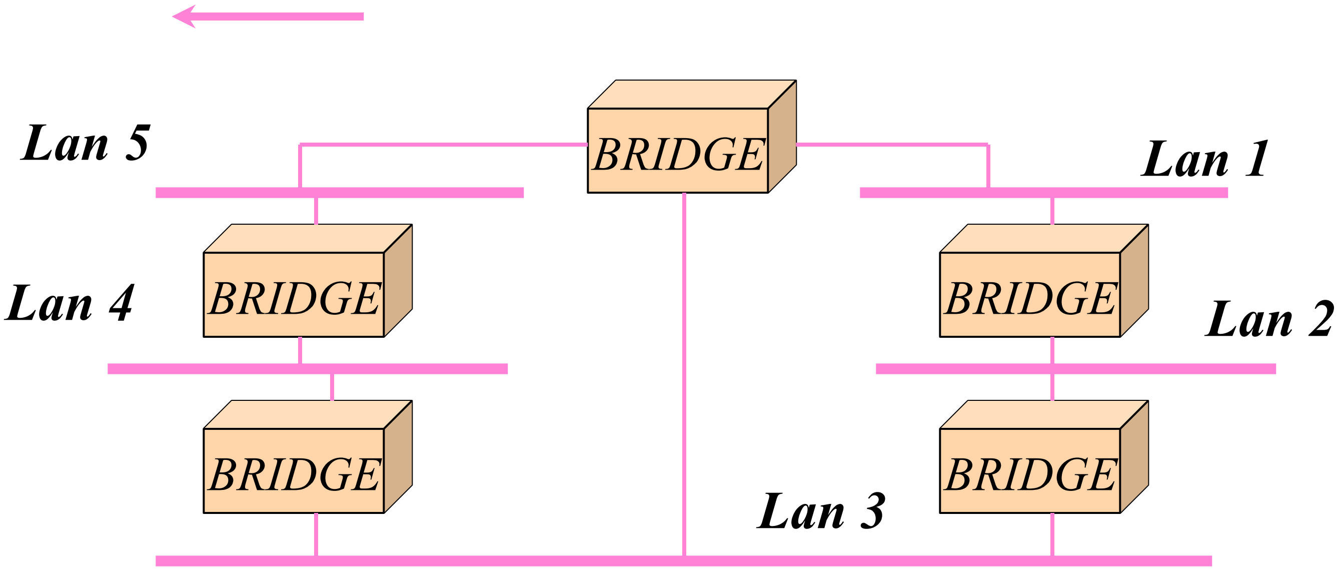
Broadcast Storm

A is not in FDB
Frame is forwarded
FDB Entry Updated



Spanning Tree

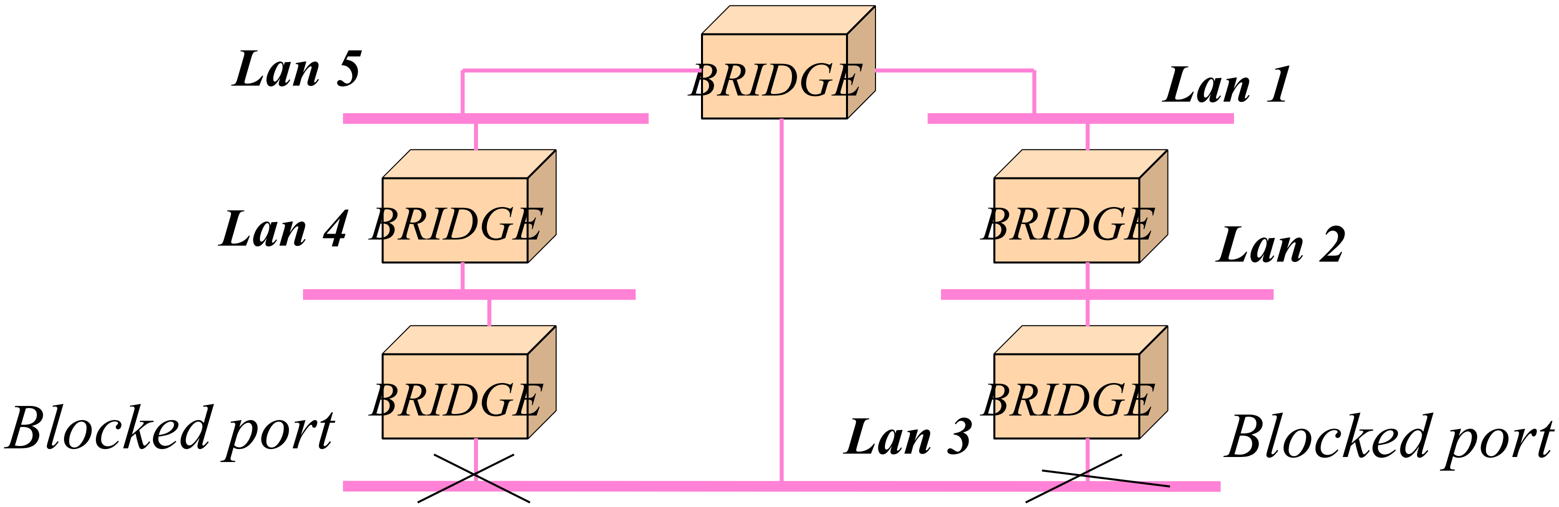
- Problem: LAN topologies are usually meshed for *fault tolerance*



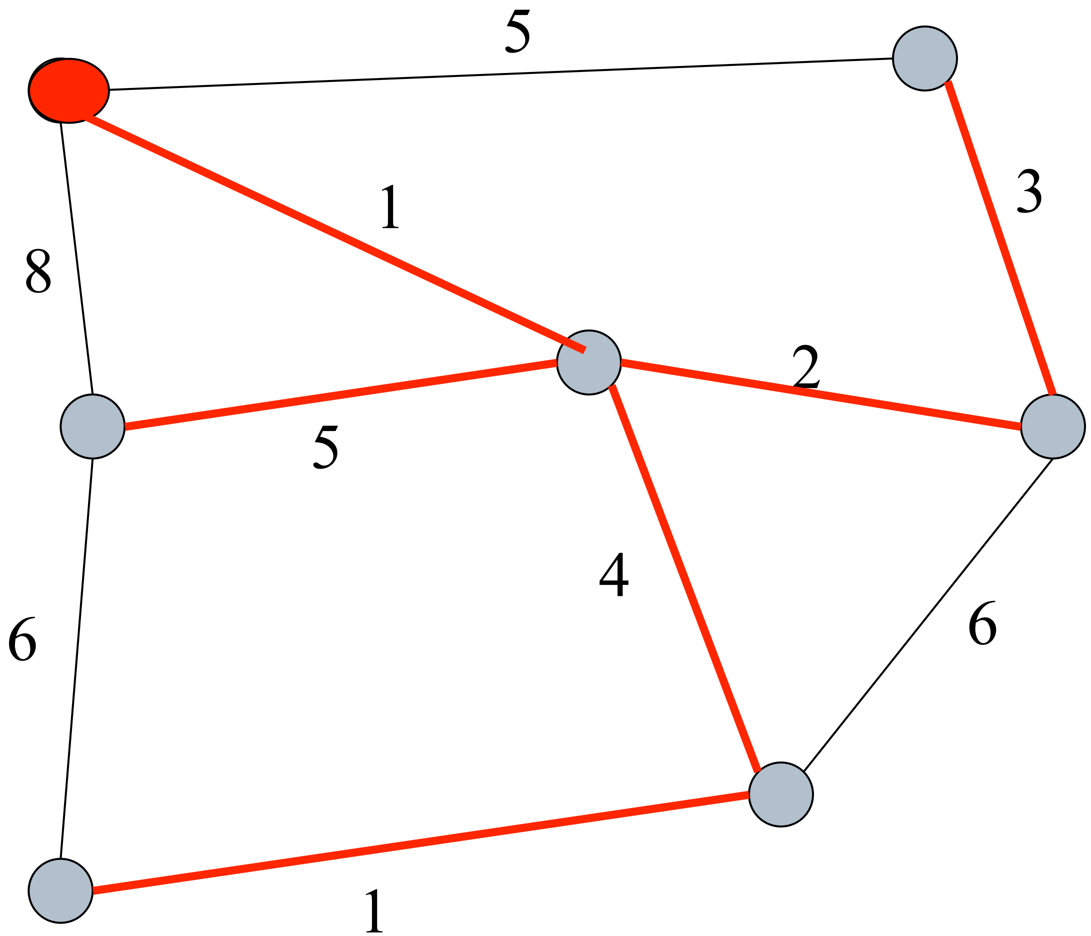
- *Bridging* and *Backward Learning* work on a tree topology
- *Broadcast Storm is due to cycles in the topology graph*

The Spanning Tree Algorithm

- To get a logical tree topology from a physical mesh one.
- The tree topology is obtained blocking some ports
- A blocked port filters data frame and relays control frames (spanning tree)



The Spanning Tree Algorithm

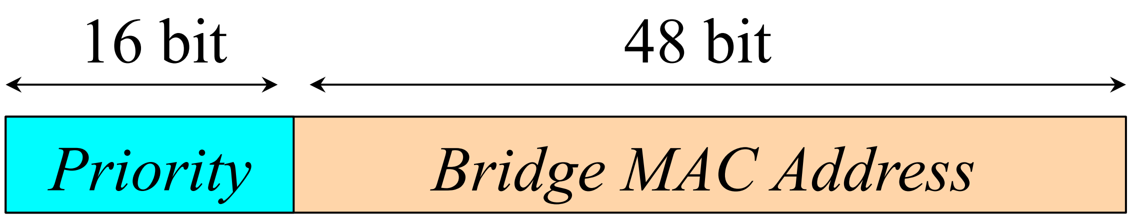


The Spanning Tree Algorithm

- The *Root bridge* is elected
- Each bridge individuates the *root port*
  - the port with the lowest distance to the *root bridge*
- In each LAN a *designated bridge* is chosen.
  - The port interconnecting the *designated bridge* with its LAN is called the *designated port*
- *The root ports* and the *designated ports* are active, the others are put in a *blocked status*. The resulting topology is a spanning tree.

Root Bridge Election

- The first Step is the *Root Bridge* Selection.
- The choice is based on the *Bridge ID (64 bits)*



- The bridge with the lowest Bridge ID is the *Root Bridge*

Root Port Selection

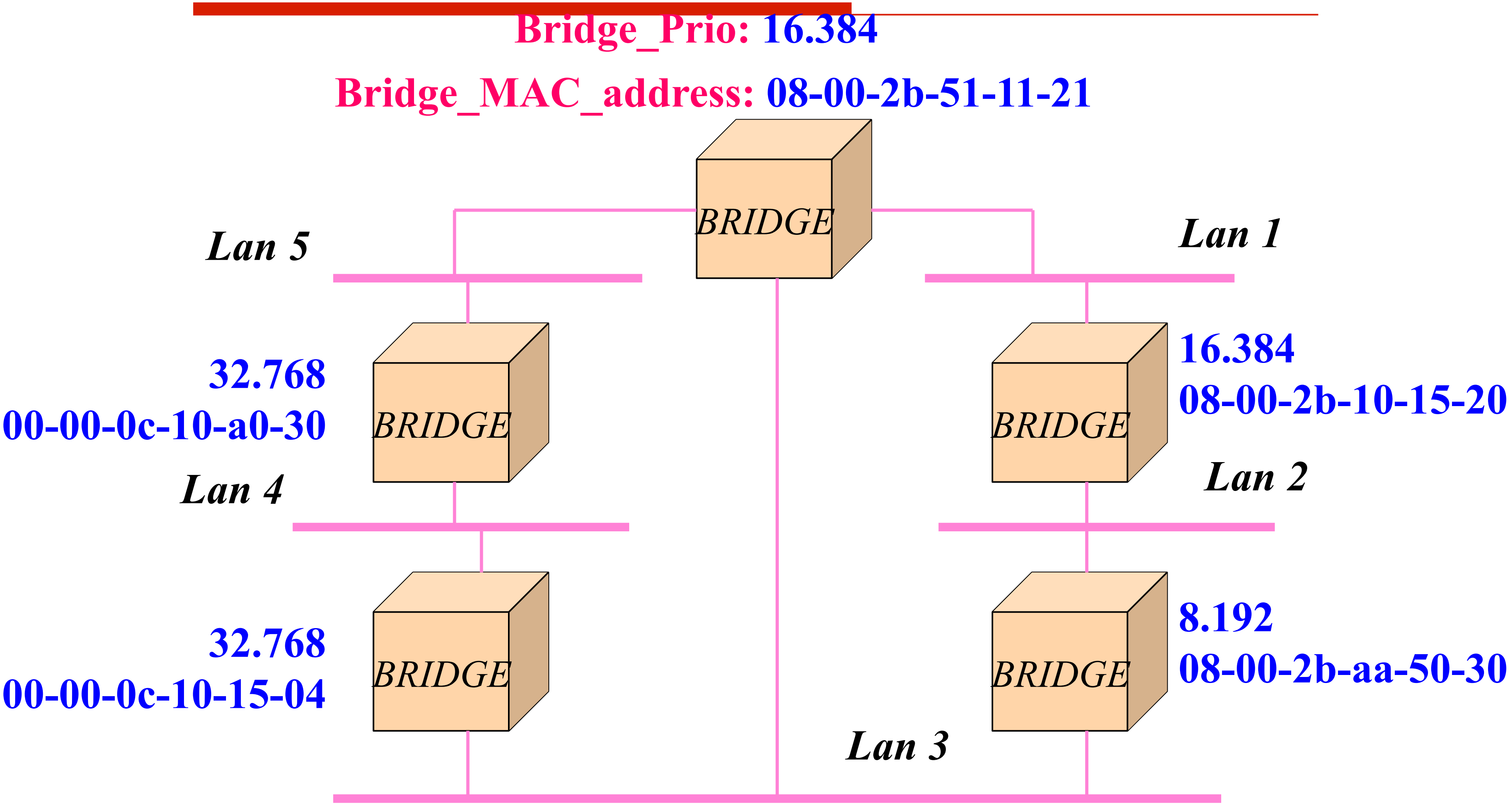
- Once the *Root Bridge* is elected, each *Bridge* selects the *Root Port*
  - lowest distance to the *Root Bridge*
- The distance is expressed as a cost through the *Root Path Cost* parameter (it often corresponds to the *number of hops*)

Designated Bridge Port Selection

- A Designated Bridge for each LAN segment is selected. It forwards the frames towards the *root Bridge*
  - The bridge with minimum distance towards the *Root Bridge* becomes the Designated Bridge (if equivalent, lowest Bridge ID criteria)
- The Designated Bridge is connected to the LAN segment through the Designated Bridge Port

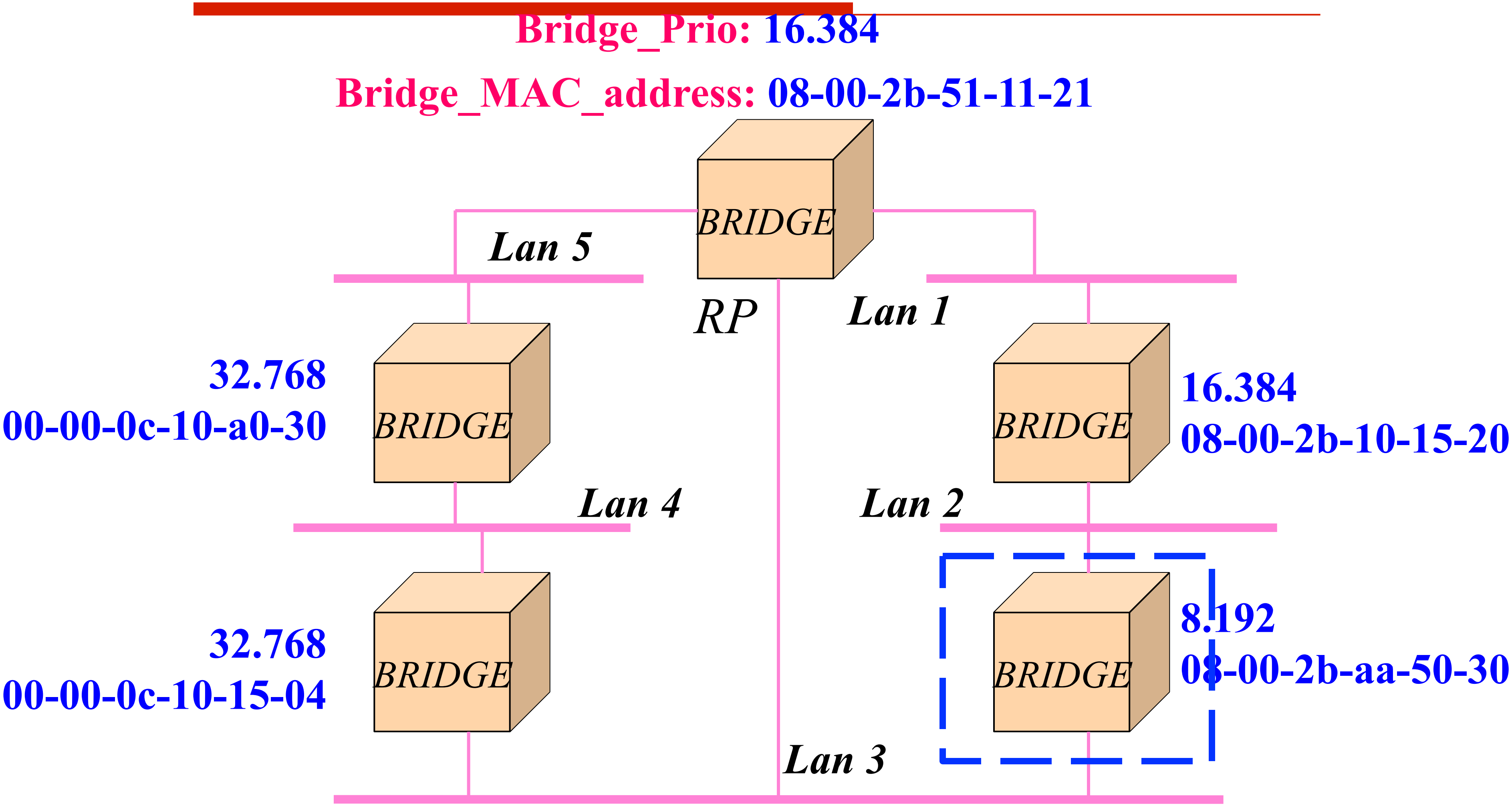
All the ports of a Root Bridge are Designated Bridge Ports !

Spanning Tree: An Example



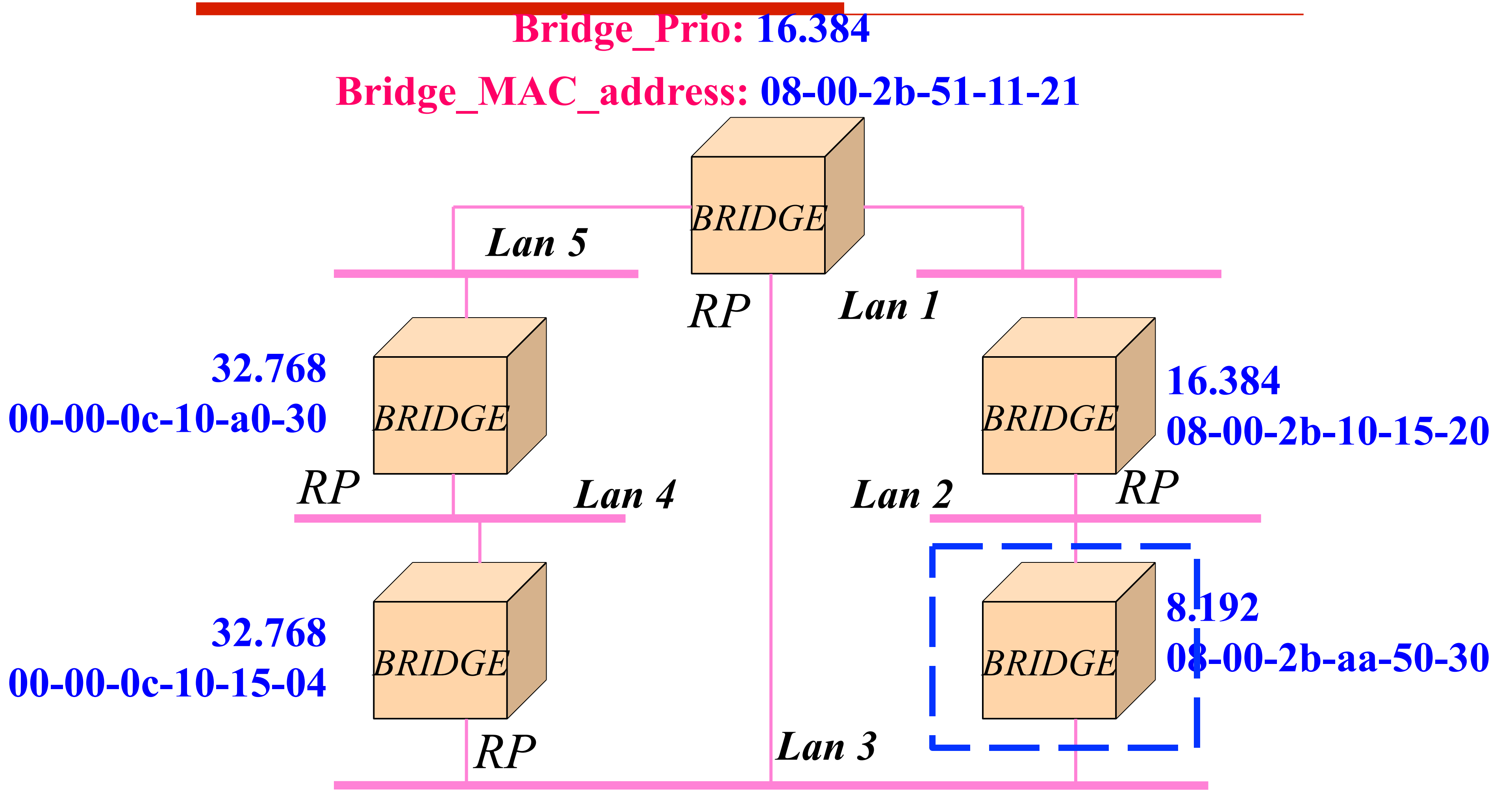
Before the ST algorithm

Spanning Tree: An Example



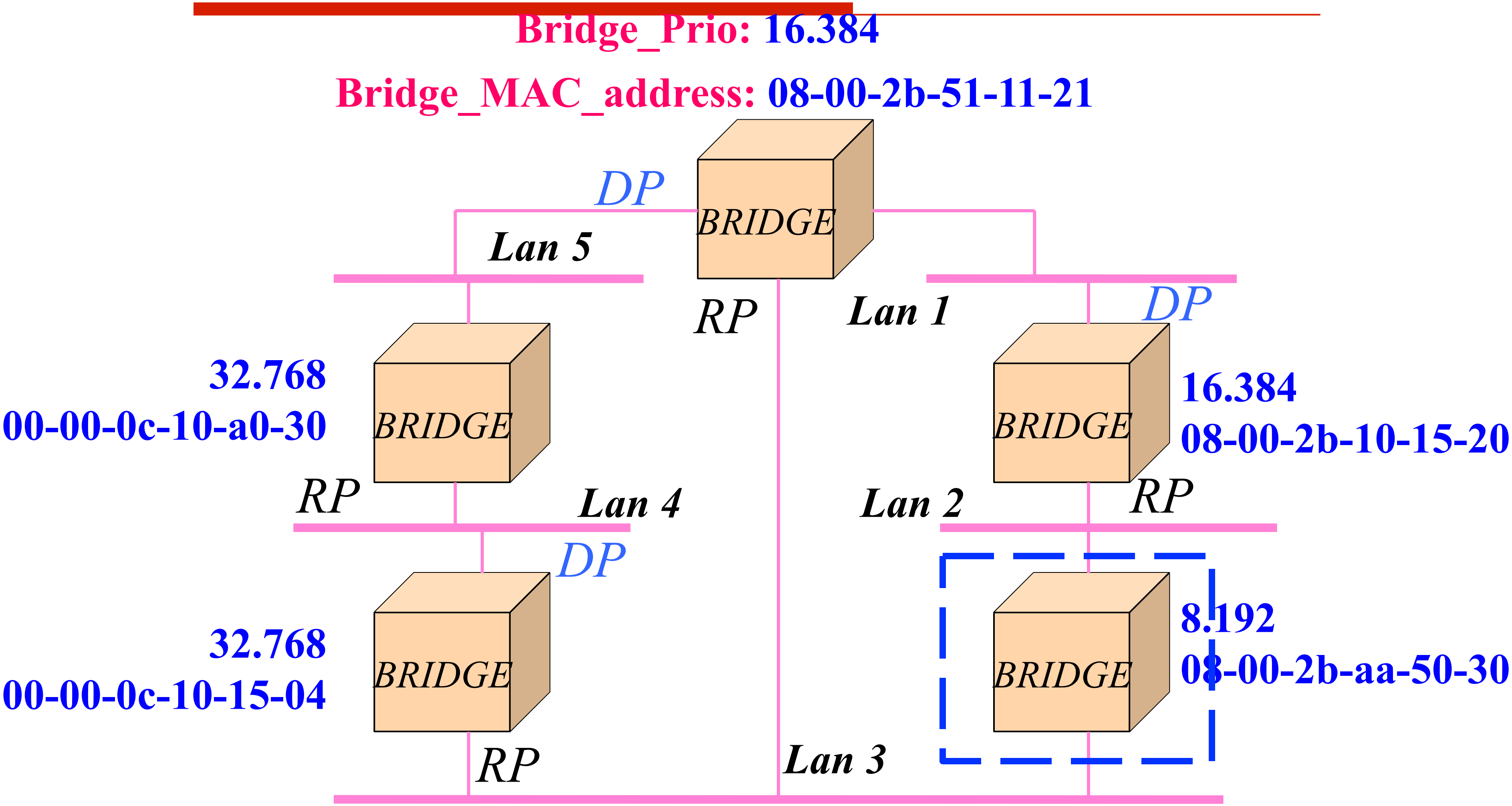
After the ST algorithm

Spanning Tree: An Example



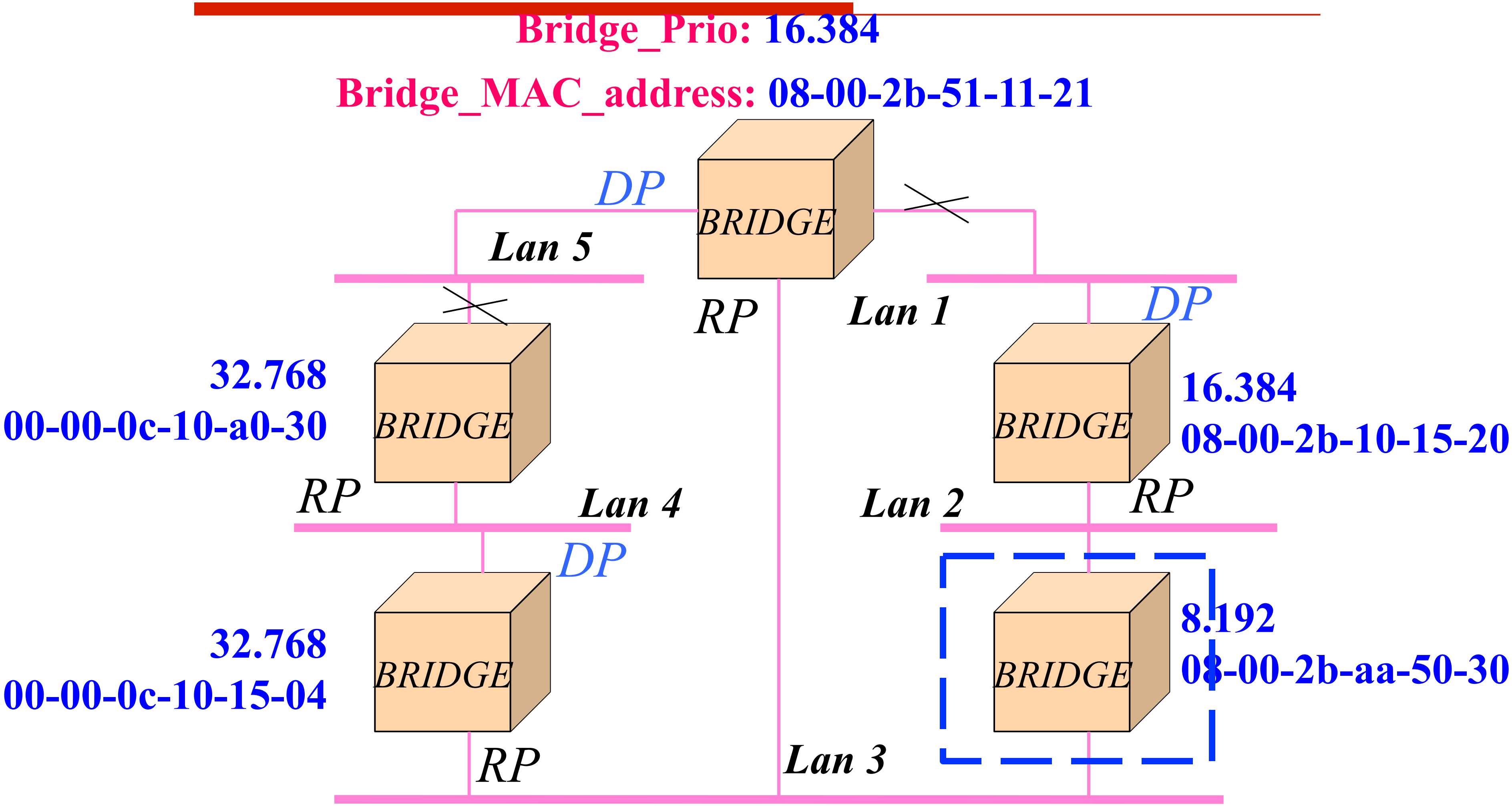
After the ST algorithm

Spanning Tree: An Example



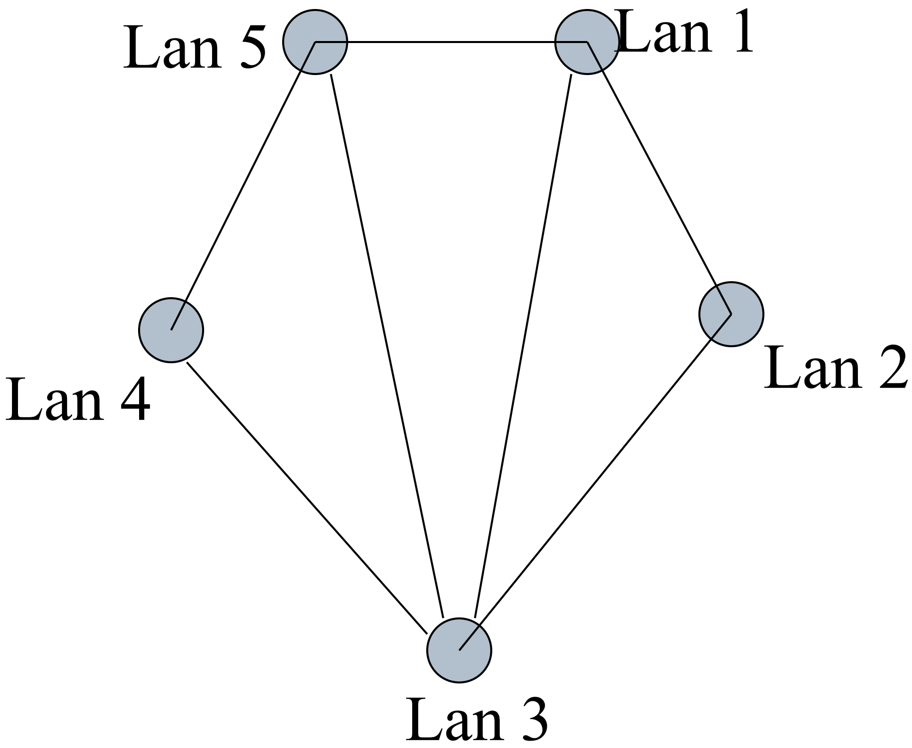
After the ST algorithm

Spanning Tree: An Example

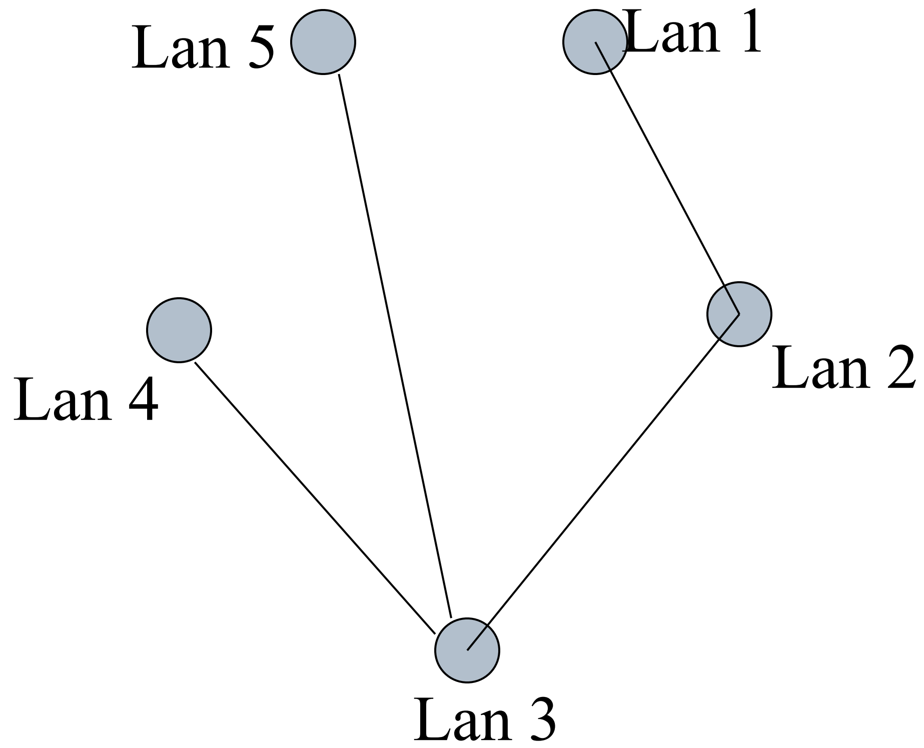


After the ST algorithm

Logical Bridging Graphs



Physical Topology



Logical Topology