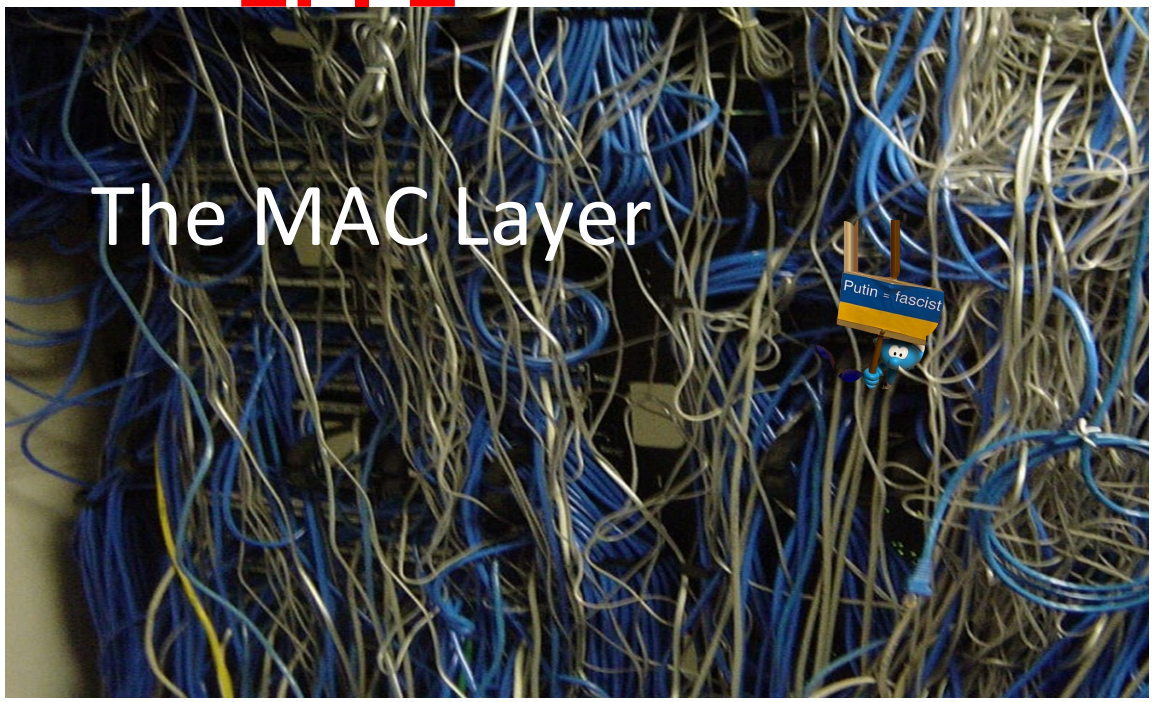
EPFL

The MAC Layer



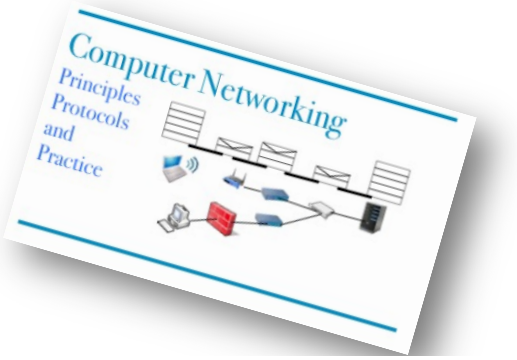
Jean-Yves Le Boudec 2022

Contents

1. MAC as Shared Medium
2. Bridges
3. Format and addresses
4. Virtual LANs
5. WiFi Distribution Systems
6. Security aspects

Textbook

Sections 2.6, 3.16 and 4.12

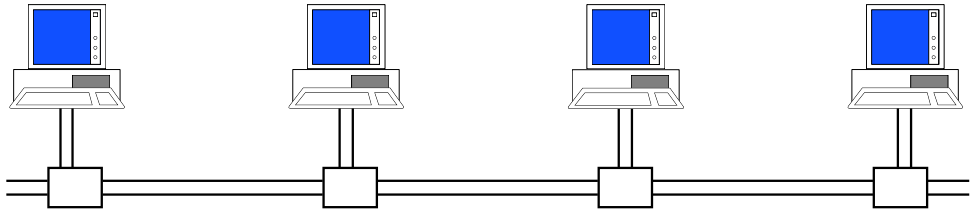


1: Shared Medium Access
MAC = medium access control

Why did engineers invent the MAC layer ?

Share a cable (Ethernet, Token Ring, Powerline)

Use a wireless radio link (GSM, WiFi, Bluetooth)



What is the problem ?

If several systems talk together, how can we decode ?

Solution 1: joint decoding (decode everyone) -- used in cellular networks

Solution 2: mutual exclusion protocol (only one system can talk at a time) + distributed (no central component)

How does MAC layer work?

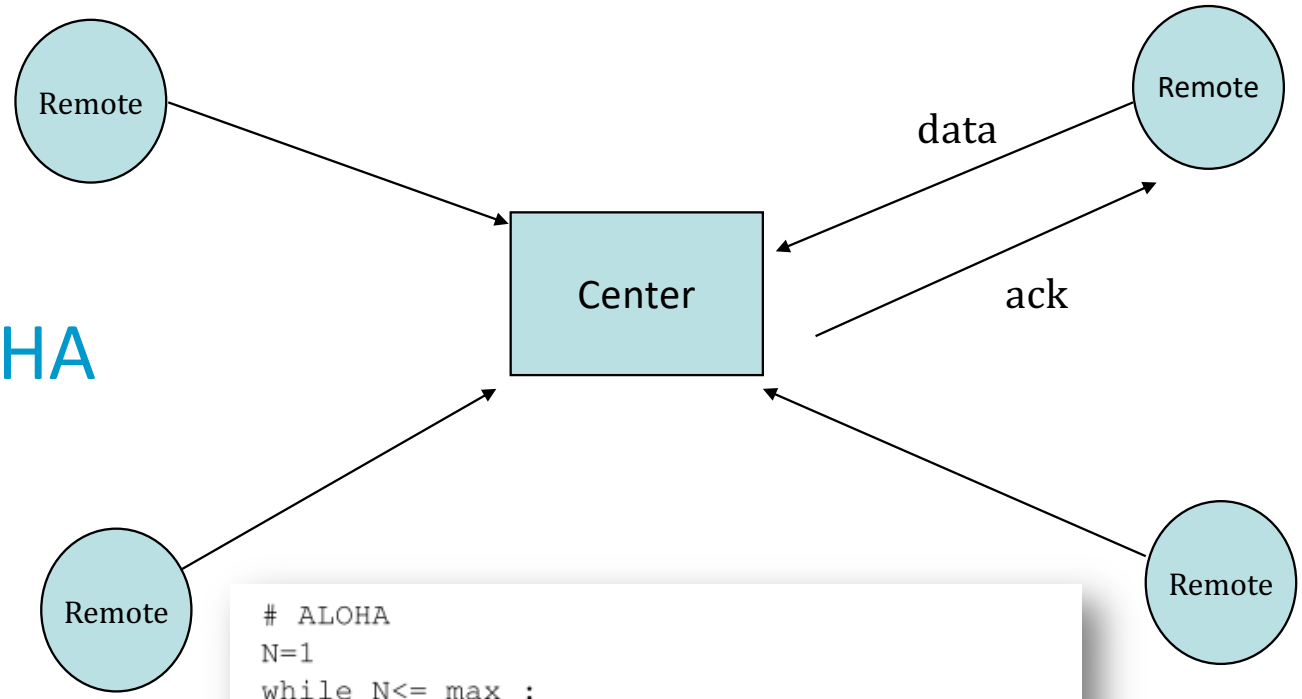
There are several types of MAC layers

- Deterministic: CAN bus, Time Division Multiple Access, Frequency Division Multiple Access (cellular networks)
  use a static schedule in time or frequency

- Deterministic: Token Ring; FDDI:
  every host takes turn and passes a token to next host

- Random Access: WiFi, historical Ethernet, Zigbee, signalling channel on cellular networks, Powerline communication
  Based on Aloha and CSMA (explained next)

ALOHA




```
# ALOHA
N=1
while N<= max :
   send(frame)
   wait(ack_on_return_channel or timeout)
   if (ack_on_return_channel):
       break  # transmission was successful
   else:
       # timeout
       wait(random_time)
       N=N+1
 else:
   # Too many transmission attempts
```

Transmission procedure at remote

Aloha is the basis of all non-deterministic access methods. The Aloha protocol was originally developed for communications between islands (University of Hawaï) that use radio channels at low bit rates.

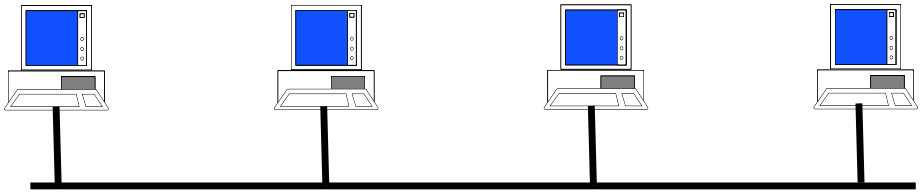
It assumes one shared channel from remote to center, and a separate (non interfering) channel from center to remotes

Collisions occur when two packet transmissions overlap, and if a packet is lost, then source has to retransmit; the retransmission strategy is not specified here; many possibilities exist. We will see the one used for CSMA/CD.

There is no feedback to the source in case of collision (was too complex to implement at that time). The picture shows a radio transmission scenario; Aloha can also be used on a cable (bus). It is used nowadays in cases where simplicity is more important than performance (for example: machine bus)

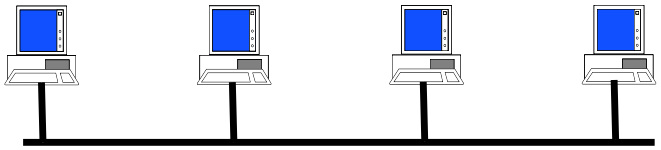
The maximum utilization can be proven to be 18%, which is small. This is assuming an ideal retransmission policy that avoids unnecessary repetitions of collisions.

CSMA (carrier sense multiple access) improves on Aloha by requiring that stations listen before transmitting



```
# Non persistent CSMA
N=1
while N<= max :
   listen(channel)
   if free(channel):
      send(frame)
      wait(ack or timeout)
      if received(ack) :
         break  # transmission was successful
      else :
         # timeout
         N=N+1
   else:
      wait(random_time)
 # end of while loop
   # Too many transmission attempts
```

assumes a single *transitive* channel (everyone can hear everyone)



CSMA collisions…

A. Cannot occur because the medium is transitive

B. Can occur if the bandwidth-delay product is smaller than 1

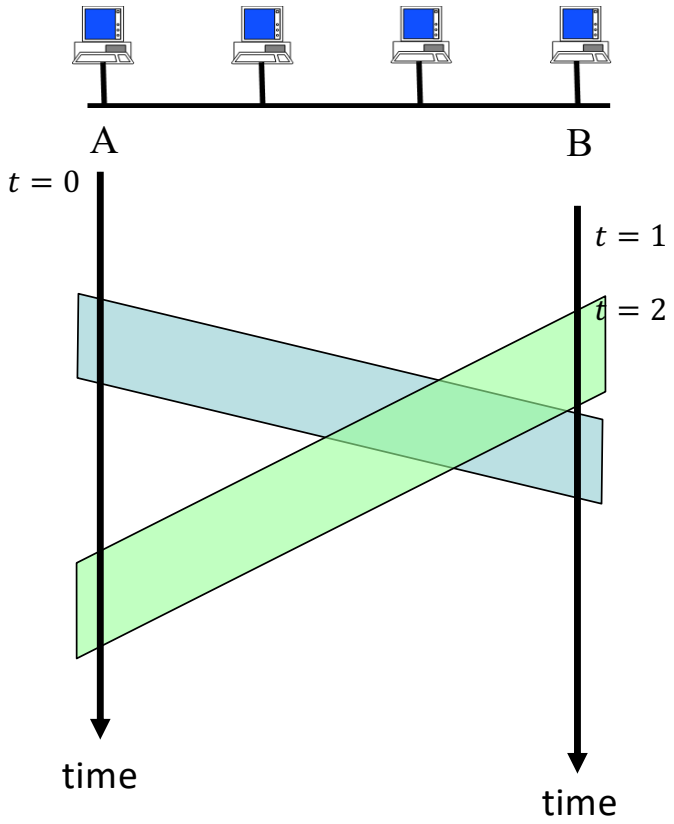
C. Can occur regardless of the bandwidth delay product

D. I don’t know

Solution

Many collisions can be avoided, but not all.

This is because of propagation delays. If two or more stations may sense that the medium (= the channel) is free and start transmitting at time instants that are close enough for a collision to occur. Assume propagation time between A and B is 2 ms and that all stations are silent until time 0. At time 0, station A starts transmitting for 4 ms, at time 1 ms, station B has not received any signal from A yet, so it can start transmitting. At time 2ms, station B senses the collision but it is too late according to the protocol.



CSMA avoids some collisions, but not all

The effect of the CSMA protocol can be expressed in the following way. Call *T* the maximum propagation time from station *A* to any other stations; if no collision occurs during a time interval of duration *T* after *A* started transmitting, then *A* has seized the channel (no other station can send).

CSMA works well only if the transmission time is much larger than propagation, namely bandwidth-delay product << frame size.

In order to avoid repeated collisions, it is required to wait for a random delay before re-transmitting. If all stations choose the random delays independently, and if the value of the delay has good chances of being larger than *T*, then there is a high probability that only one of the retransmitting stations seizes the channel.

CSMA / CD (Collision Detection) detects collisions as they occur

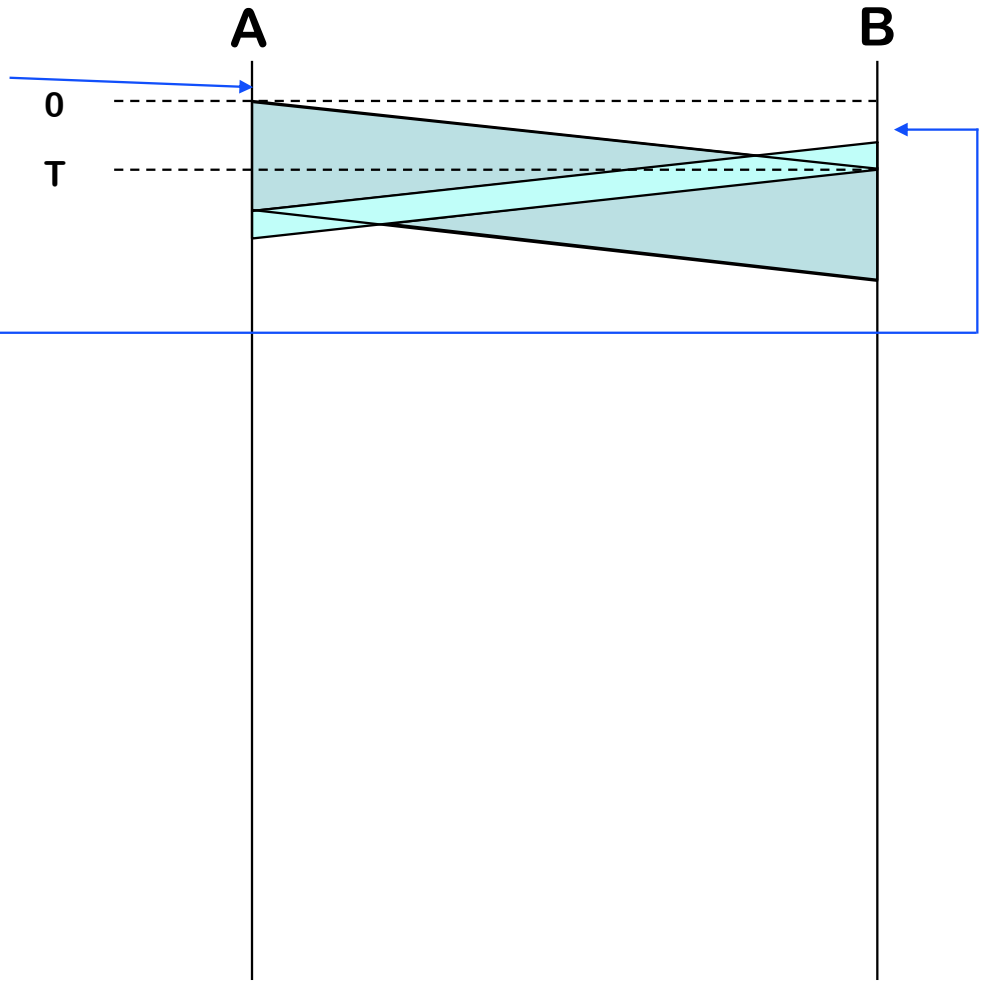
Ack replaced by CD

```
# CSMA/CD pseudo-code
N=1
while N<= max :
   wait(channel_becomes_free)
   send(frame)
   wait_until (end_of_frame) or (collision)
   if collision detected:
        stop transmitting
        send(jamming)
        k = min (10, N)
        r = random(0, 2^k  - 1) * slotTime
        wait(r*slotTime)
        N=N+1
   else :
        wait(inter-frame_delay)
        break
 # end of while loop
   # Too many transmission attempts
```

CSMA / CD Time Diagram 1

A senses idle channel, starts transmitting

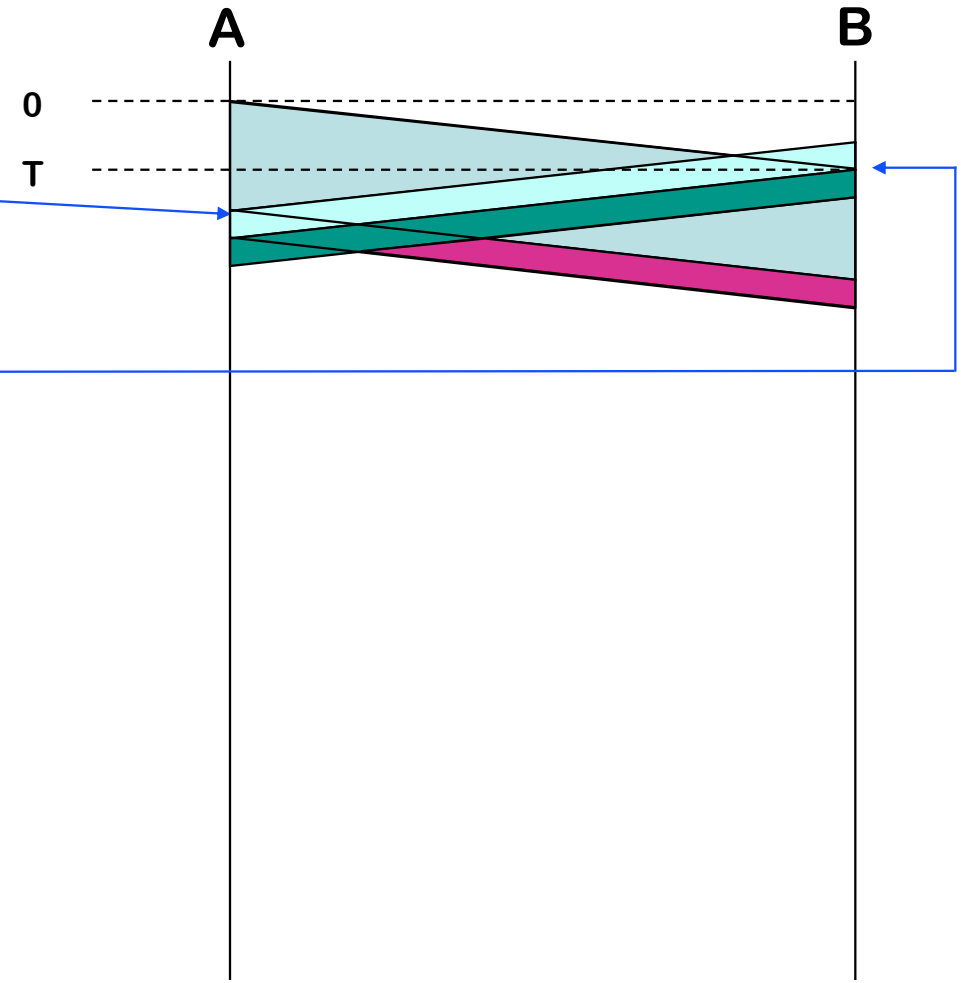
shortly before *T*, B senses idle channel, starts transmitting



CSMA / CD Time Diagram 2

A senses collision, continues to transmit 32 bits (“jam”)

B senses collision, continues to transmit 32 bits (”jam“)



CSMA / CD Time Diagram 3

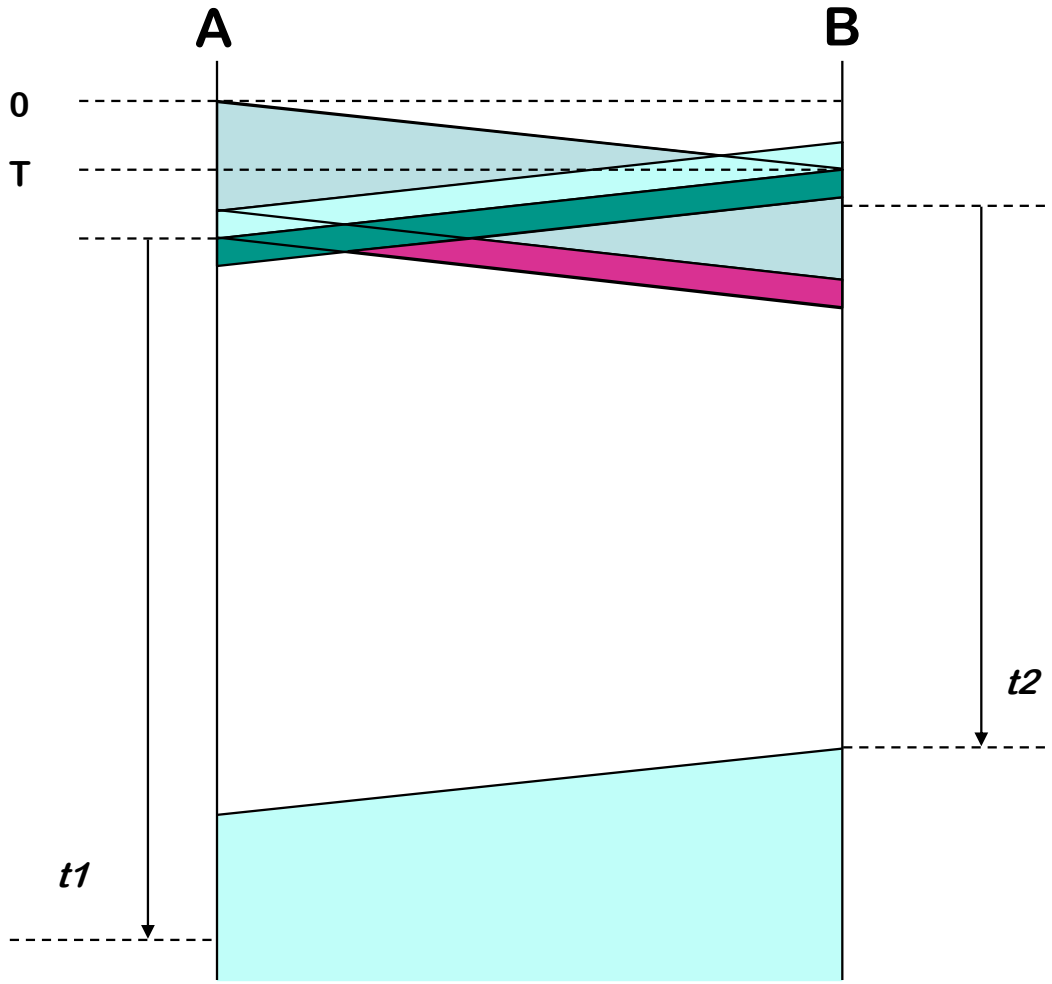
A waits random time *t1*

B waits random time *t2*

B senses channel idle and transmits

A senses channel busy and *defers* to B

A now waits until channel is idle



CSMA/CD improves on CSMA by requiring that stations detect collisions and stop transmitting (after 32 bits, called *jam* bits, in order to ensure that all circuits properly recognize the presence of collisions).

CSMA/CD has a better performance than Aloha or CSMA

After a collision is detected, stations will re-attempt to transmit after a random time.

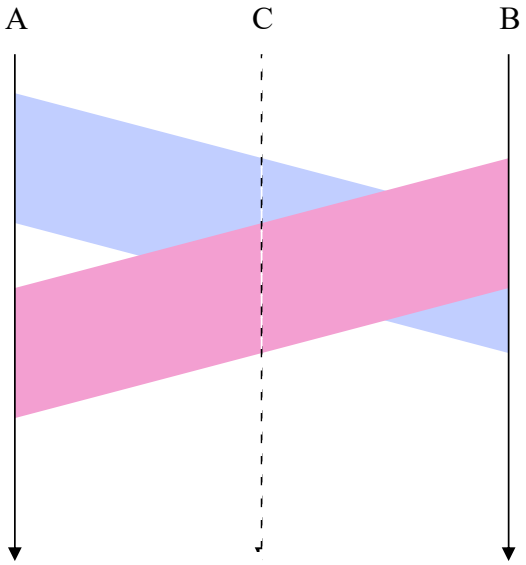
Acknowledgements are not necessary because absence of collision means that the frame could be transmitted (see “Minimum Frame Size“).

The interframe delay (“gap”) is 9.6 µs. It is used to avoid blind times, during which adapters are filtering typical noise at transmission ends.

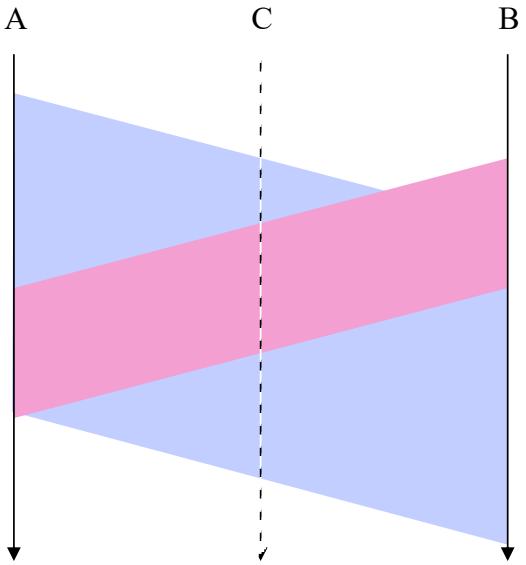
The random time before retransmission (exponential backoff) is chosen in such a way that if repeated collisions occur, then this time increases exponentially. The effect is that in case of congestion (too many collisions) the access to the channel is slowed down.

A Minimum Frame Size is Necessary to Guarantee Collision Detection

Frame sent by A is too short: collision is ***not*** visible at A but is visible at C

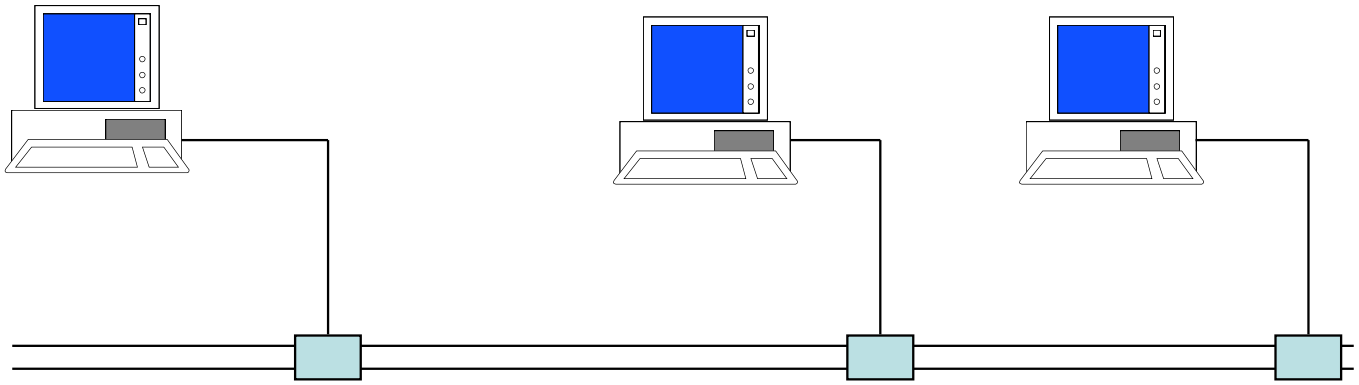


Frame sent by A is large enough collision ***is*** visible at A



CSMA/CD imposes a minimum frame size = 64 B and a maximum network diameter : 2 km for 10Mb/s, 200m for 100 Mb/s

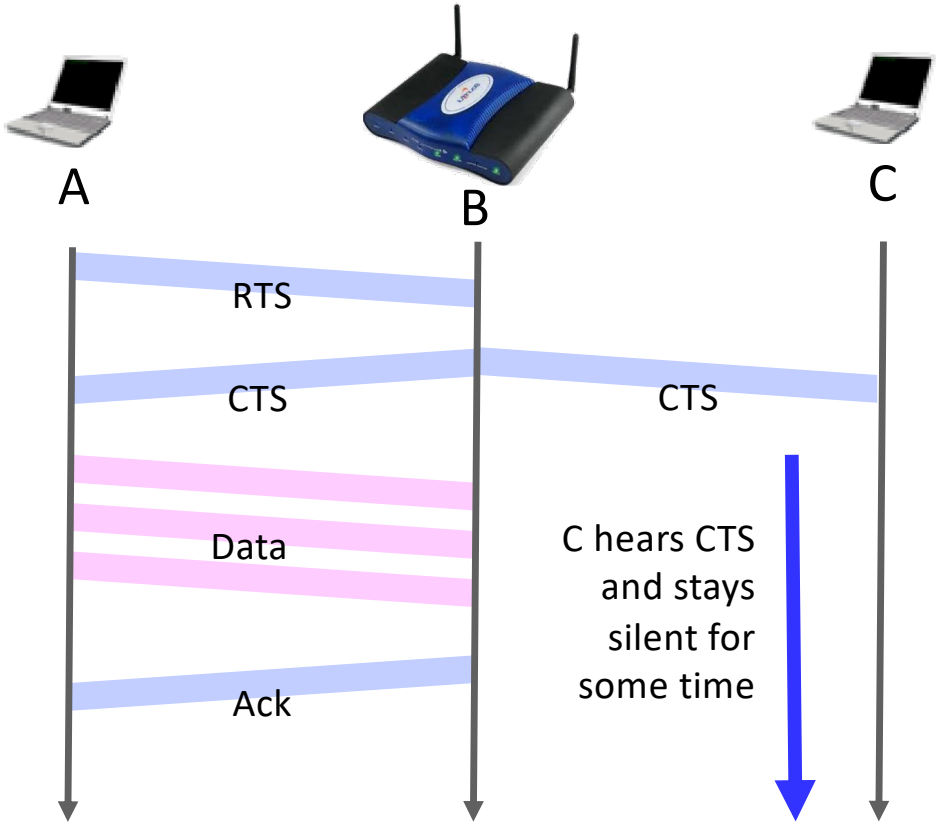
CSMA/CD = Historical Ethernet (= on Coax Cables)



This is (old style) Ethernet. It uses CSMA/CD.
Also called “half-duplex Ethernet”

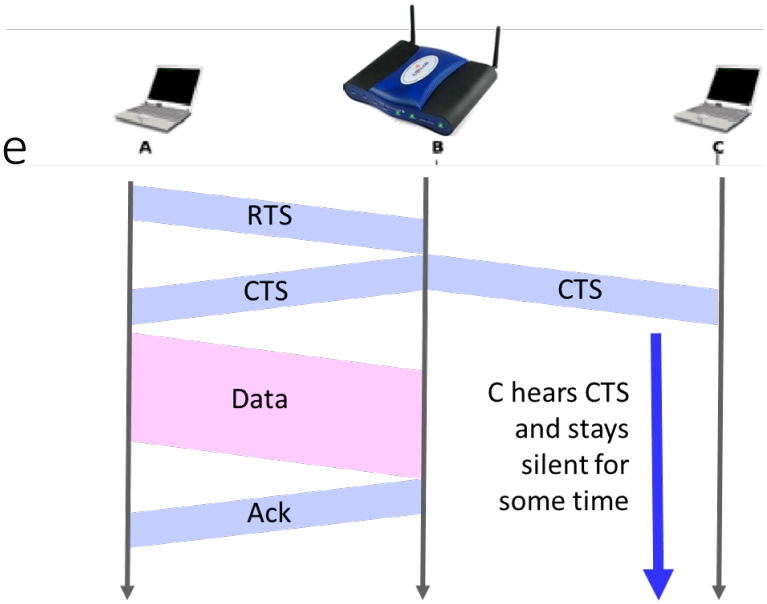
WiFi = CSMA/CA (Collision Avoidance) = variant of CSMA/CD

- Acks are used to detect collisions (by absence)
- Carrier sensing does not always work (C is hidden from A); can be mitigated with RTS/CTS
- Many other optimizations to reduce collisions (more conservative random backoffs, “network allocation vector” can be used to avoid collision during Ack,e tc.)



In a WiFi network with a single channel and a single base station, how many data frames can be transmitted concurrently ?

A. At most one in total
B. One from mobile to base station and one from base station to mobile
C. At most one per mobile to base station and one from base station
D. At most one per mobile to base station and one per mobile from base station
E. I don't know

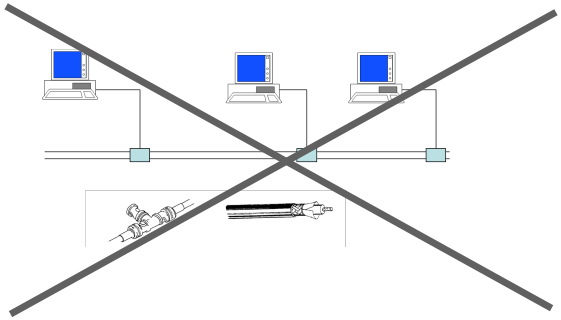


Solution

Answer A
Here the MAC layer is controlling access to the radio channel between all senders. It makes sure that only one can send at a time

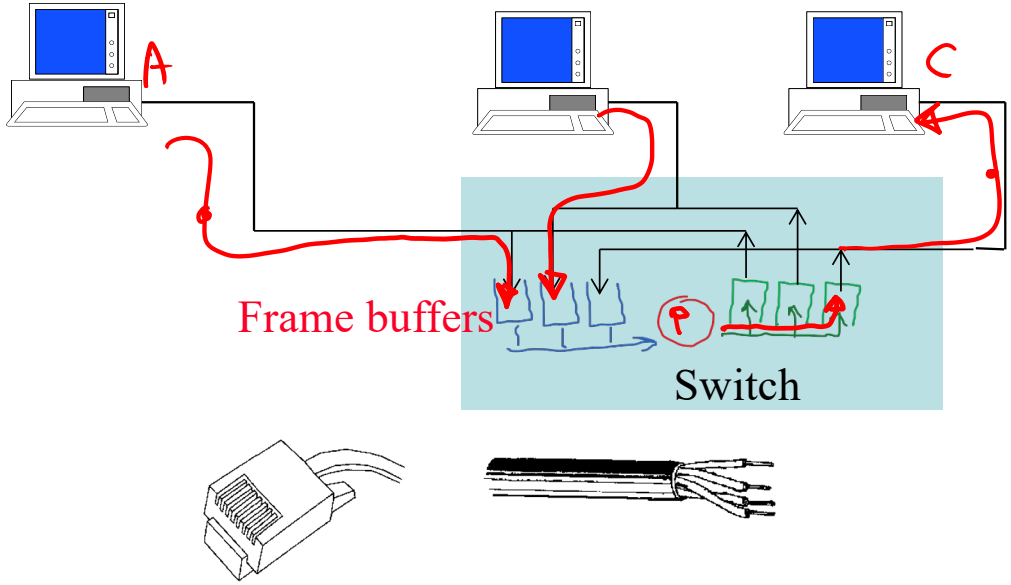
2. Ethernet LANs Use Switches

Ethernet was historically shared medium
But today is based on switches = bridges
a bridge is an intermediate system of the MAC layer =
it forwards packet based on MAC addresses.



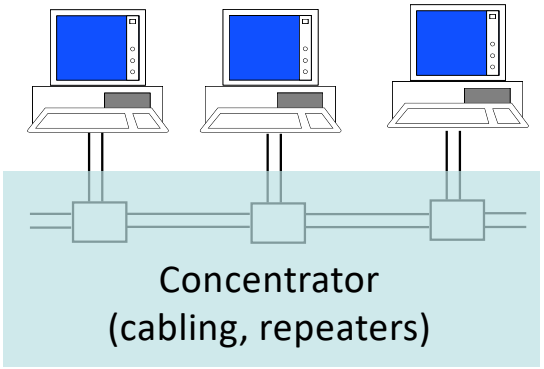
A bridge is a queuing
system

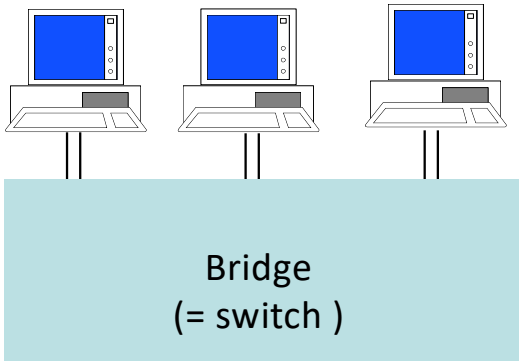
Here the links are full-duplex
Ethernet: there is no
MAC protocol (no
CSMA/CD) and packets can flow
simultaneously in both directions.



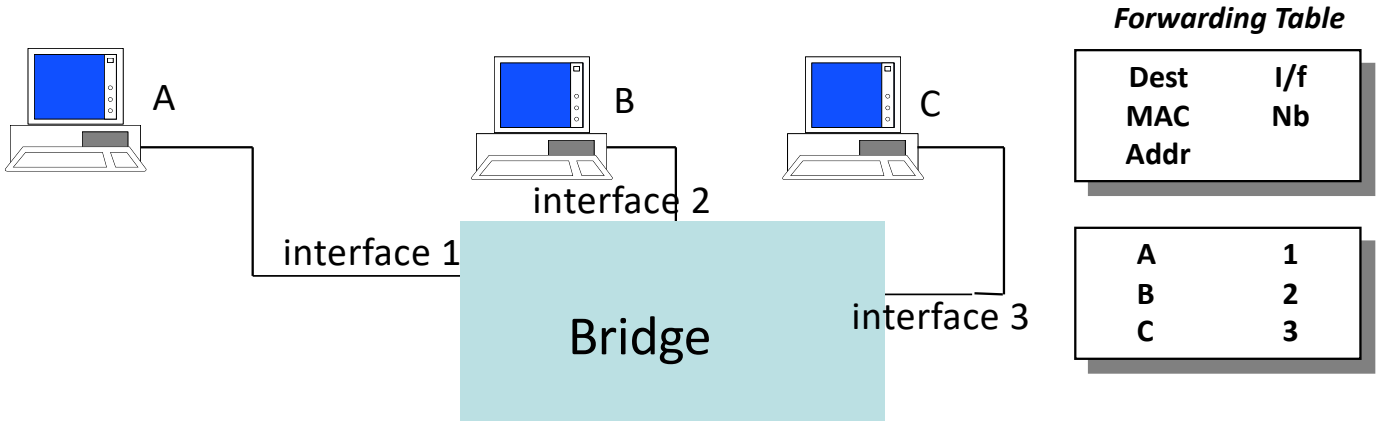
Bridges are Transparent

The design of bridges requires them to be transparent, i.e. there should be no difference for the protocol used by end-systems when they are connected to a concentrator (cabling only) or a bridge.





Bridges use a forwarding table with *exact match*



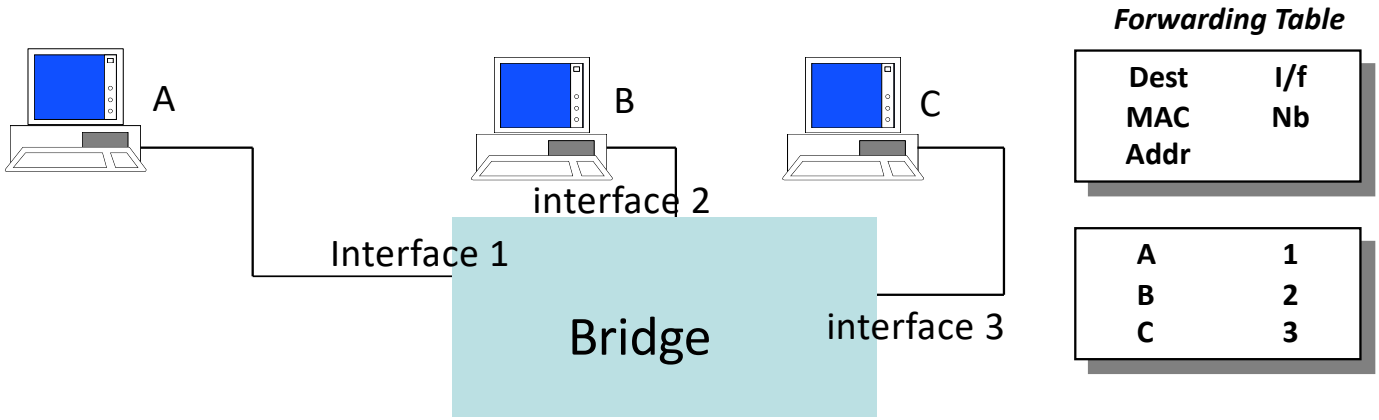
Forwarding table contains list of destination MAC addresses and interfaces (also called “ports”).

Bridge algorithm

listen to all traffic on all interfaces
read destination MAC address and obtain destination interface from table if it exists
if destination interface is same as origin, discard frame;
else send to destination interface.

If destination address not in table, broadcast to all interfaces

Bridges learn addresses by observing traffic



How can a bridge build its forwarding table ?

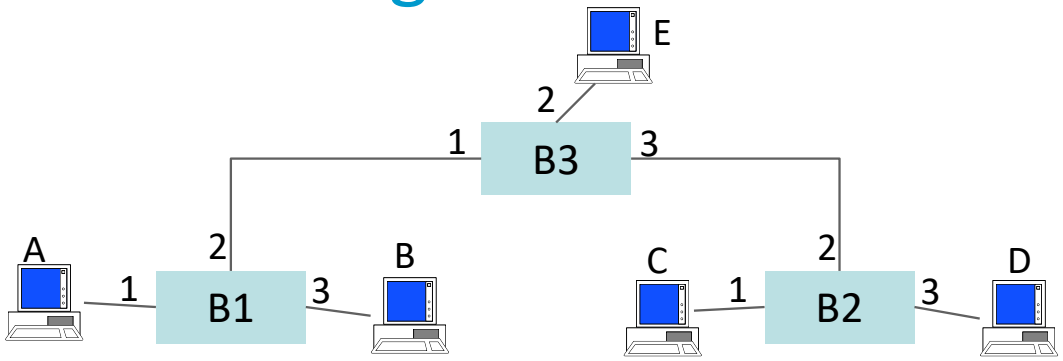
table built by learning from Source Address field in MAC frame

learnt addresses time out if not re-learnt

If destination address not in table, frame is broadcast to all interfaces

Multicast addresses are handled separately – see later.

The method of learning can be extended to networks of bridges



Forwarding Table at B3

Dest MAC Addr	I/f Nb
A	1
B	1
C	3
D	3
E	2

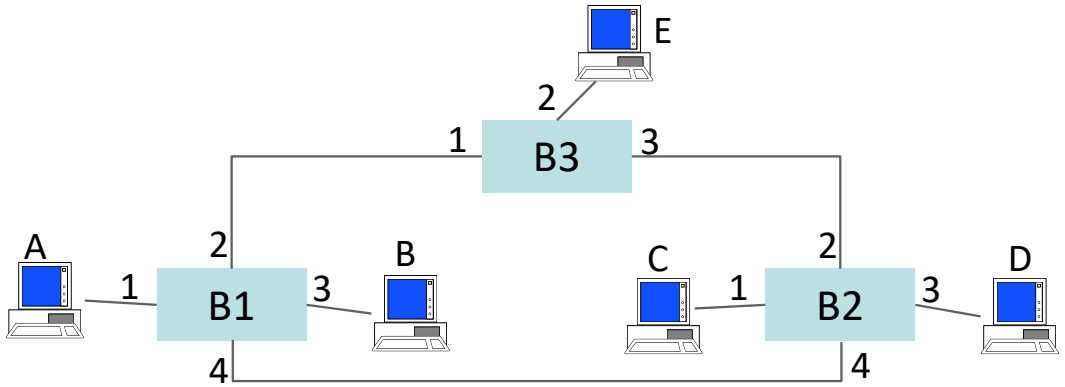
Does this always work ?

A. Yes because learning propagates
B. Yes because MAC addresses are unstructured
C. Only in some special cases
D. No, never
E. I don’t know

Solution

Answer C

It does not work if there are multiple paths in the topology, i.e. if there are loops.



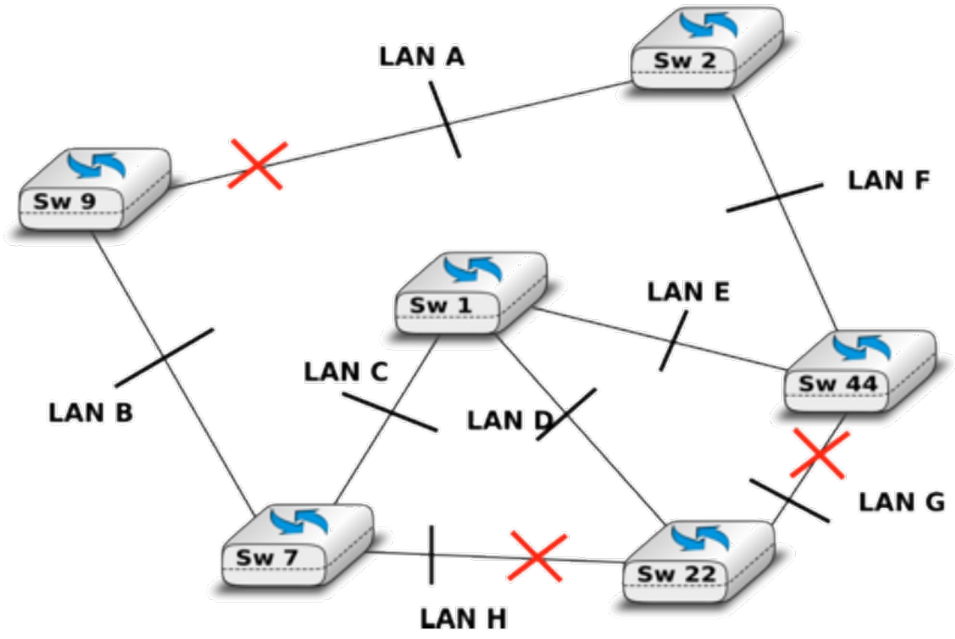
For example: here B3 does not know whether A is on port 1 or 3, as flooded packets may arrive from either side.

Also, flooding may create packets that loop indefinitely

The Spanning Tree Protocol Forces the Active Topology to be Loop-free, i.e. to be a Tree

Bridges run a protocol between neighboring bridges, called the Spanning Tree Protocol

What: de-activate some ports such that the remaining topology is a tree that spans all bridges
+ adapt to failures and additions



The Spanning Tree Protocol: How

Algorhyme

I think that I shall never see
a graph more lovely than a tree.
A tree whose crucial property
is loop-free connectivity.
A tree that must be sure to span
so packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced. → using the Bellman-Ford algorithm, see "Distance Vector"
In the tree, these paths are placed.
A mesh is made by folks like me,
then bridges find a spanning tree.

Radia Perlman
(inventor of the Spanning Tree Protocol)

https://youtu.be/ERFohfN_H40

Specification of the Spanning Tree Protocol

1. Bridges elect one root bridge
   = bridge with smaller (configurable) bridge label
2. Only links that go to root bridge along the shortest path are active
   Spanning Tree = set of shortest paths to root
3. All bridges monitor that the root is reachable and if not, trigger re-computation of a new spanning tree

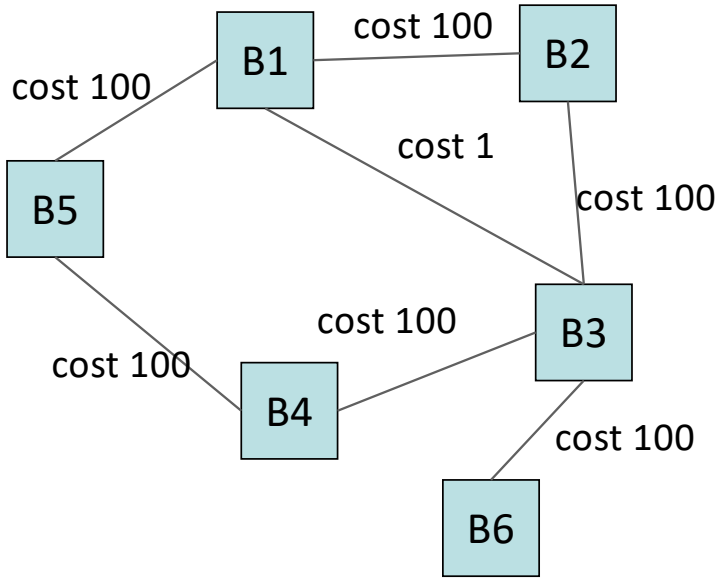
Each LAN between bridges has a (configurable) cost,
by default, decreasing function of bit rate.

Distributed (no central control, all of this is done by bridges themselves) using a variant of the Bellman-Ford algorithm.

Details are a bit more complicated when links between bridges are shared medium and not point to point: see textbook.

Which of the following links are on the spanning tree ?

A. B4B5
B. B4B3
C. Both
D. None
E. I don't know



Solution

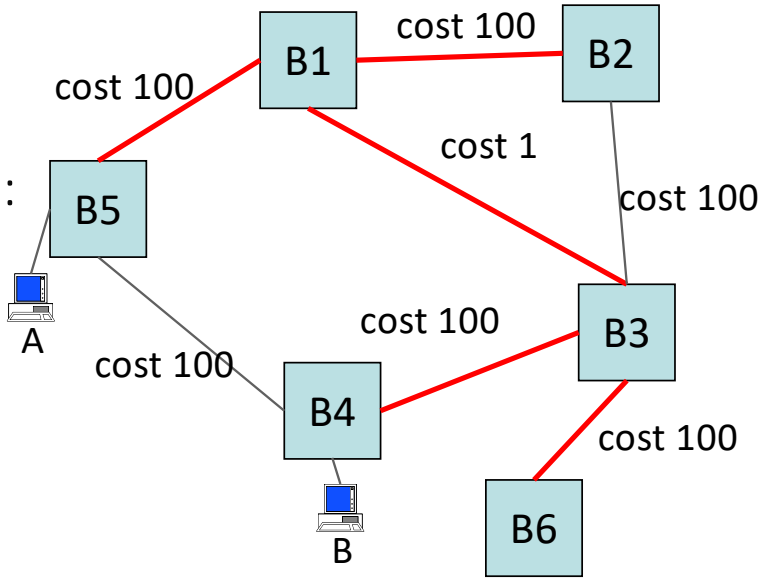
Answer B

The root is B1 (smallest label)

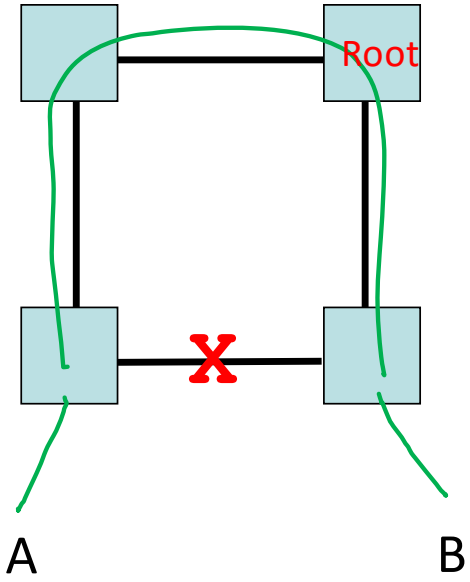
The shortest path tree is shown →

The path for packets from A to B is:

B5-B1-B3-B4



What does STP produce on this network of bridges ?



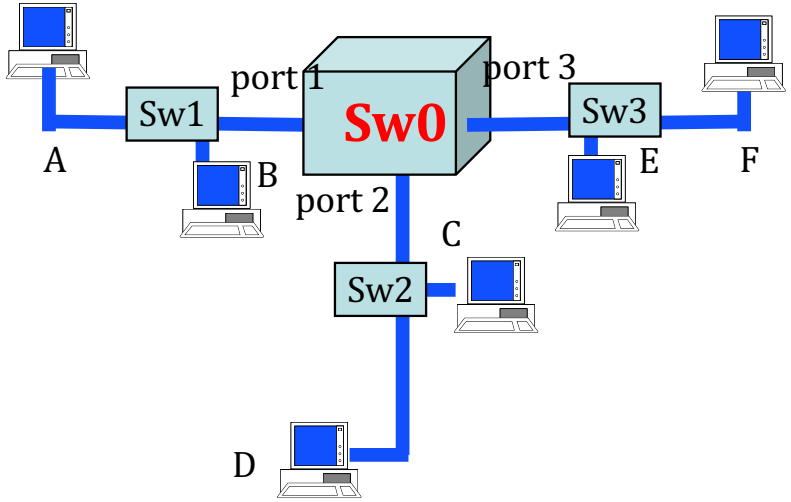
The loop is broken at one link.

Path from A to B is not optimal:

- All frames go through the spanning tree
- Less efficient than shortest path - some more sophisticated bridges implement Shortest Path Bridging (see “link state” lecture) instead of STP, which is the default

How many frames can be transmitted in parallel (all boxes are switches, i.e. bridges) ?

A. 1
B. 3
C. 6
D. > 6
E. I don't know

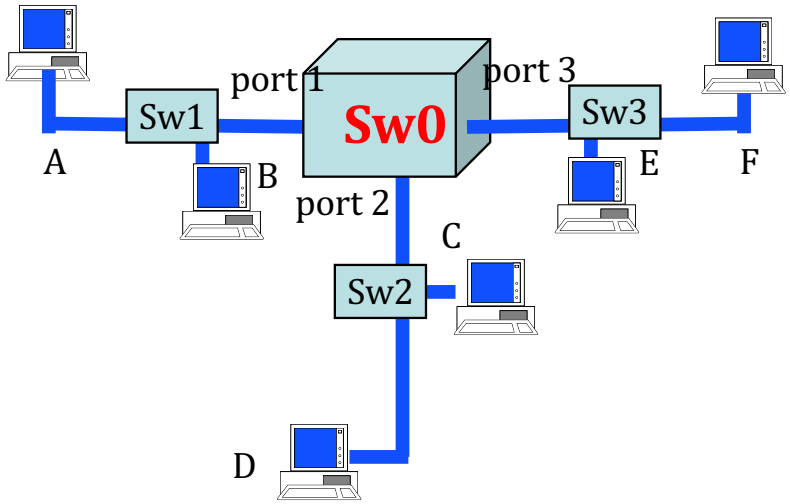


Solution

Answer D

Let us assume that switches can transmit on all ports in parallel (one LAN adapter per port) and the links are full duplex. There can be up to 2 frame transmissions on every link (one in each direction), therefore up to 18 transmissions in parallel.

If links are half-duplex (which is unusual) there can be up to 9 transmissions in parallel.



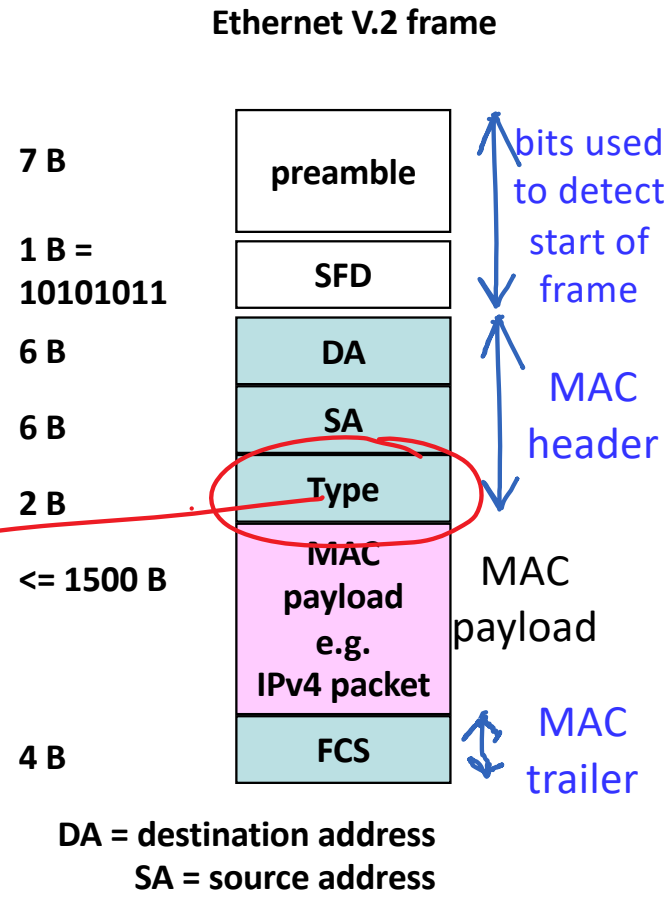
3. Ethernet Frame format

Ethernet frame = Ethernet Protocol Data Unit (PDU)

An Ethernet frame typically transports an IP packet, sometimes also other

Type of protocol contained in the Ethernet packet (hexa):

0800: IPv4
0806: ARP (used by IPv4)
86dd: IPv6
88e5: MACSEC (authenticated/encrypted frame)
88f7: Precision Time Protocol



The preamble is used for the receivers to synchronize (01010101… terminated by 0). With Ethernet, transmission starts asynchronously (stations start independently), and between transmissions, the channel is idle. SFD (start frame delimiter) is used to validate the beginning of a frame.

Destination length is used to indicate the total length before padding. Padding is required if the minimum frame size of 512 bits = 64 bytes is not reached.

There is no frame length field in the header. It is up to the layer using Ethernet to find out the length of the payload. Frames have to be at least 64B and at most 1500B. If needed, padding is inserted to make the frame long enough.

The format shown here is called Ethernet v2. There exists another format on Ethernet called IEEE 802.1. With WiFi the format is similar, but the MAC header contains additional fields.

Error Detection

Frame Check Sequence (FCS) is an error detection mechanism

Why ?

Bit errors due to transmission (Bit error rate $< 10^{-10}$ on cables) can corrupt frame. We want to detect and discard such frames.

How ? A Cyclic Redundancy Checksum (CRC, 32 bits) is computed for every frame and inserted in the FCS field. It is a polynomial code. The receiver recomputes the FCS and checks if equal.

This *detects* frames that have an error. The FCS can detect all single, double, triple errors, all error bursts of length <= 32, most double bursts of length up to 17. The probability that a random collection of bit errors is undetected is 2e-32.

Addressing

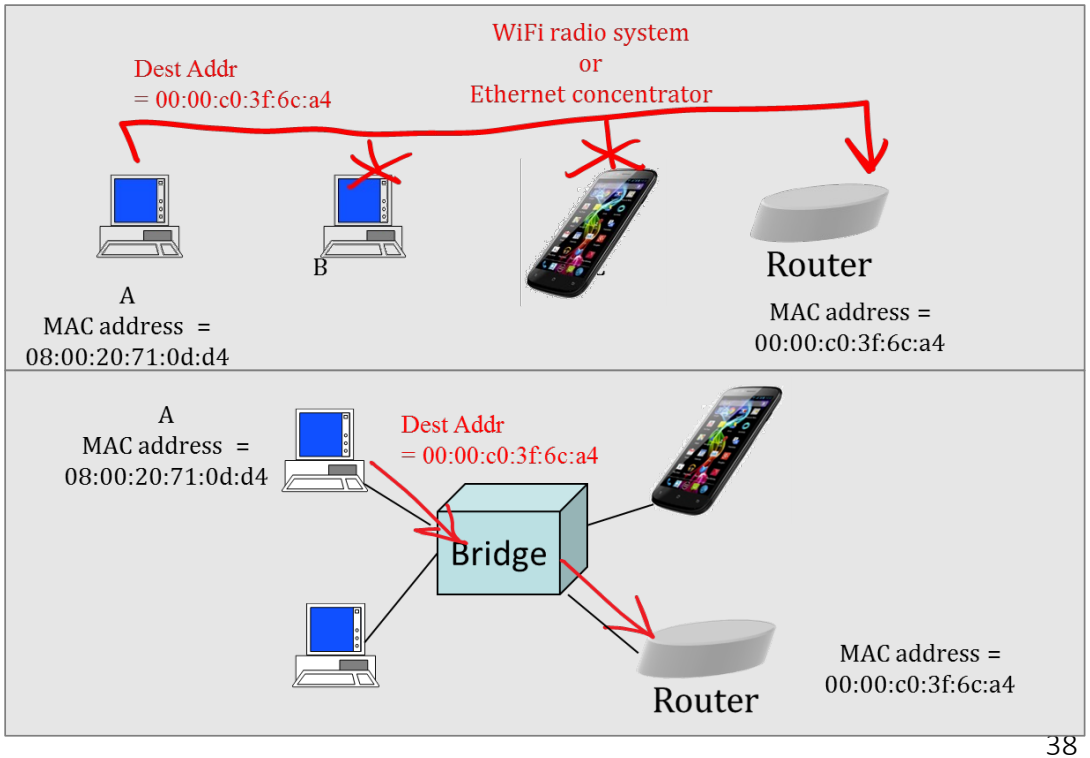
MAC address: 48 bits = adapter number (in principle – can be configured). Unique worldwide, in principle. Sender puts destination MAC address in the frame

On shared medium LAN: all stations read all frames; keep only if destination address matches. Bridges forward frames only to port that needs it (in principle)

Address sent in the clear, no encryption.

On a shared medium LAN (WiFi, Powerline, CAN bus), all systems see the frame and discard it except destination.

With a bridge, only destination sees the frame.



Ethernet addresses are known as MAC addresses. Every Ethernet interface has its own MAC address, which is in fact the serial number of the adapter, put by the manufacturer.
MAC addresses are 48 bit-long. The 1st address bit is the individual/group bit, used to differentiate normal addresses from group addresses. The second bit indicates whether the address is globally administered (the normal case, burnt-in) or locally administered. Group addresses are always locally administered.

When A sends a data frame to B, A creates a MAC frame with source addr = A, dest addr = B. The frame is sent on the network and recognized by the destination.

Data on Ethernet is transmitted least significant bit of first octet first (a strange behaviour dictated by Intel processors). Canonical representation thus inverts the order of bits inside a byte(the first bit of the address is the least significant bit of the first byte);

The first 24 bits are the Organization Unique Identifier (OUI); the remaining 24 bits are allocated by organization that owns the OUI

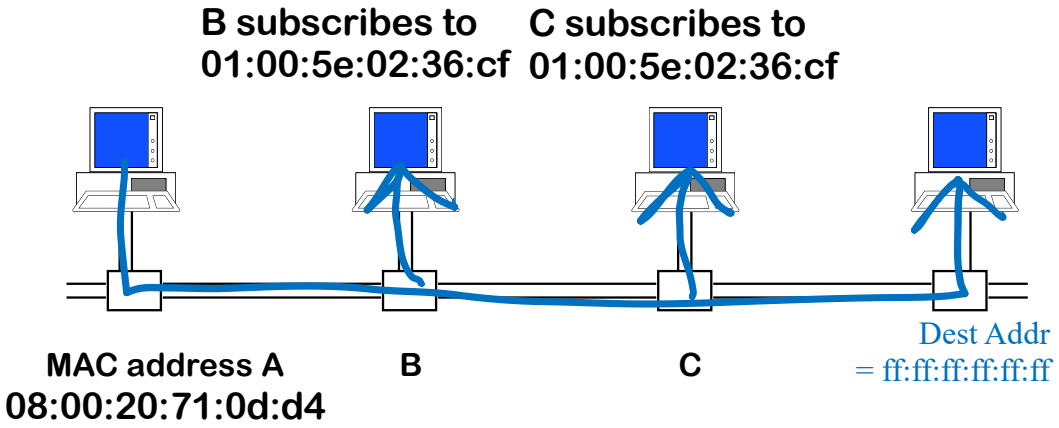
examples of addresses:

08:00:20:71:0d:d4 (a SUN machine)
00:00:c0:3f:6c:a4 (a PC)
00:00:0c:02:78:36 (a CISCO router)
ff:ff:ff:ff:ff:ff the broadcast address

The MAC Broadcast Address

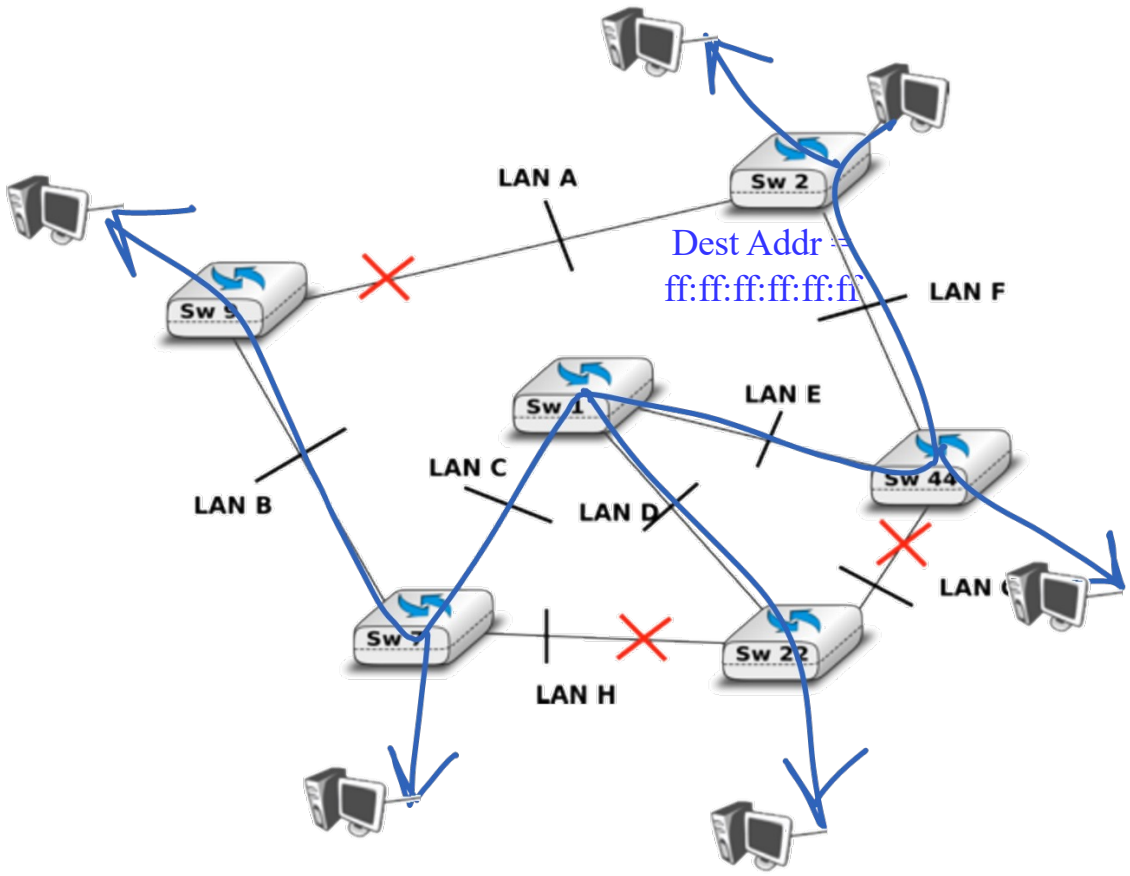
ff:ff:ff:ff:ff:ff = broadcast address

On shared medium LAN, all machines receive packet and do not discard it



With Bridges…

Broadcast frames are sent on all nodes and ports on the spanning tree



MAC Multicast Addresses

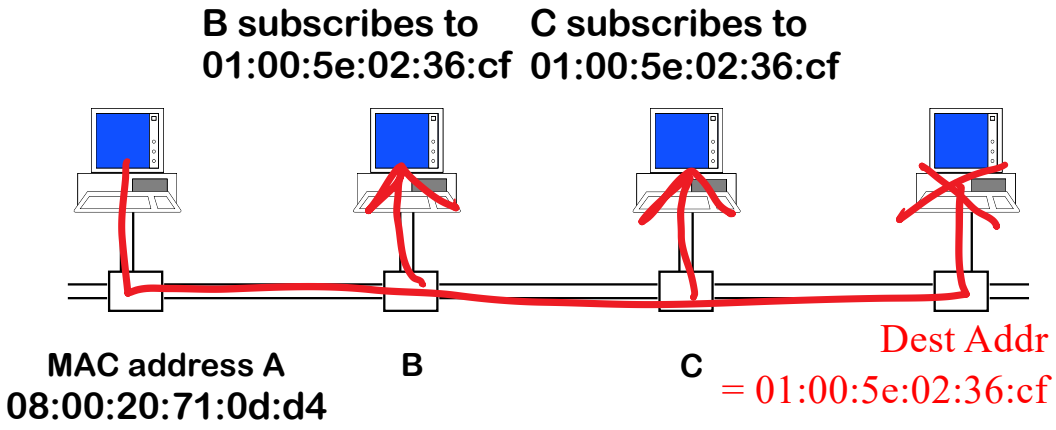
Multicast = Group ;

MAC Addresses with bit 8 == 1 are multicast addresses

Ethernet adapter discards such packets except if host subscribes to address

Non-smart bridges broadcast such frames; smart bridges send only to nodes that subscribe to multicast address using IPv4 or IPv6 subscription protocols
(see “multicast” lecture).

01-00-5e-XX-XX-XX	IPv4 multicast
33-33-XX-XX-XX-XX	IPv6 multicast
01-80-c2-00-01-81	Precision Time Protocol

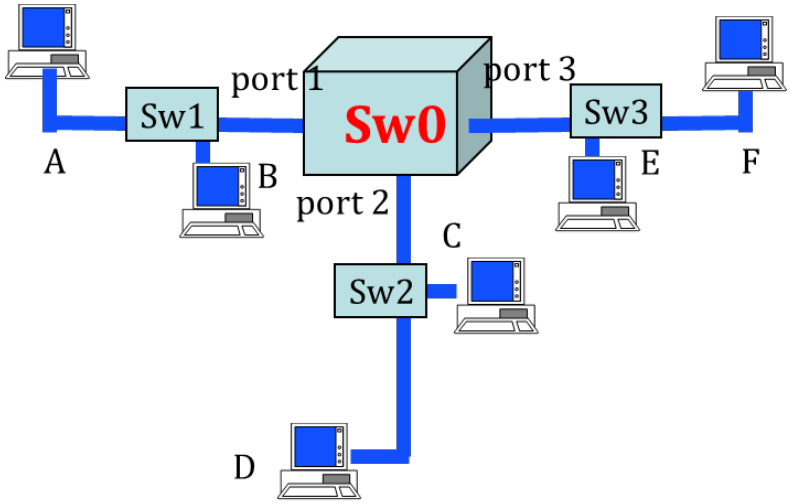


The protocol type in an Ethernet frame header indicates whether…

A. … the data is a TCP packet or not
B. … the data is an IPv4 packet or not
C. Both
D. None
E. I don’t know

We have a working environment with IPv4, using Ethernet bridges plus other things. We add IPv6 to the network. Do we need to change the bridge configurations ?

A. No, since bridges look only at MAC addresses
B. Yes since the protocol type is different
C. It depends whether SLAAC is used
D. I don't know



Solution

Answer B: the Ethertype indicates an IPv4 or IPv6 packet

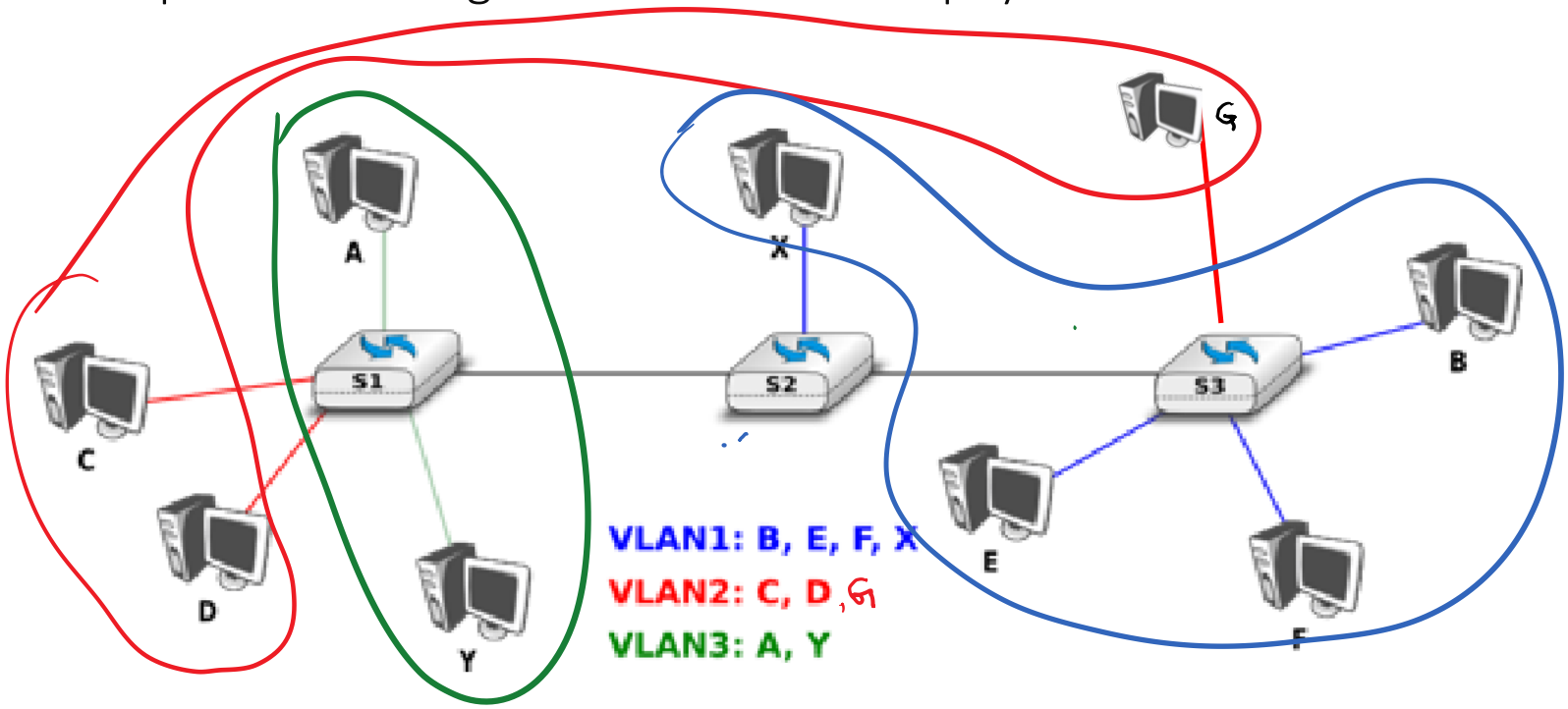
Answer A: the bridges work only at MAC layer and continue to work the same whether IPv4, IPv6 (or something else) is used.

One exception: smart “bridges” that build multicast trees by looking at IP layer group membership

4. Virtual LAN

Why invented ?

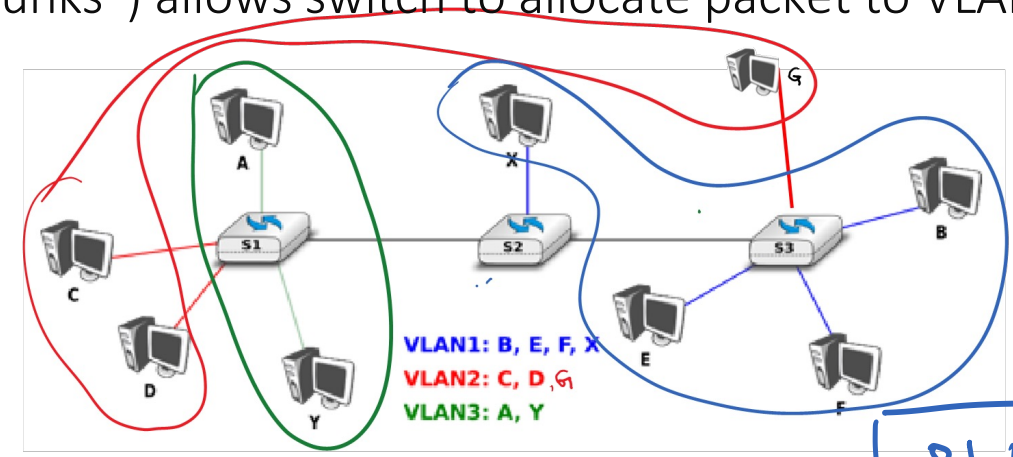
- Some systems use LAN as access control (e.g. all external web servers machines on one LAN)
- Some applications work only on LANs (e.g. airplay, miracast, iTunes)
- Goal: Decouple who belongs to which LAN from physical location

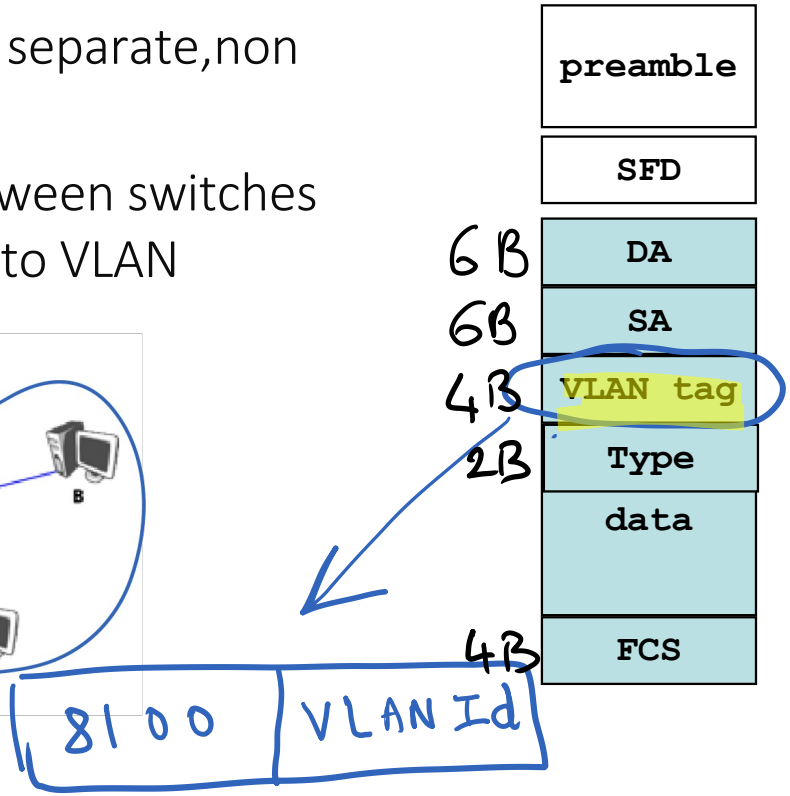


Virtual LAN

VLANs are handled as if they were physically separate: different forwarding tables, different spanning trees. How does it work ?

- Configure (by network management) which switch port belongs to which VLAN
- Switch handles ports of different VLANs as separate,non communicating worlds
- VLAN label in Ethernet header on links between switches (“trunks”) allows switch to allocate packet to VLAN





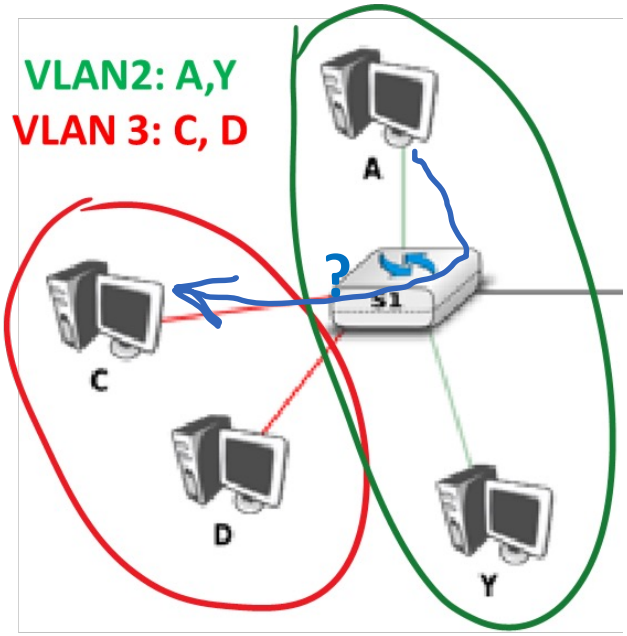
A has an IP packet destined to C; S1 is a switch; what should A do ?

A. Send an ARP request in order to obtain the MAC address of C (then send packet to this MAC address)

B. Send an ARP request in order to obtain both the MAC address and the VLAN label of C (then send packet to this MAC address and to this VLAN)

C. None of the above

D. I don't know



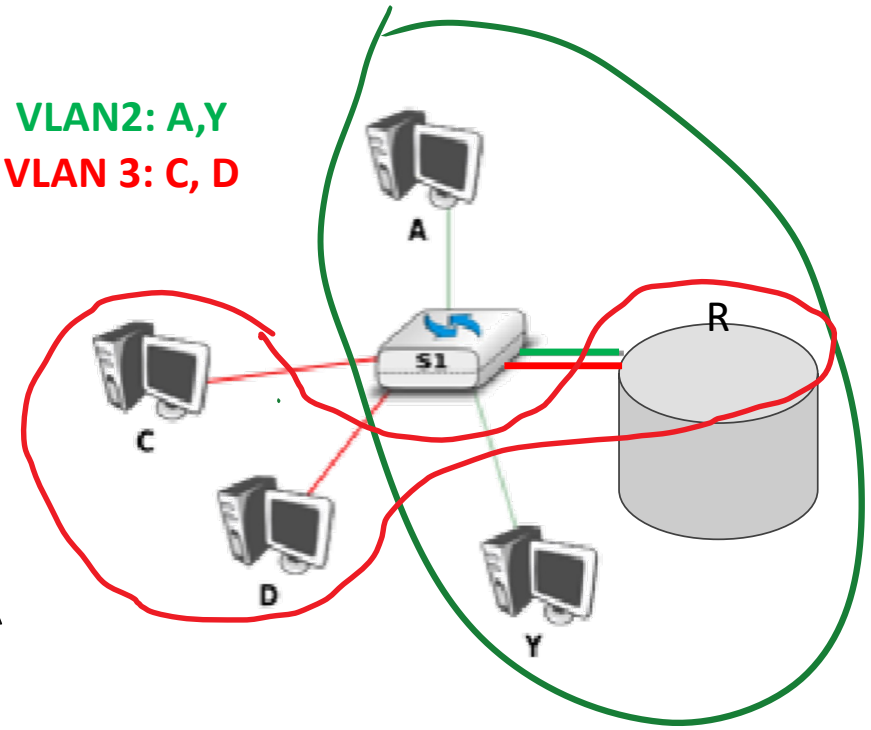
Solution

Answer C
VLANs 2 and 3 are two separate LANs, it is not possible to send directly from A to C, must go through a router R

R belongs to both VLAN2 and VLAN 3 (and has several logical interfaces)

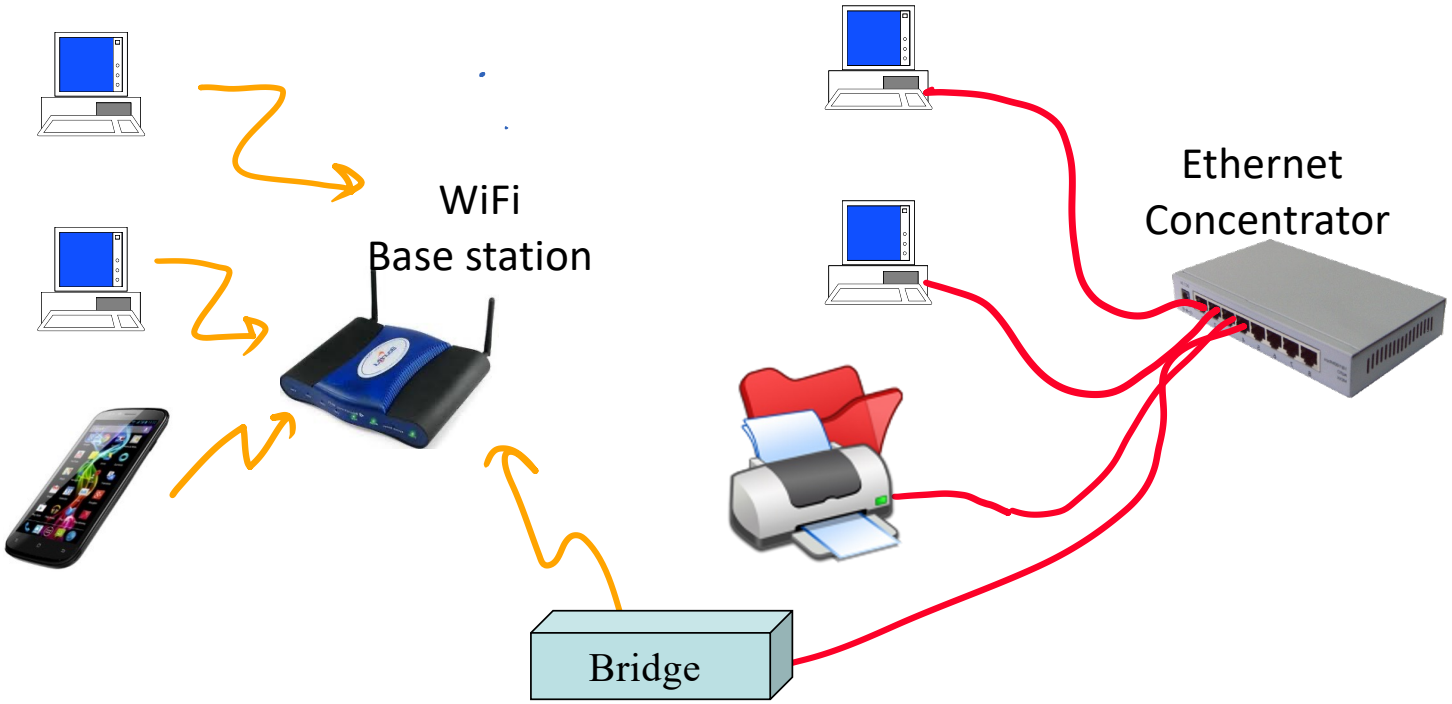
A sends an ARP request to obtain R's MAC address and sends IP packet to R's MAC address. R receives packet on its green VLAN, goes to routing table and sends packet to red VLAN.

Usually, R and S1 are in the same box, else R is connected to a trunk port, i.e. a port on which VLAN tags are present.



5. Bridges can be used between different MAC layer technologies

We have here one single LAN



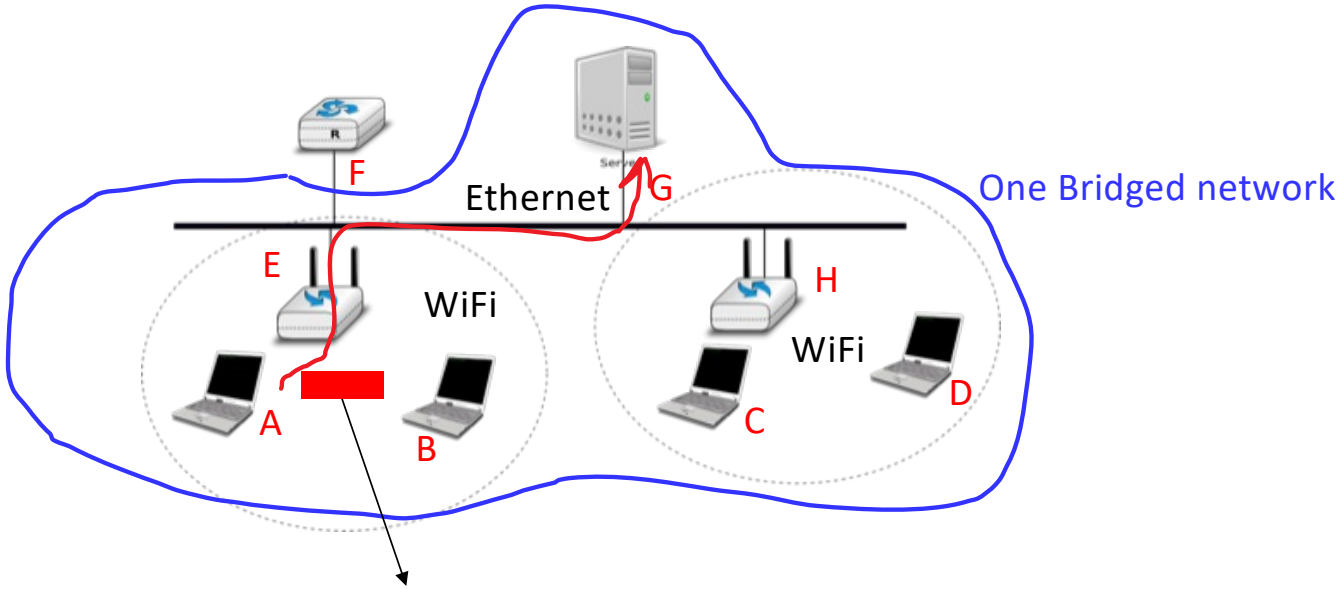
WiFi Access Points

A WiFi access point is usually the combination of

A WiFi base station (implements the WiFi MAC protocol)

A bridge with an Ethernet backend (called the Distribution System)

In this case, WiFi frames differ a bit from Ethernet frames in that there can be 3 MAC addresses: source, destination and access-point.

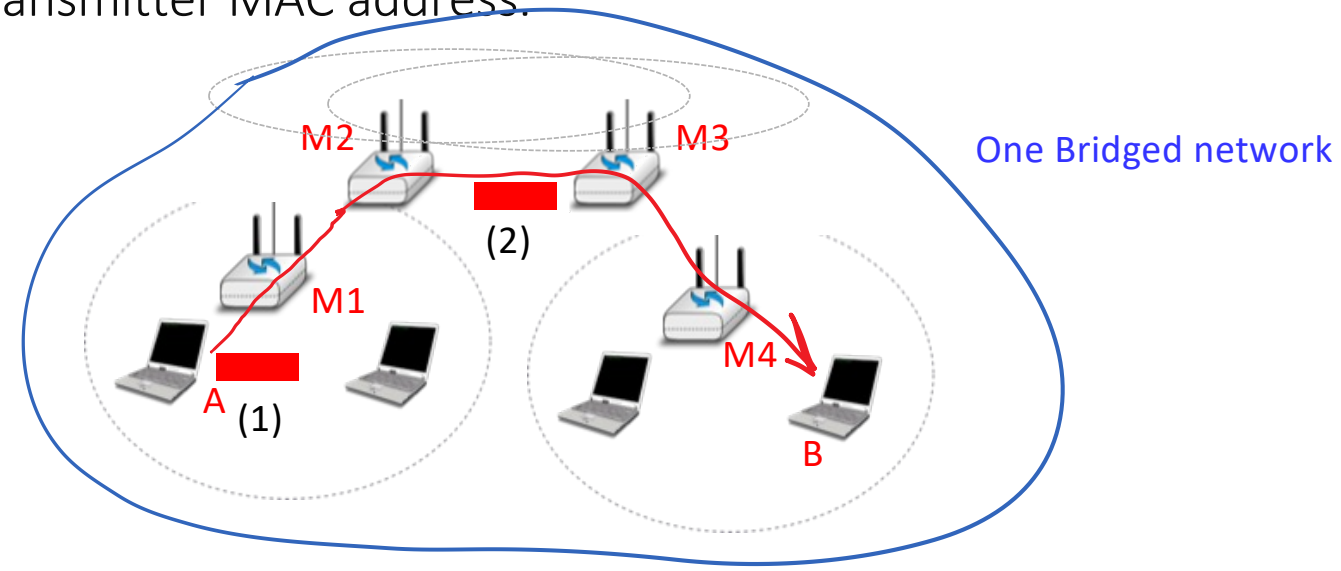


In packet sent by A to G: source MAC addr = A, dest MAC addr = G, access-point address = E

WiFi Access Points with Wireless Distribution System

In some cases, the interconnection between WiFi base stations can be done using WiFi relays, which act as bridges.

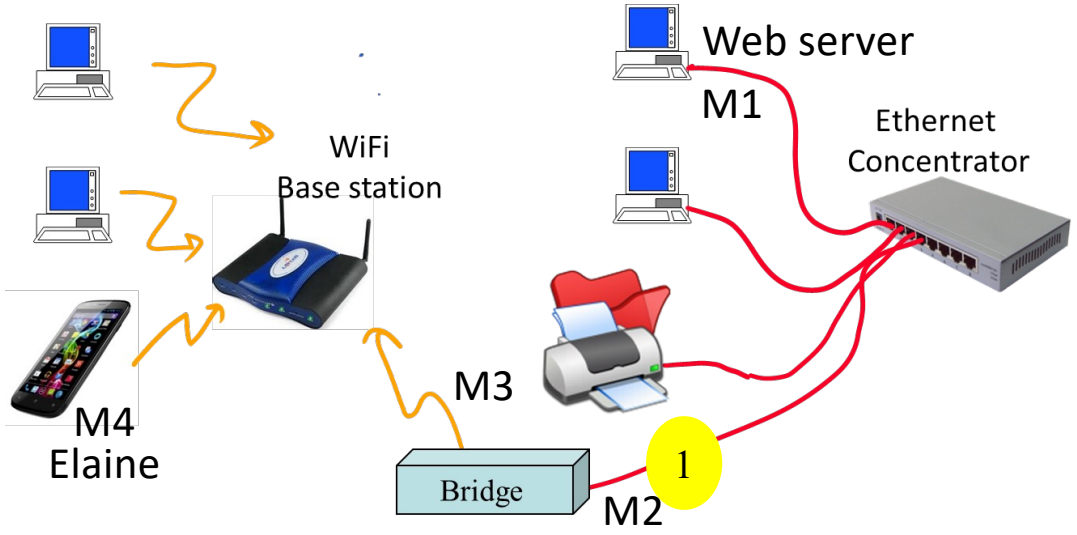
In this case, the WiFi frames may carries 4 MAC addresses: source, destination and receiver / transmitter MAC address.



In packet sent by A to B: at (1) source MAC addr = A, dest MAC addr = B, access-point address = M1
At (2) : source MAC addr = A, dest MAC addr = B, tx-access-point address = M2,
rx-access-point address=M3

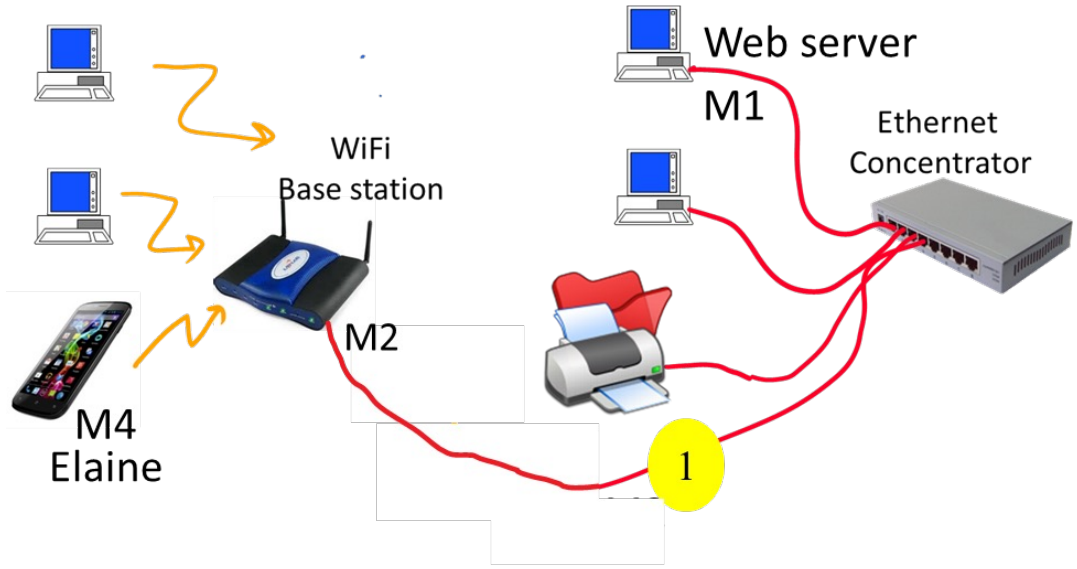
Web server sends one IP packet to Elaine. At (1), the MAC destination address is…

A. M1
B. M2
C. M3
D. M4
E. I don't know



Web server sends one IP packet to Elaine. At (1), the MAC destination address is…

A. M1
B. M2
C. M4
D. I don't know



Solution

Answer D

Answer C

In both cases the answer is M4.

In both cases there is only one large bridged network.

In the first case, the WiFi base station is not connected to a wired infrastructure. Instead, it is connected wirelessly to a bridge. This is an atypical scenario – it is not very efficient because all WiFi traffic that goes to the bridge uses WiFi twice, once to access the base station, once to go from base station to the bridge.

The second case is the classic use of an access point. The base station contains a bridge and is itself connected by Ethernet to the rest of the network.

6. Security Aspects

MAC addresses are sent in the clear. Attacks that result ?

- Eavesdropping (hearing someone else's data)
- Free riding (inserting a device into a network without authorization)
- Impersonation (send data with someone else's MAC address; replay attacks)

Solutions (MACSEC)

- Access control require user to show credentials before allowing a given MAC address into the network;
  - shared secret (e.g. WiFi WPA-Personal): user knows a password, same for all
  - per-user authentication (e.g. WiFi WPA-Enterprise): user has a personal password
- Authentication: every MAC frame is signed using cryptography and numbered
  - prevents free riding
  - prevents impersonation if access control is per user;
- Encryption: MAC frame payload is encrypted (not MAC address) -> prevents eavesdropping

Examples: At EPFL, WiFi uses access control, authentication and encryption; impersonation is not possible. In contrast, Ethernet is not secured (only non authenticated verification of MAC address). At home: WiFi WPA-Personal provides access control but does not authenticate individual users. Impersonation is possible by who knows the shared password.

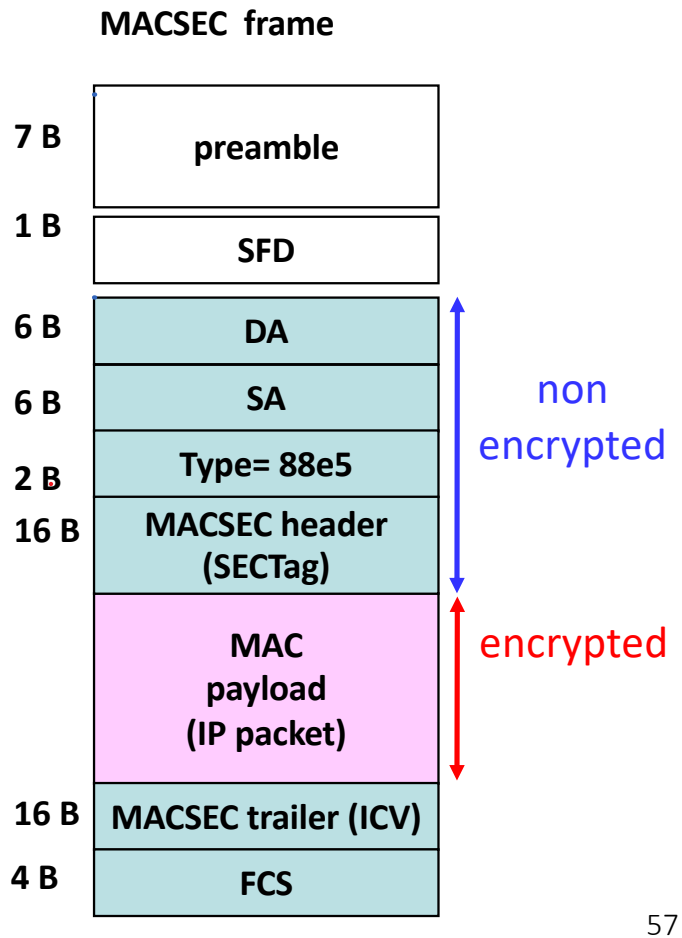
MAC Security (MACSEC)

Host negotiates access with WiFi base station or Ethernet switch, who contacts an authentication server.

If host is accepted, Ethernet frames between host and WiFi BS /Ethernet switch are authenticated and encrypted.

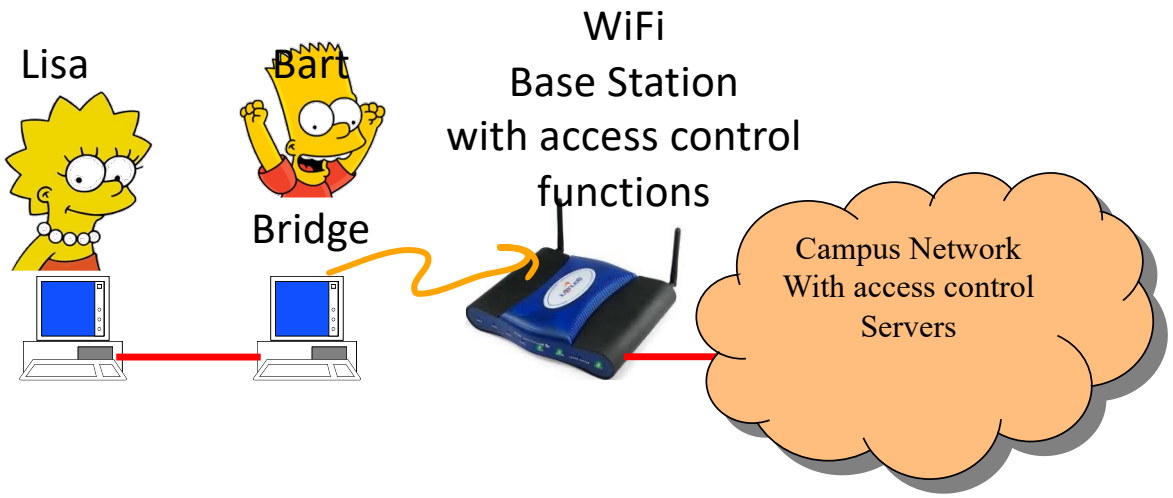
Authentication covers MAC payload and MAC source address.

IEEE 802.1X for access control
IEEE 802.1AE for authentication and encryption



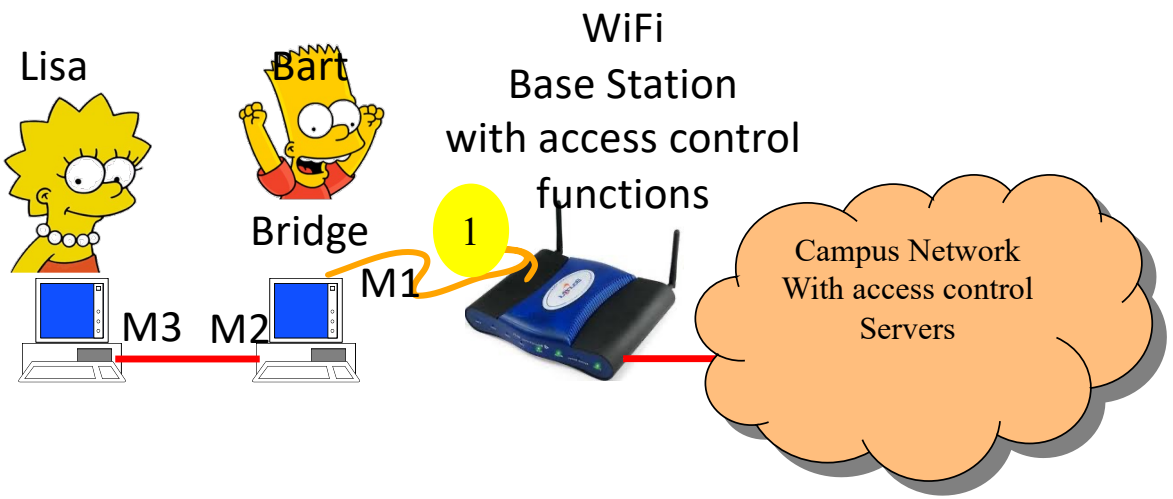
Bart, who is EPFL student, connects his PC to EPFL WiFi, which uses per-user access control, authentication and encryption. Bart configures his PC to be a *bridge* between Ethernet and WiFi and allows Lisa, who is not allowed to access EPFL, to connect to his Ethernet port. Will Lisa be able to connect to the Internet in this way ?

A. Yes, provided that Lisa can configure an IP address on the same subnet as Bart
B. Yes, it will work in all cases
C. No, it will not work in any case
D. I don't know



Solution

Answer C



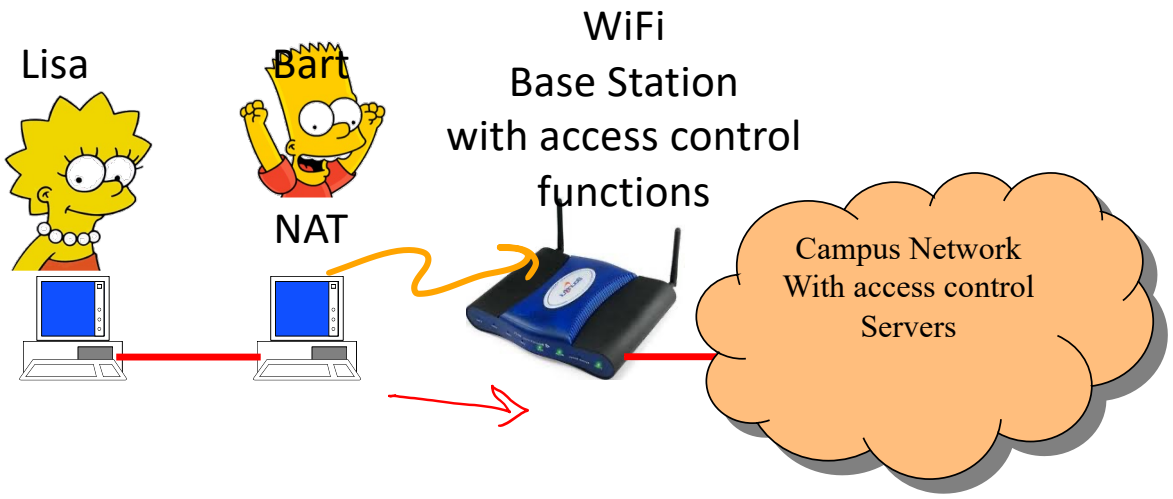
Access control accepts Bart's MAC address; MAC frames at (1) are authenticated to verify that the source MAC address is M1 (Bart's MAC address on WiFi). WiFi base stations accept only authenticated MAC frames.

Lisa's data will appear at (1) with source address = M3(≠M1). Even if Bart's PC authenticates Lisa's MAC frames, M3 is not a valid MAC address for this authentication data, base station discards Lisa's frames.

Lisa will not be able to connect to campus network in this way.

Bart, who is EPFL student, connects his PC to EPFL WiFi, which uses per-user access control, authentication and encryption. Bart configures his PC to be a *NAT* between Ethernet and WiFi and allows Lisa, who is not allowed to access EPFL, to connect to his Ethernet port. Will Lisa be able to connect to the Internet in this way ?

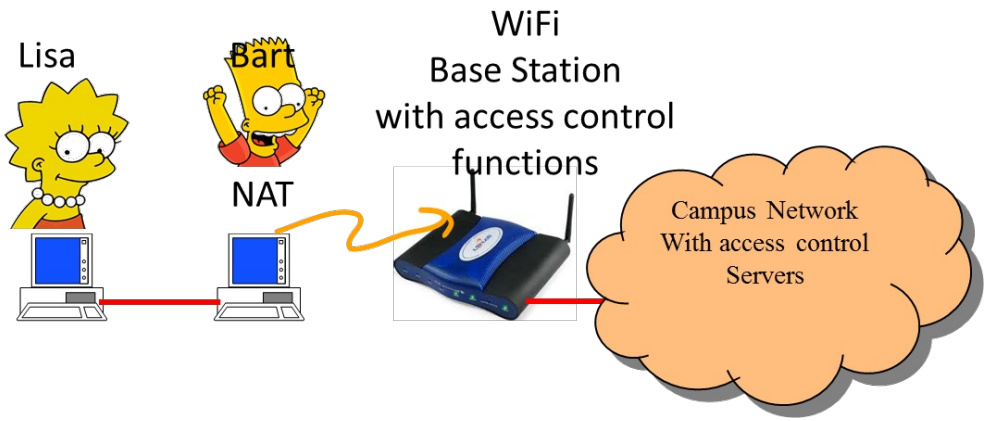
A. Yes if Bart himself is not behind a NAT
B. Yes
C. No, it will not work in any case
D. I don't know



Solution

Answer B, it will work.

There are 2 layer-2 domains separated by the NAT. The WiFi frames seen by the WiFi base station belong to the WiFi domain i.e. they are seen as originated by Bart, and thus they are accepted as legitimate. Lisa can access the campus network without being a legitimate user !



Answer A is incorrect: it will work if Bart is behind a NAT. In this case, it is the NAT that must connect to the EPFL WiFi network. Since we assume that Bart could connect to EPFL, it means that the NAT box succeeded to connect to EPFL. In this case, Bart can connect and Lisa also. The EPFL network sees Bart's NAT and thinks that Bart and Lisa are programs on Bart's NAT. An example where this occurs is if Bart uses a PC with virtual box, connects it EPFL, and then uses a guest OS inside the PC.

Conclusion 1

The MAC layer for wireless medium (WiFi) takes care of sharing the radio waves in a local environment.

The MAC layer for wired medium (Ethernet) was originally designed for sharing a cable; in this form it uses a protocol, called CSMA/CD very similar to WiFi.

The MAC layer for wired medium (Ethernet) has now got rid of any protocol (“full duplex Ethernet”), using interconnected bridges. It thus forms an interconnection layer of local scope.

Conclusion 2

We have seen two mechanisms to interconnect devices: the network layer (with IP routers) and the MAC layer (with bridges). What are the differences ?

Bridges use routing tables that are not structured. A bridge must search the entire table for *every* packet. The table size and lookup time are prohibitive if network becomes very large. We prefer routers for large networks.

Bridges are independent of whether IPv4 or IPv6 is used. They are plug-and-play and are independent of IP numbering plans.

Historically, routers were in software and bridges in hardware. Nowadays every combination exists