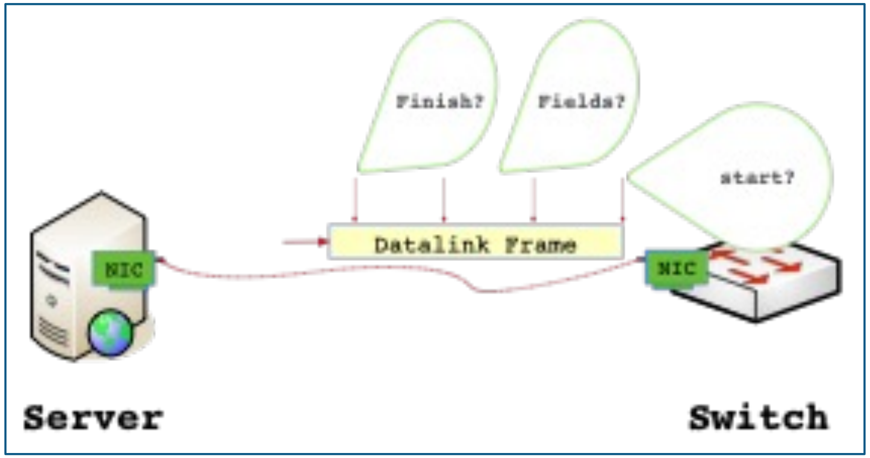
LAN SWITCHING AND SPANNING TREE PROTOCOL

May/2025

Ethernet Frame

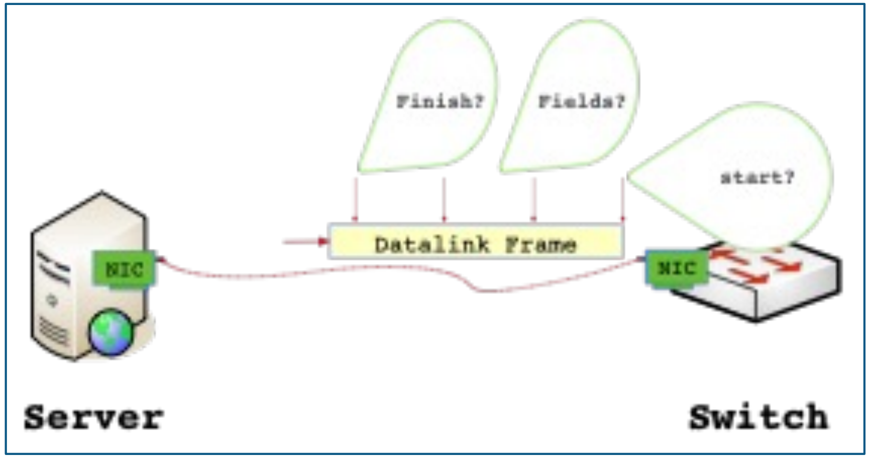
Destination Address	Source Address	Ethertype	Payload	CRC32
48-bit MAC address	*48-bit MAC address*	*16-bit Multiplexing Key*	*Variable-length*	*Error Control*

- MAC Addresses
  - Unicast: Represents a single station
  - Broadcast: All stations in LAN
    All 1's address: 0xffffffffffff
  - Multicast: A subset of stations



NIC: Network Interface Card

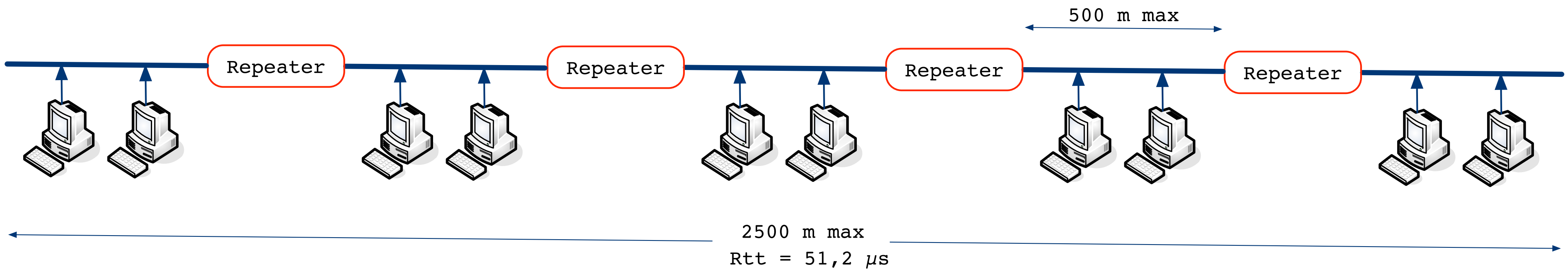
- Each NIC:
  - NIC has a MAC address of its own
  - Sends frames
  - Receives frames
- Frames accepted by a NIC:
  - If promiscuous mode is <u>SET</u>
    - All frames
  - If promiscuous mode is <u>NOT</u> SET
    - Frames which Dest MAC is == NIC's MAC
    - Or frames having Dest MAC == Broadcast (All 1)



Limitations of Ethernet

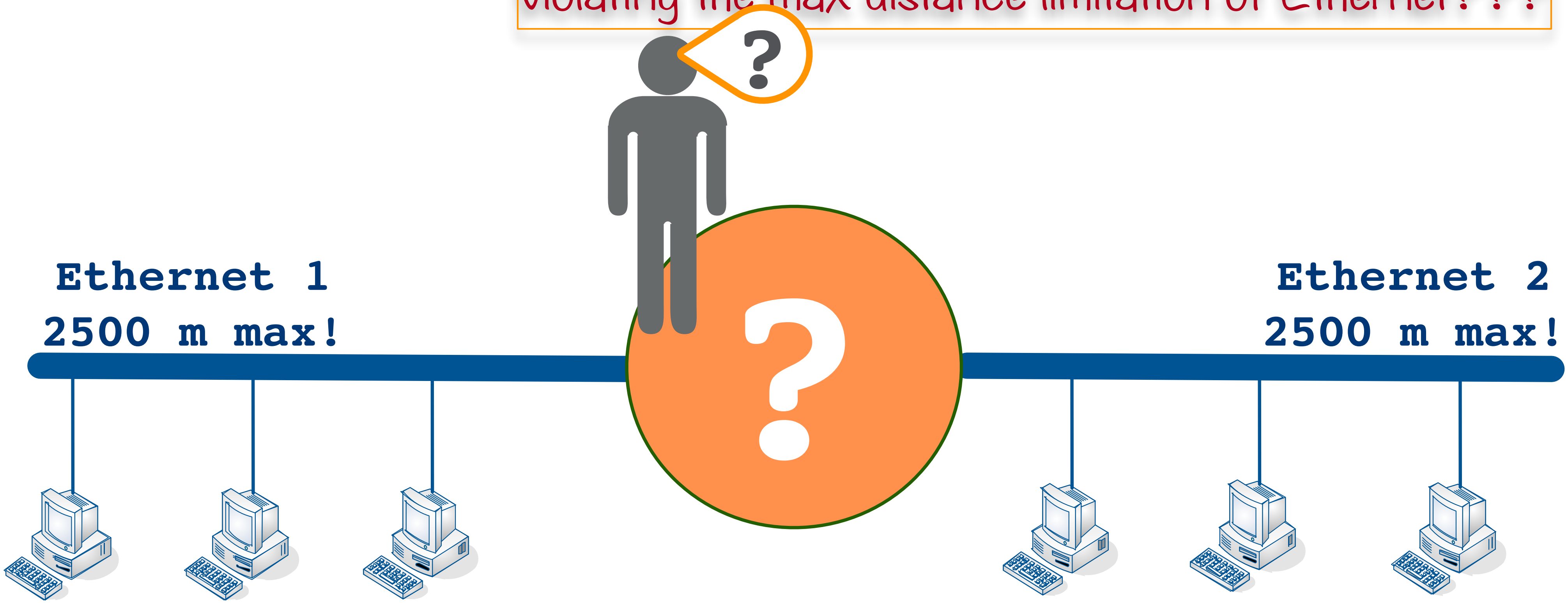
Multiple Ethernet **segments** can be joined together by using ***4 repeaters*** at most

- A *repeater* is a device that *regenerates digital signals*.
  - No more than four repeaters
  - 2500m max
  - Limited total number of stations (Computers)
- Broadcast media: A sent frame is received by all the stations, this is inevitable
- Half duplex: Only one frame can be being transmitted at any one time
- HUB = Multiport *repeater*

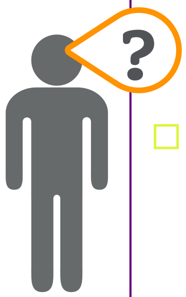


Communicating two max Ethernets

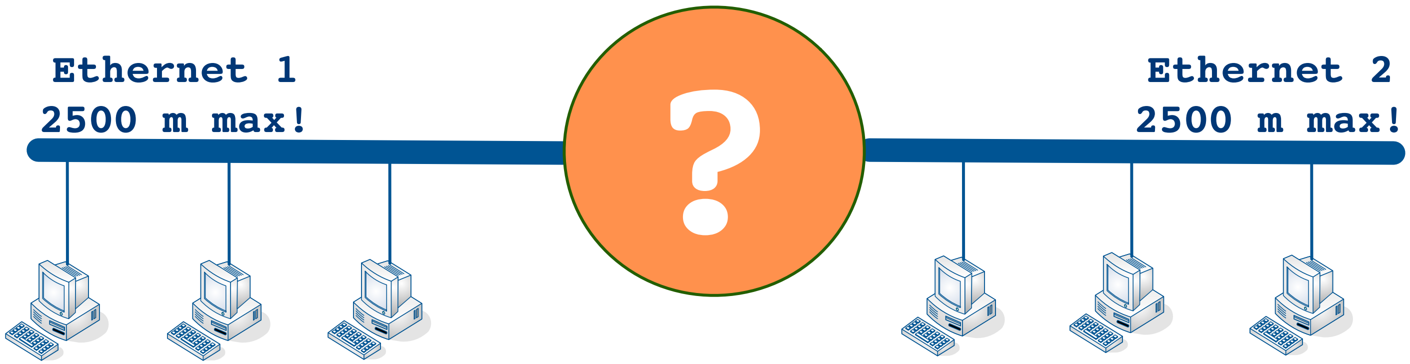
What network device can communicate E1 with E2
without
violating the max distance limitation of Ethernet???



Connecting two max Ethernets



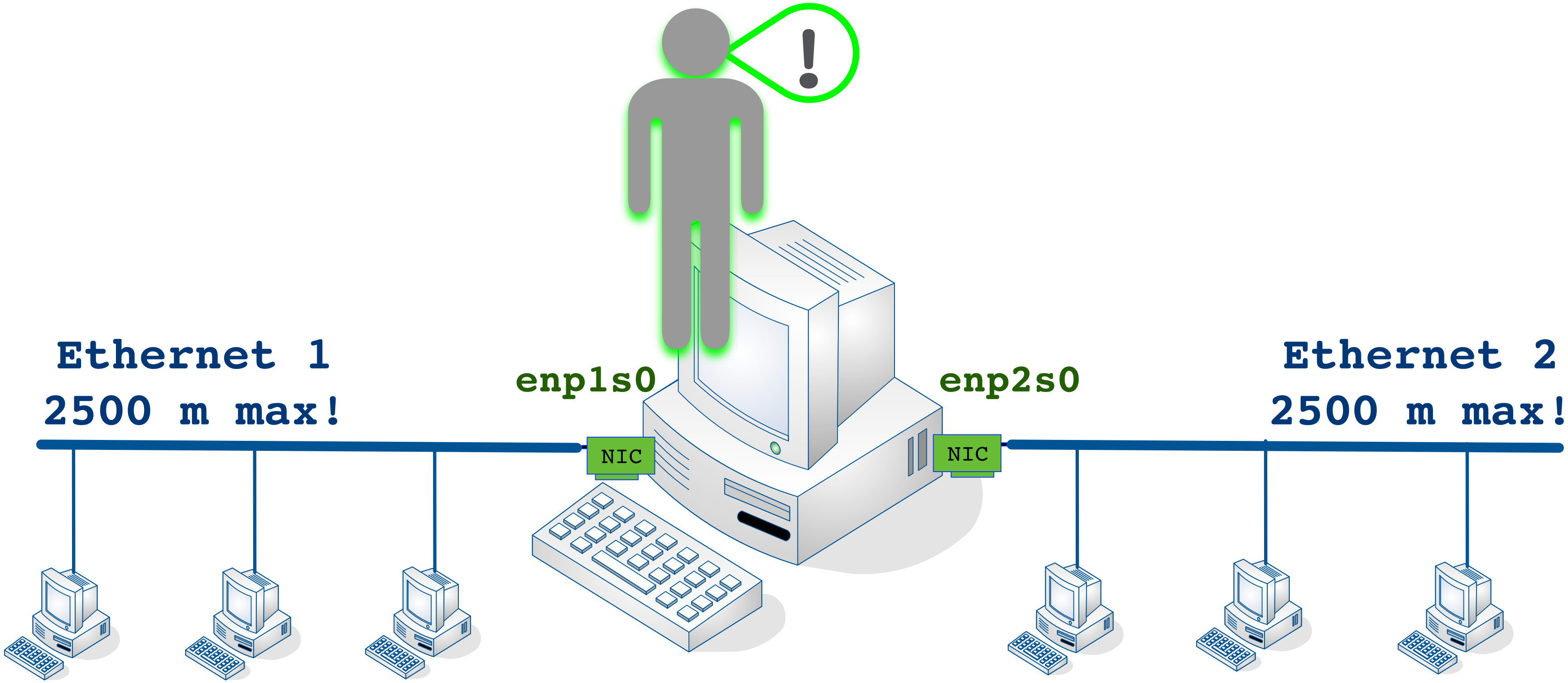
- **A) Repeater in between them? *No!***
  - It might exceed the physical limitation of the Ethernet
- **B) Hubs? Hub simply regenerate electrical signals: *No!***
  - Same limitations as repeaters.
  - Hubs are Physical-layer devices
- **C) Network equipment** that forwards **frames** between segments
  - **LAN Switch!**



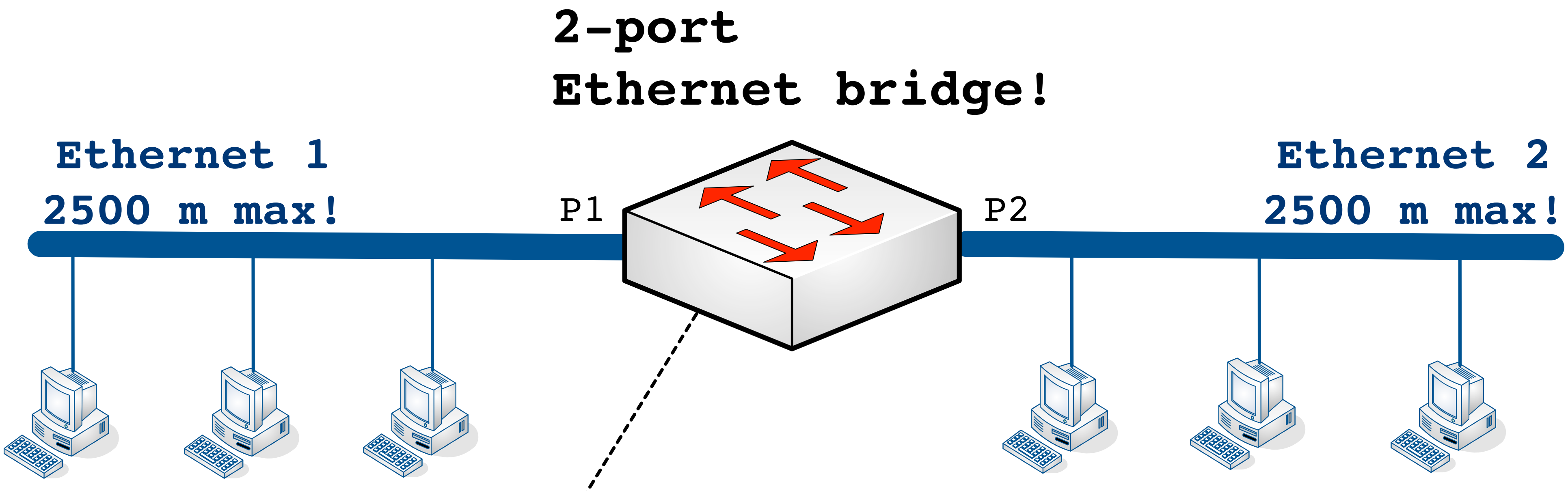
Connecting two max Ethernets

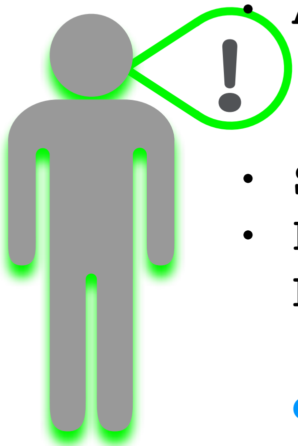
Bridge or **Switch** service (Software) communicates Ethernets as though they were one!

```
doCopy(receivedFrame, destinationPort);
```



Connecting two Ethernets: Bridge or Switch





- Electrically separates Ethernets
- At least, 2 ports:
  - NIC on port 1
  - NIC on port 2
- Store/Forward device = S/F
- Bridge service (Software) communicates Ethernets as though they were one!

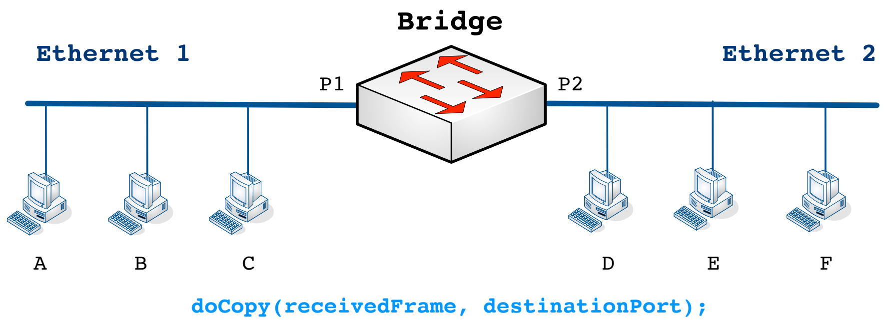
```
doCopy(receivedFrame, destinationPort);
```

Extended LAN domains

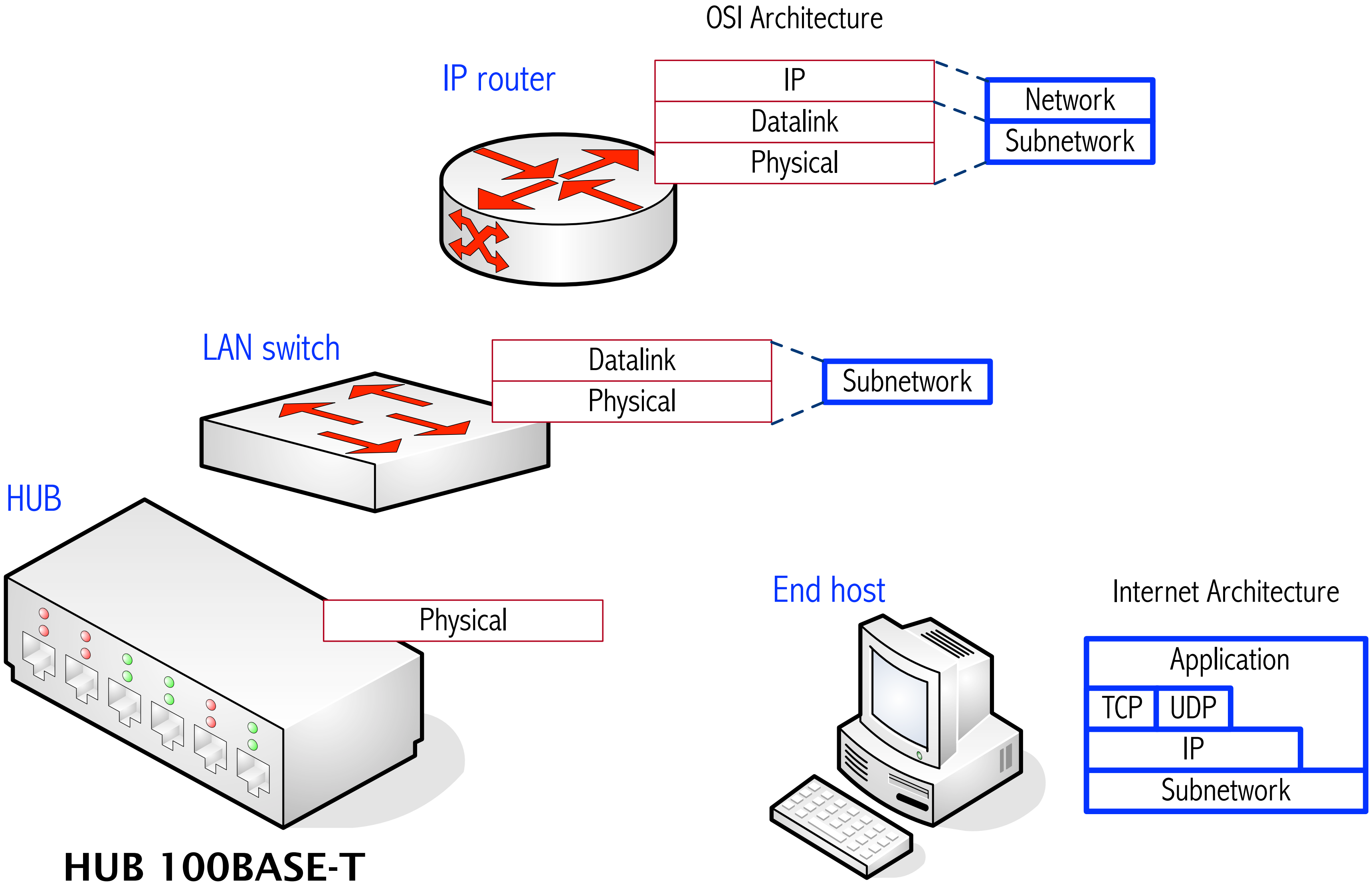
- Can A collide with B? Yes, it can since A and B are connected to the *same* Ethernet segment
- Can A collide with X? No, since A and X belong to different Ethernet segments

- There exist TWO segments (Or collision domains):
  - Ethernet segment 1: A, B, C and bridge port P1
  - Ethernet segment 2: D, E, F and bridge port P2

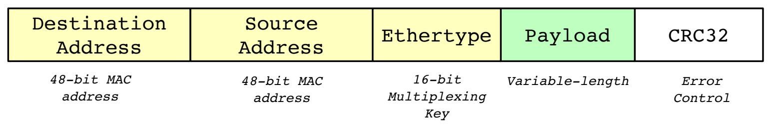
- Note, however that there is only one Extended LAN (Network)
  - When a *broadcast frame* is sent by any host, it is received by *all of the* network hosts. We say that the LAN spans a single BROADCAST DOMAIN



Devices and Network Architecture



LAN Switching algorithm



11

1. **Receive.**
   Receive frame and store it in a buffer associated with the port
2. **Learn.**
   Record the frame's source MAC address into the forwarding table
3. **Forward.**

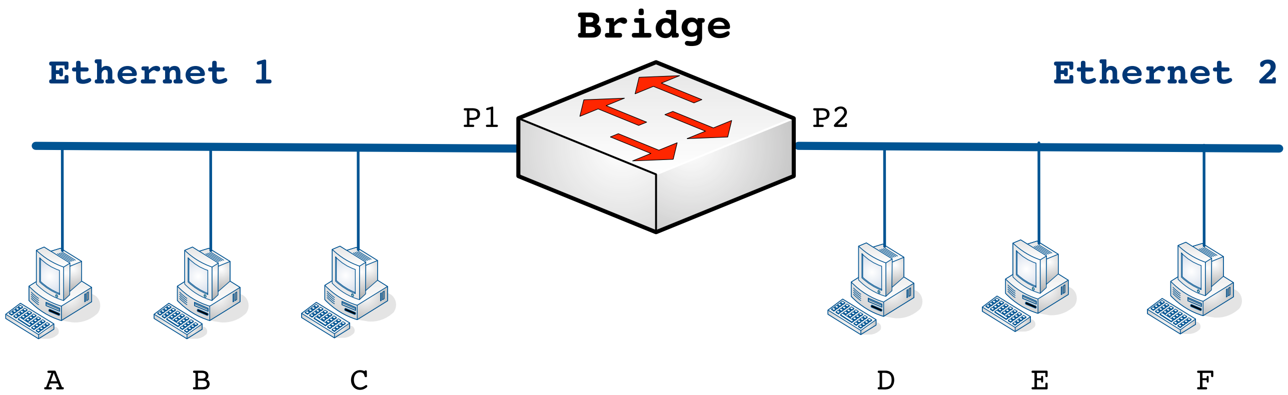
   if destination MAC has not been recorded into the forwarding table yet **or** it is the Broadcast address:

   - Flood the frame (Send it onto all ports except the one it was received onto)

   else

   if destination MAC belongs to the another port

   - send it onto that port *when possible*



doCopy(receivedFrame, destinationPort);

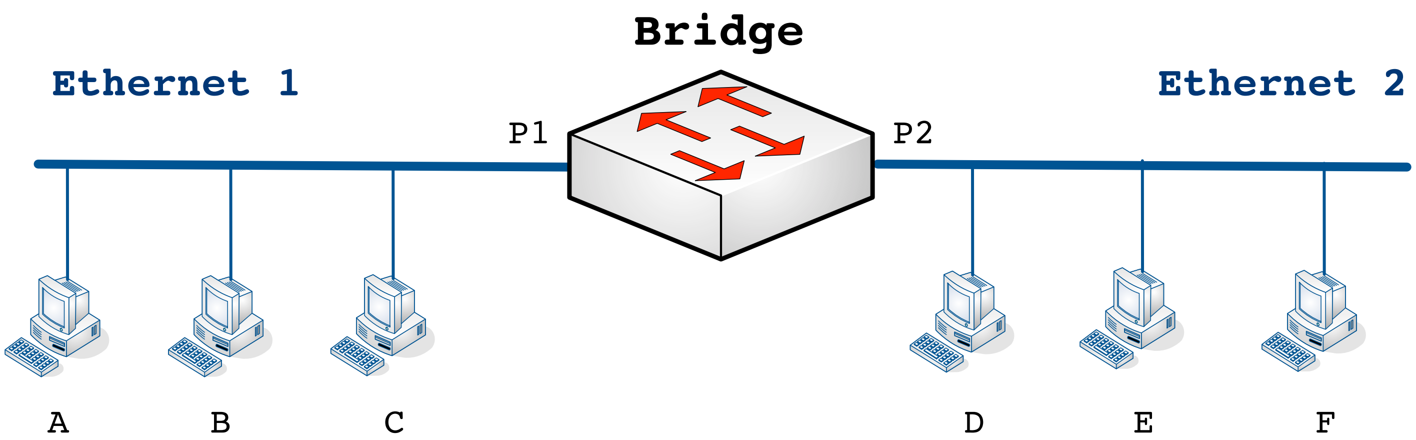
LAN Switching algorithm

Destination Address	Source Address	Ethertype	Payload	CRC32
48-bit MAC address	*48-bit MAC address*	*16-bit Multiplexing Key*	*Variable-length*	*Error Control*


```
void switch_frame(L2frame *frame, unsigned int port){

        doLearnMAC(frame.srcMAC, port);

        if ( port = (doLookupMAC(frame->destinationMAC)) == UNKNOWN){
              doFlood(frame, port);
        }else{
              doCopy(frame, port);
        }
}
```

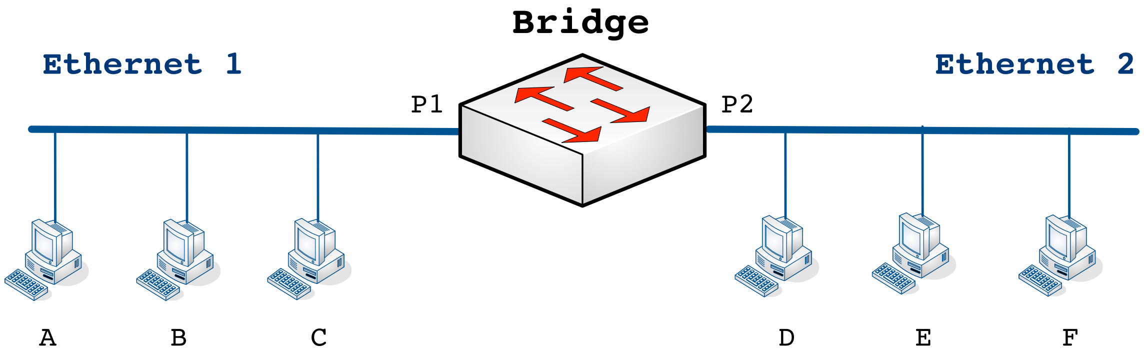


doCopy(receivedFrame, destinationPort);

Exercise 1: Forwarding Table

- In the past, the bridge has received frames from hosts A, B, C, D, E and F.
- For example: The frame sent by A caused the switch to learn its MAC address
- Same with the others

- **Exercise 1.** Check that the adjoining *Forwarding Table* is correct, *i.e.*, that the MAC addresses do belong to their respective segments on port P1 or Port P2



doCopy(receivedFrame, destinationPort);

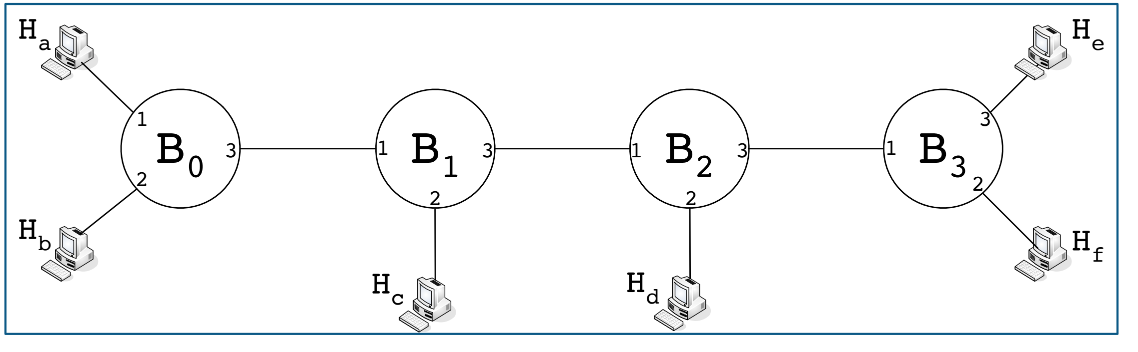
Forwarding table

Host MAC	Port
A	P1
B	P1
C	P1
D	P2
E	P2
F	P2

Exercise 2: Forwarding Tables of an Extended LAN comprised of four switches

Exercise 2. The switches in the adjoining diagram (B_1, B_2, B_3 and B_4) are **powered** up, consequently their forwarding tables are **empty**. Hosts H_a to H_f do the following sends; explain the effect those sends produce in the switches's forwarding tables:

- H_a sends a frame to H_d:
  - *B_0 learns H_a (Associated with Port 1)*
  - *B0's FT doesn't contain H_d, therefore it floods frame; all switches learn H_a*
- H_d sends a frame to H_a:
  - *B_2 learns H_d (Associated with Port 2). This switch already learned H_a (Port 1), it forwards the frame through that port.*
  - *The frame is received by B_1 which allows it to learn H_d (Associated with Port 3). This switch in the past learned H_a (Port 1), therefore it forwards the frame through port 1.*
  - *The frame is received by B_0 which allows it to learn H_d (Associated with Port 3). This switch in the past learned H_a (Associated with Port 1), therefore it forwards the frame through port 1.*
  - *The frame is received by Ha which stack accepts it for further deeancpsulation/demultiplexing.*
- H_f sends a frame which destination MAC is broadcast:
  - *B_3 learns H_f (Associated with Port 2).*
  - *Since the destination MAC is broadcast, B_3 floods the frame; all switches learn H_f and flood the frame, as well*
  - *All hosts receive the frame which payload will be deencapsulated/demultiplexed across the relevant protocols in the stack.*



Exercise 3: Forwarding Tables in real network equipment

Exercise 3. In this exercise we set out to **practically** check two facets of the Lan Switching Algorithm, the *Learning* and the *Forwarding.* For sending test Ethernet frames we'll use again the program from practice #3.

By using that program, each student will send an Ethernet frame which Destination Address is not used by any NIC in Lab B6 *–as yet we're sure about that the MAC addresses that we use here are unassigned.* When we send to one of those MAC addresses the switches we'll necessarily flood the frame since no host might have used one of those addresses as its own MAC address, unless deploying a small scale cyberattack which we are discounting here.

1. Download the program sources:

```
$ wget http://paloalto.unileon.es/cn/Q/dgramPFPACKETSend.c

$ [In Lab B6] wget http://192.168.1.89/cn/Q/dgramPFPACKETSend.c
```

2. Edit the C source program. Change the destination MAC that the program is to use. Enter the following MAC address in function void setDestinationMAC(byte *p); the last two digits will be allocated by the teacher:

0x0102030405**??** *(Assigned by the teacher)*

3. Compile it: $ gcc dgramPFPACKETSend.c –o send

Switch learning and forwarding

4. Note the switch number (253, 252, 251 or 250) and the port number where your NIC is connected:

Switch:____ **Port number:____**

5. Send the frame as super user by assuming your NIC label is enp1s0:

```
$ su
# ./send enp1s0 "Peace!"
```

6. Download the image that contains the Forwarding Tables of the four switches in Lab B6: 192.168.1.250, 192.168.1.251, 192.168.1.252 and 192.168.1.253 and that must have already been uploaded by the teacher:

```
$ wget http://192.168.1.89/FT.253.jpg
$ wget http://192.168.1.89/FT.252.jpg
$ wget http://192.168.1.89/FT.251.jpg
$ wget http://192.168.1.89/FT.250.jpg
```

The download IP address might be different. ***Check with the teacher!***

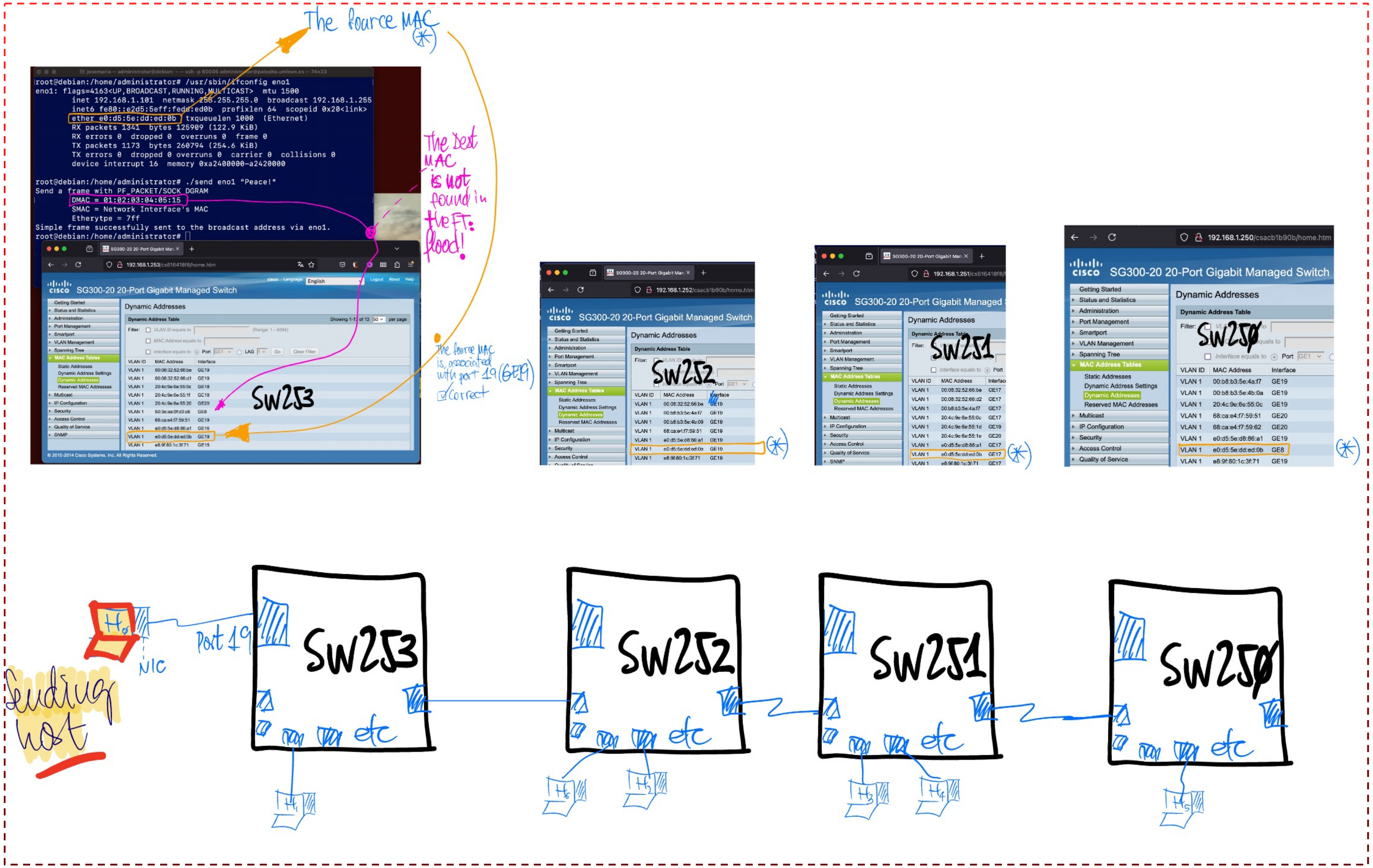
The analyses that we set out to do which involve the switches forwarding tables can only tentatively illustrate the forwarding algorithm, on the face of it, an exhaustive proof should result impractical and very expensive. Our only aim here is to gain familiarity with the notion of FT by using real CISCO switches.

7. Execute ifconfig to print out the MAC address of your main NIC:

Your NIC's MAC Address: __:__:__:__:__:__

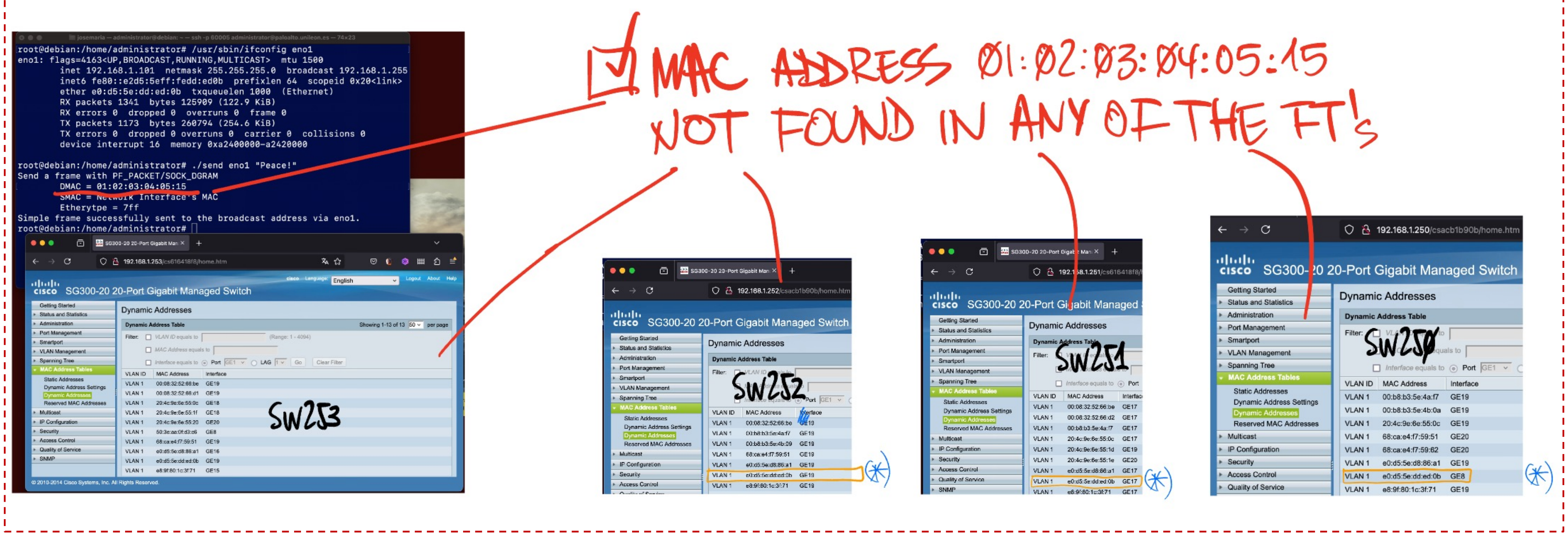
Switch learning and forwarding

8. **[LEARNING]** Check that your switch did learn the MAC address used by your NIC by highlighting it on the relevant switch FT corresponding to your direct-connection switch (The specific switch your host is connected to by a TP cable). *To help you understand, I have done an example below.* You must do the same in your Lab Book, this time using the FTs that the teacher will make available for you.



Switch learning and forwarding

9. **[FORWARDING]** Since none of our host's NICs has the MAC address that we use as Destination MAC in the frames that we send, none of the FTs should contain the MAC address that you have used as Dest Mac. *In my case the Dest MAC is 01:02:03:04:05:15*. Check that that MAC is not found in any of the FTs:



Switch learning and forwarding

10. [FORWARDING] Resend the frame by using ff:ff:ff:ff:ff:ff as Destination Address and perform the same checks as in step 8. This is, essentially the same exercise, so you'll use the **same FT's** we gave you earlier. We simply seek your providing us a correct explanation, highlighting that the Lan Switching Algorithm behaves equally when the Destination Address is unknown as when the destination address is the broadcast address

- Edit the program and set the broadcast address as destination
- Compile it
- Run it as su
- Check that the switches, all of the did flood the frame by observing that all of the FT's contain an entry for your SOURCE MAC.

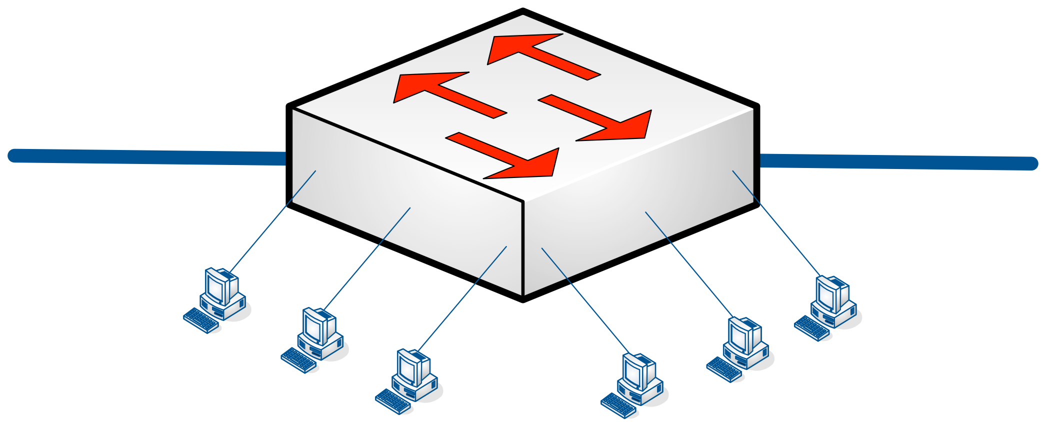
20

Switched LANs and Spanning Tree Protocol

Broadcast and link redundancy cause major trouble: broadcast storms

LAN Switch

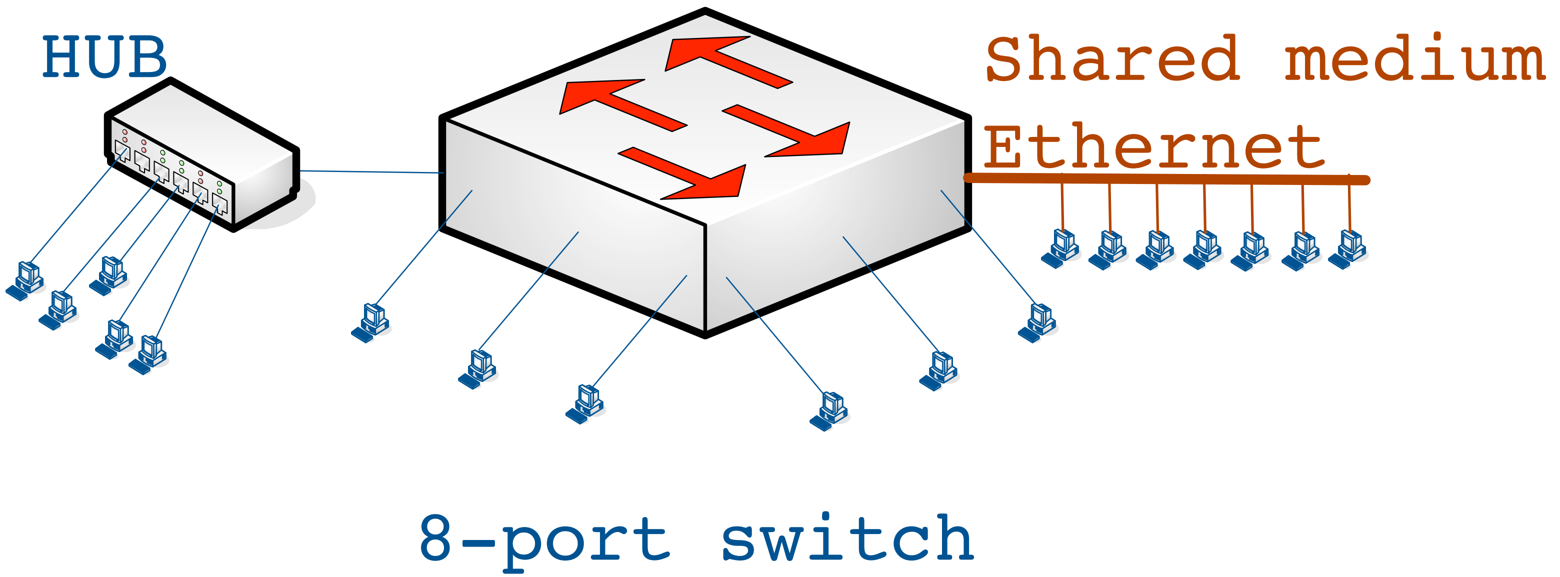
- S/F device
- 8 ports = 8 collision domains
- 1 broadcast domain = 1 Switched LAN
- Enlarge the Switched LAN, how?



8-port switch

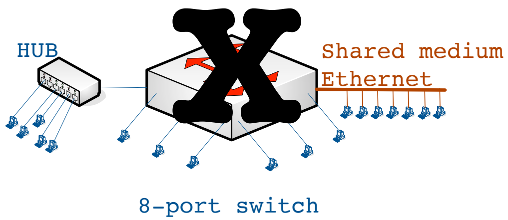
Enlarge the Switched LAN, how?

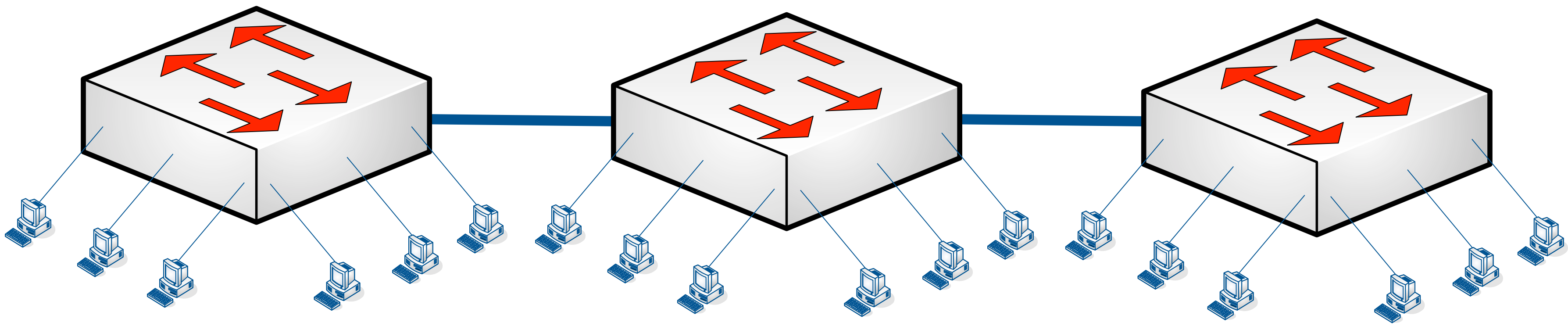
- Connect more hosts to each port
  - Recall: 1 port = 1 collision domain
  - Connect several stations to each port
  - Complex; damages performance; unreliable



Enlarge the Switched LAN, how?

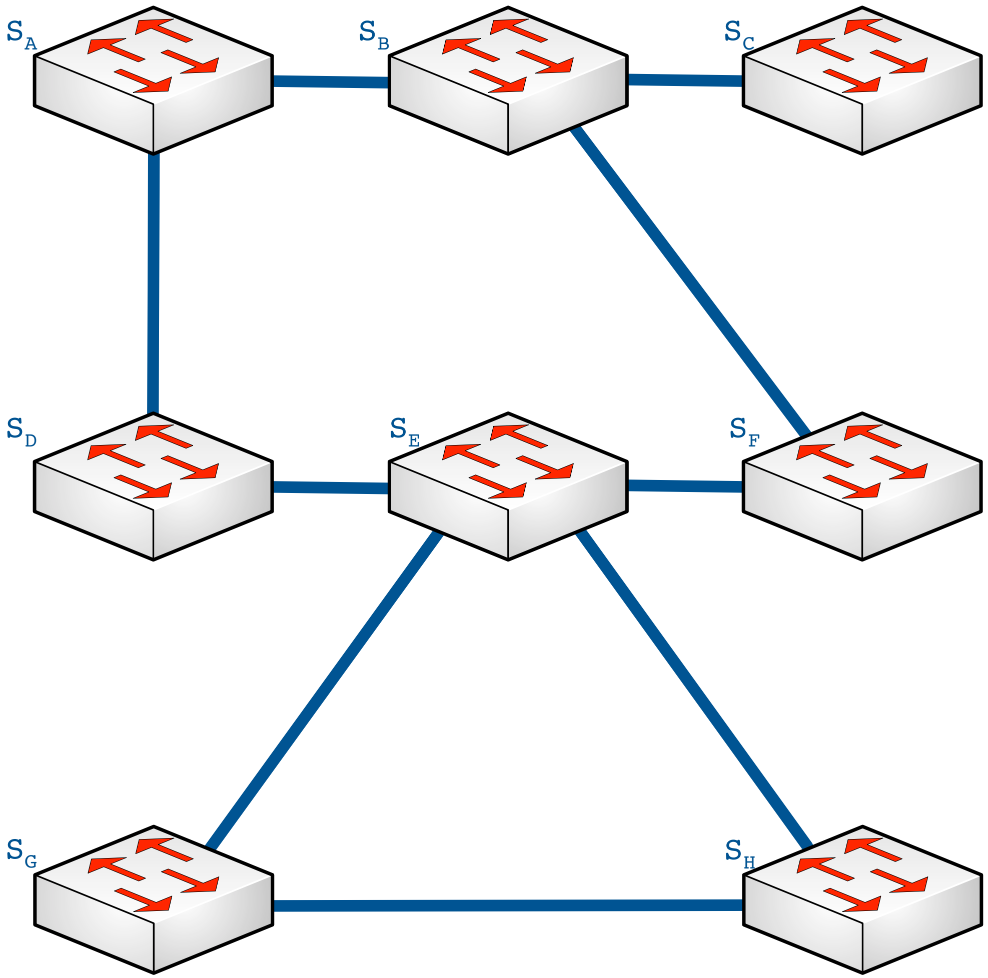
- Connect additional switches
- Preserves performance if well designed
- Even, create redundant connections
  - In case of link or cable faults





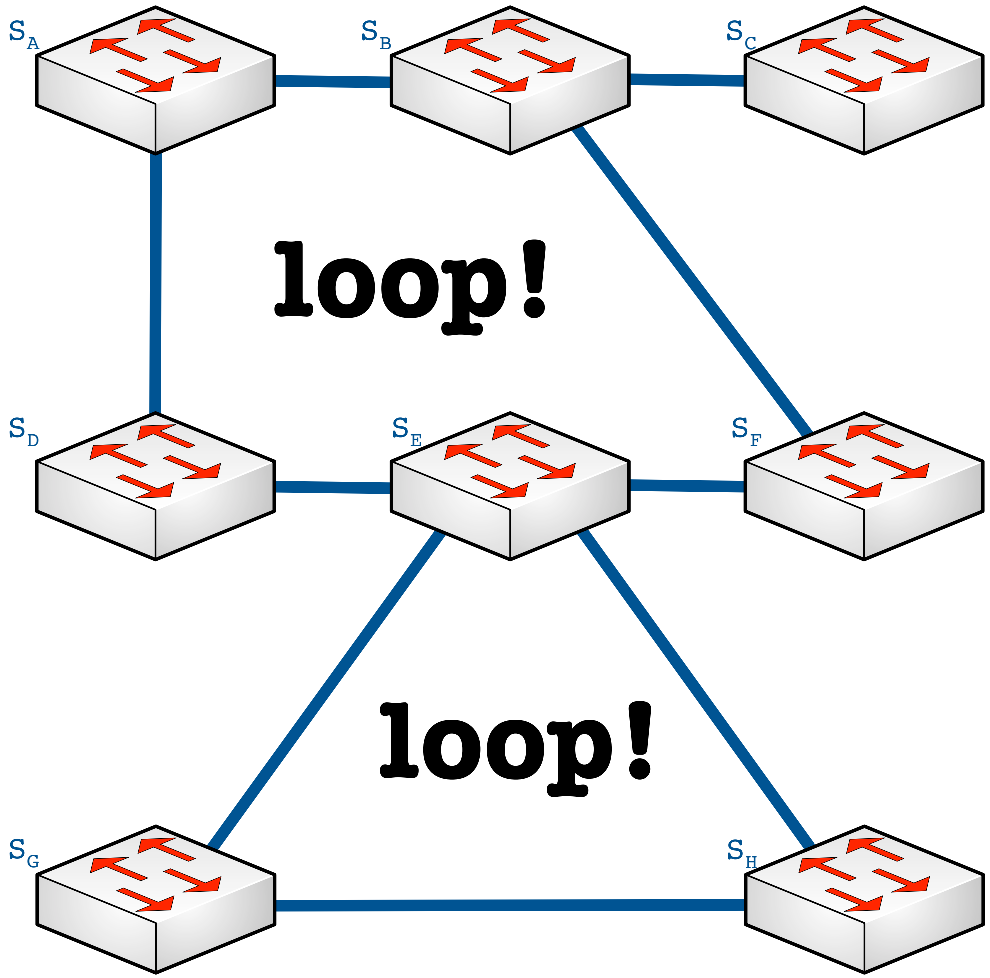
Enlarge the Switched LAN, how?

- Connect additional switches
- Preserves performance while increasing utilization
- Even, create redundant connections
  - Loops
  - Good for faults



Loops help with link faults

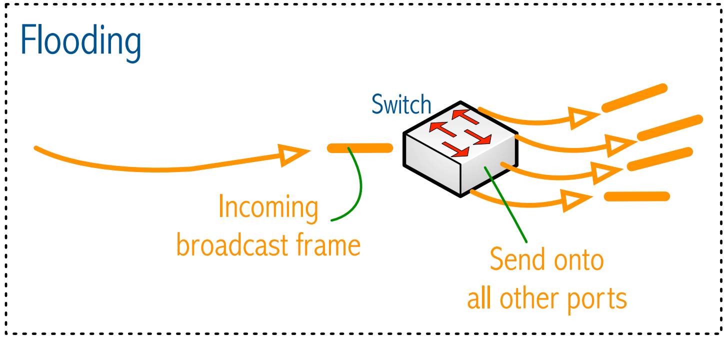
- Redundant paths
- Good when faults happen
- Drawbacks
  - Loops, when not properly managed, cause broadcast storms
  - Frames proliferate out of control
  - The whole switched LAN becomes useless
  - Hundreds of stations will not be able to communicate

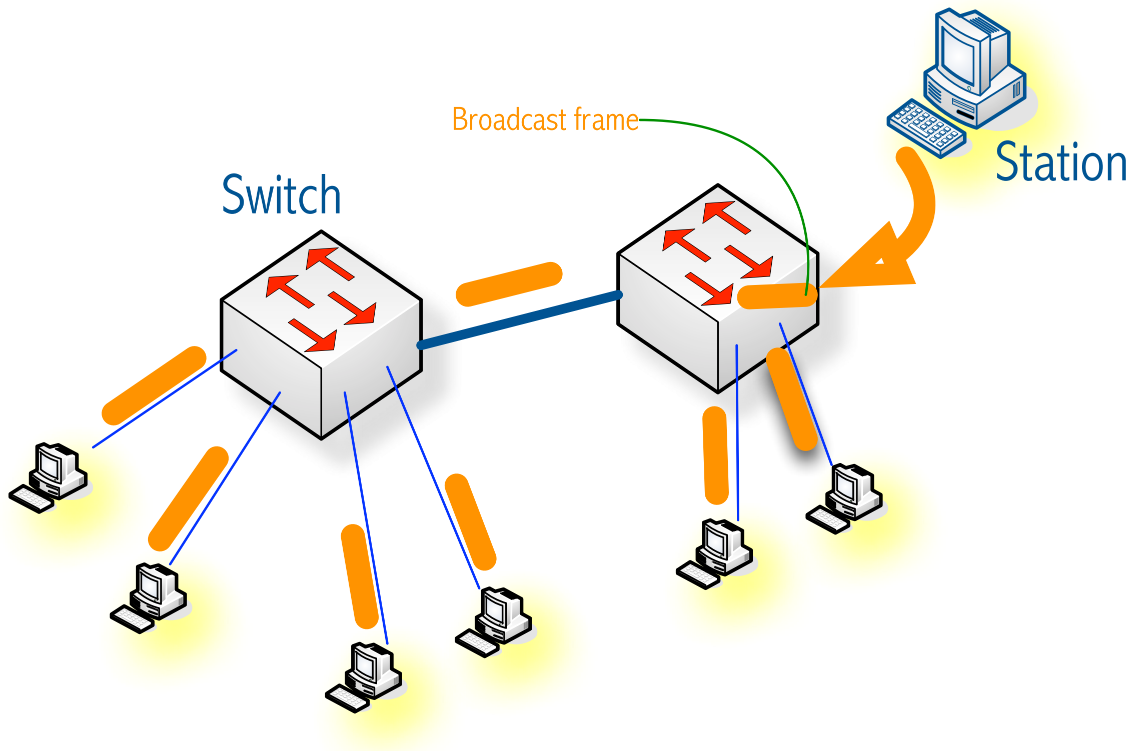


Review: LAN Switching Algorithm

- A switch floods every frame
  - Which Destination MAC is the broadcast address
  - Or, which destination address is not found in its FIB

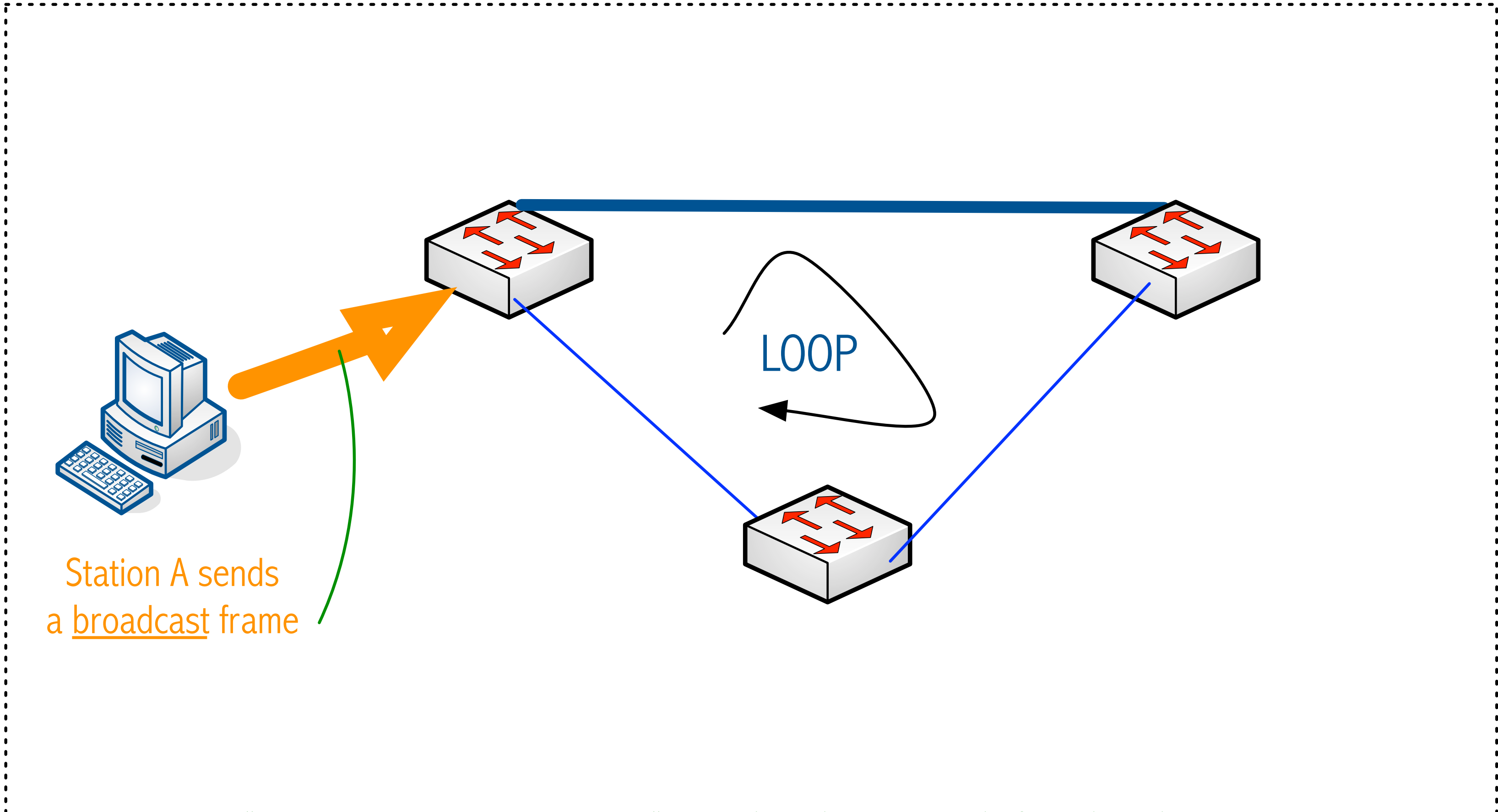
Essential!





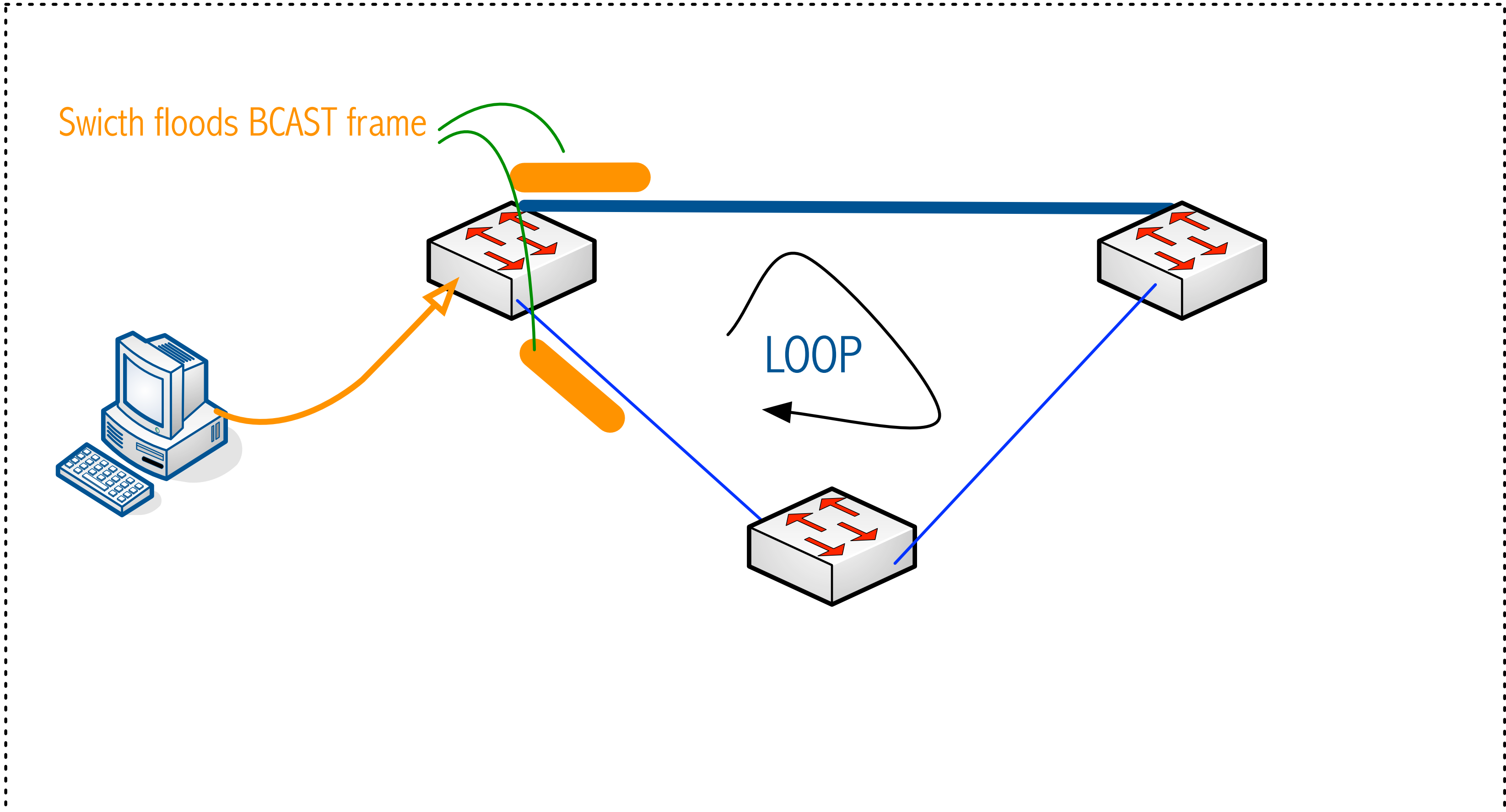
Switched LANs, broadcast and loops

- Station A sends a BROADCAST frame



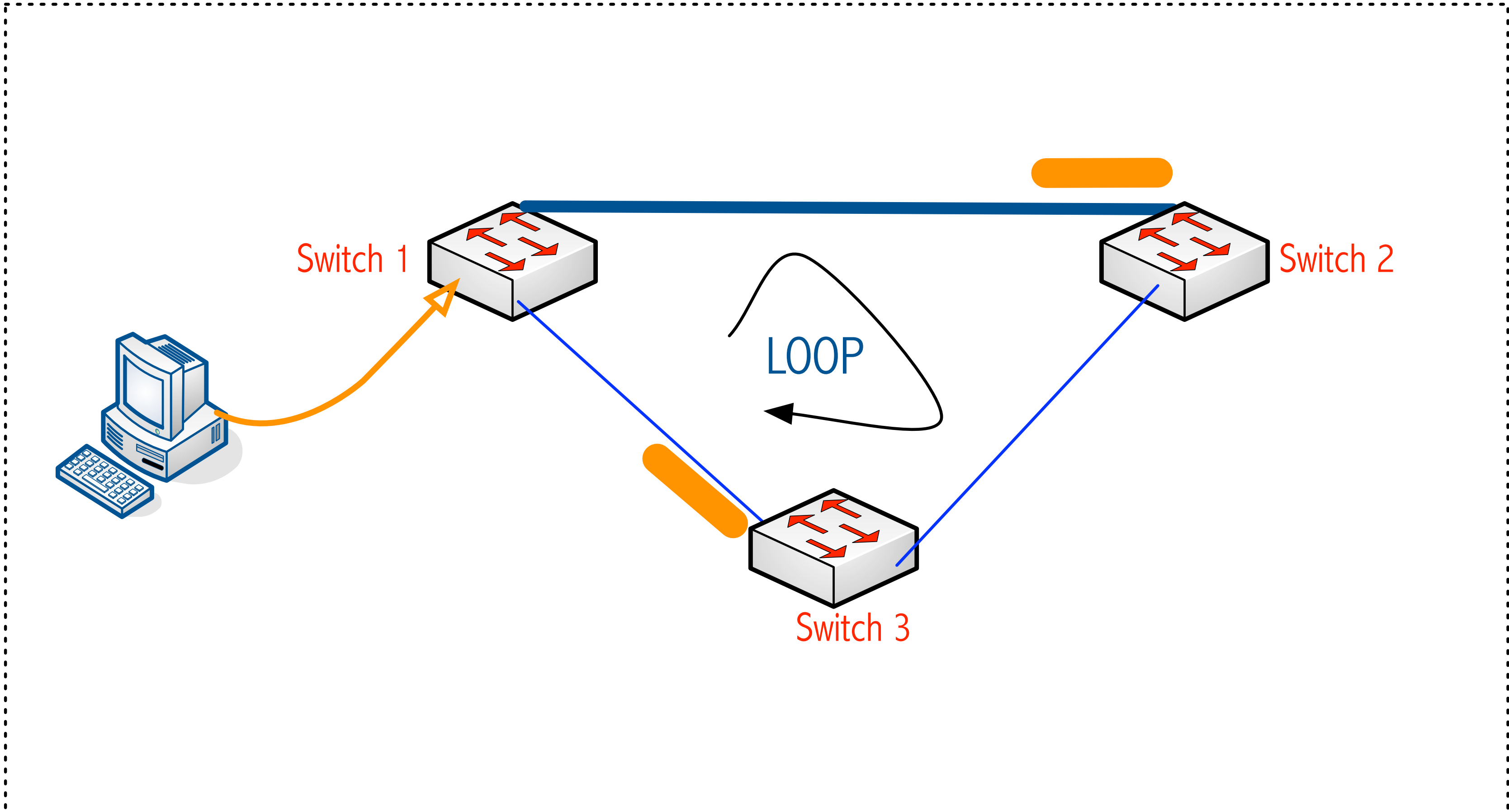
Switched LANs, broadcast and loops

- BCAST frame ingresses in switch
  - Switch will <u>flood</u> it: send it over all ports except the port over which it was received



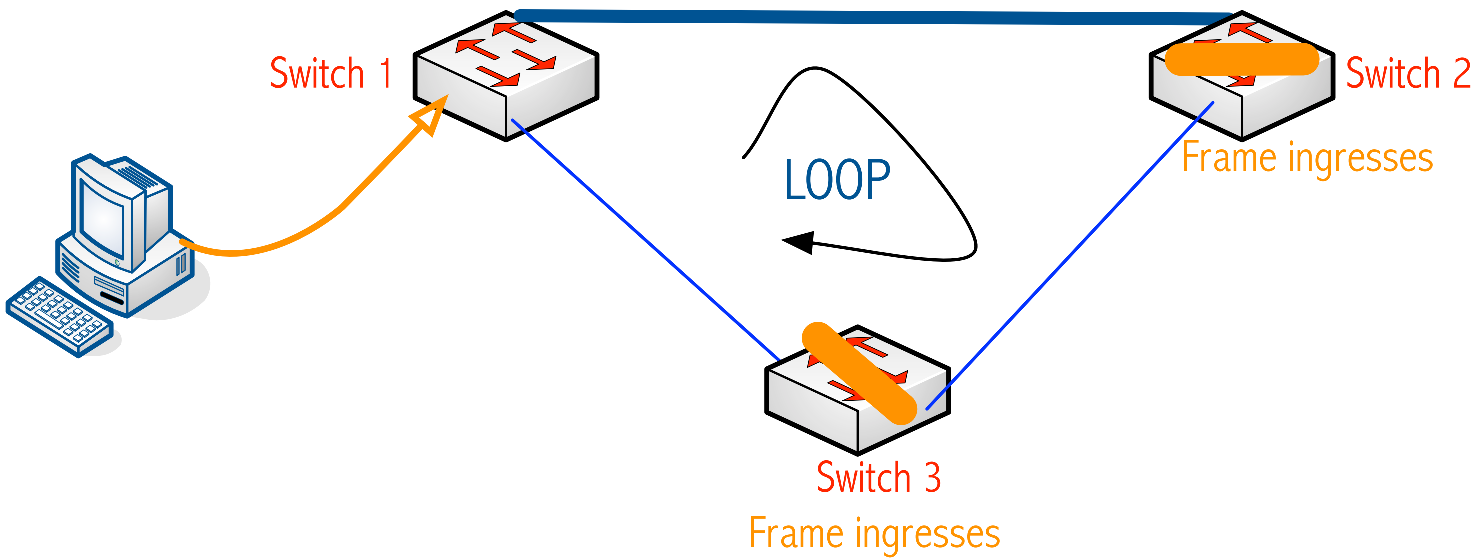
Switched LANs, broadcast and loops

- BCAST frame is received by Switch 2 and Switch 3



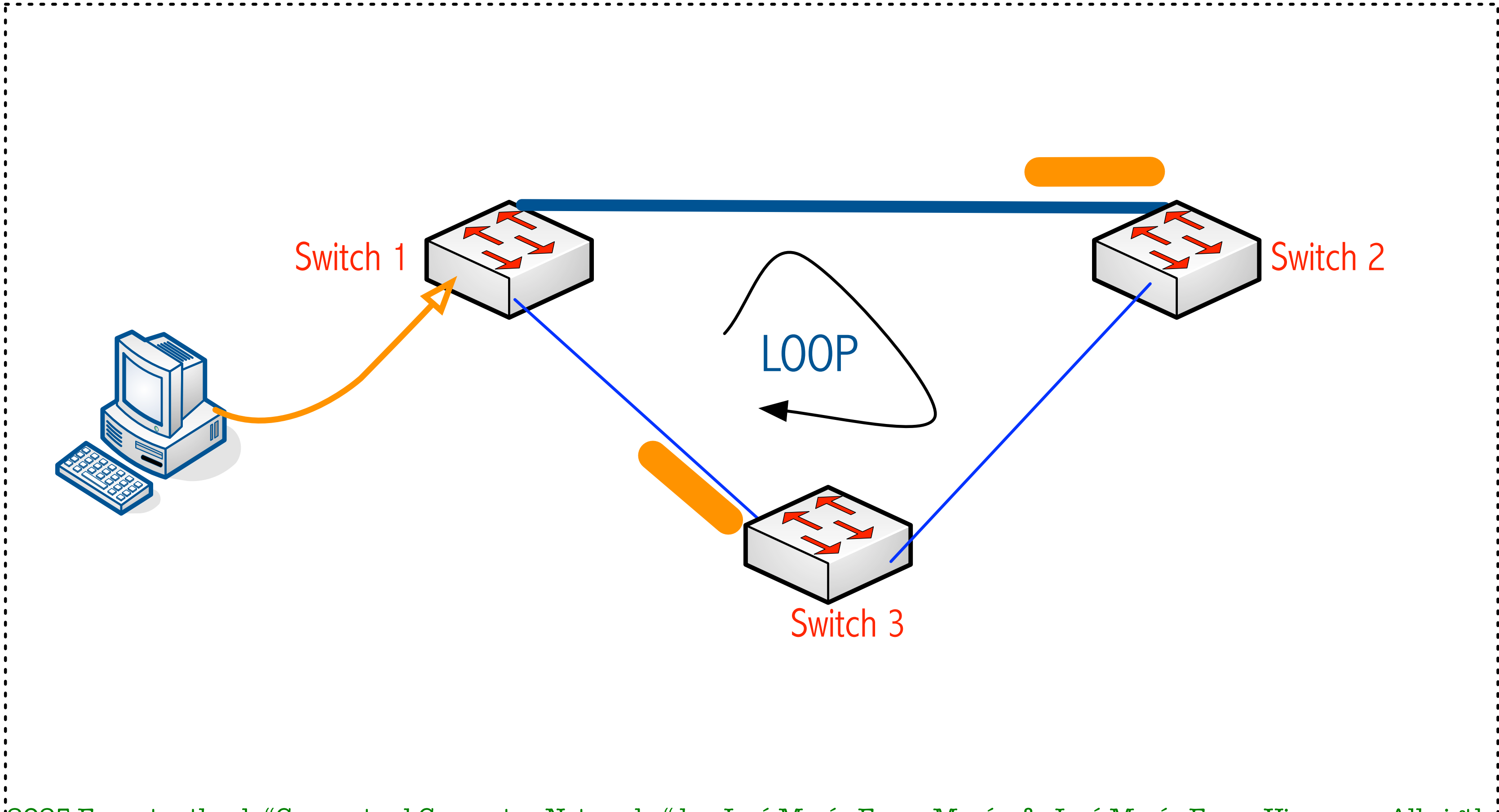
Switched LANs, broadcast and loops

- BCAST frame ingresses into Switch 2 and Switch 3



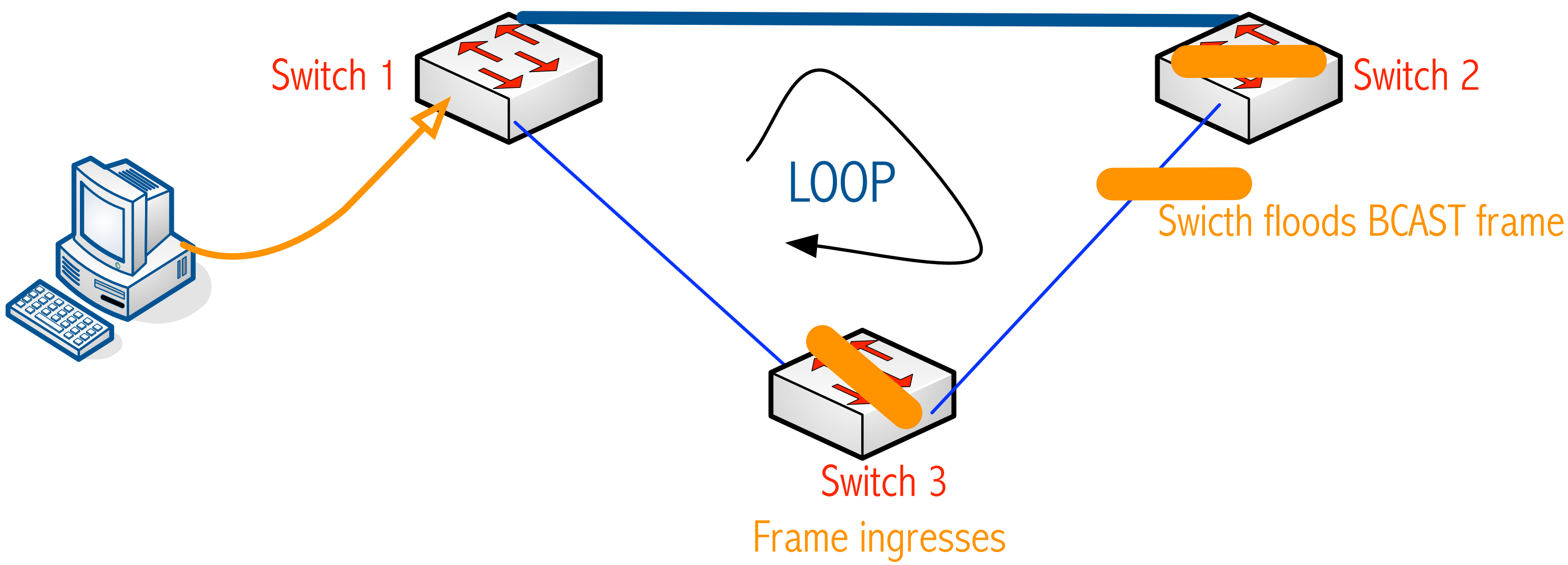
Switched LANs, broadcast and loops

- BCAST frame ingresses into Switch 2 and Switch 3



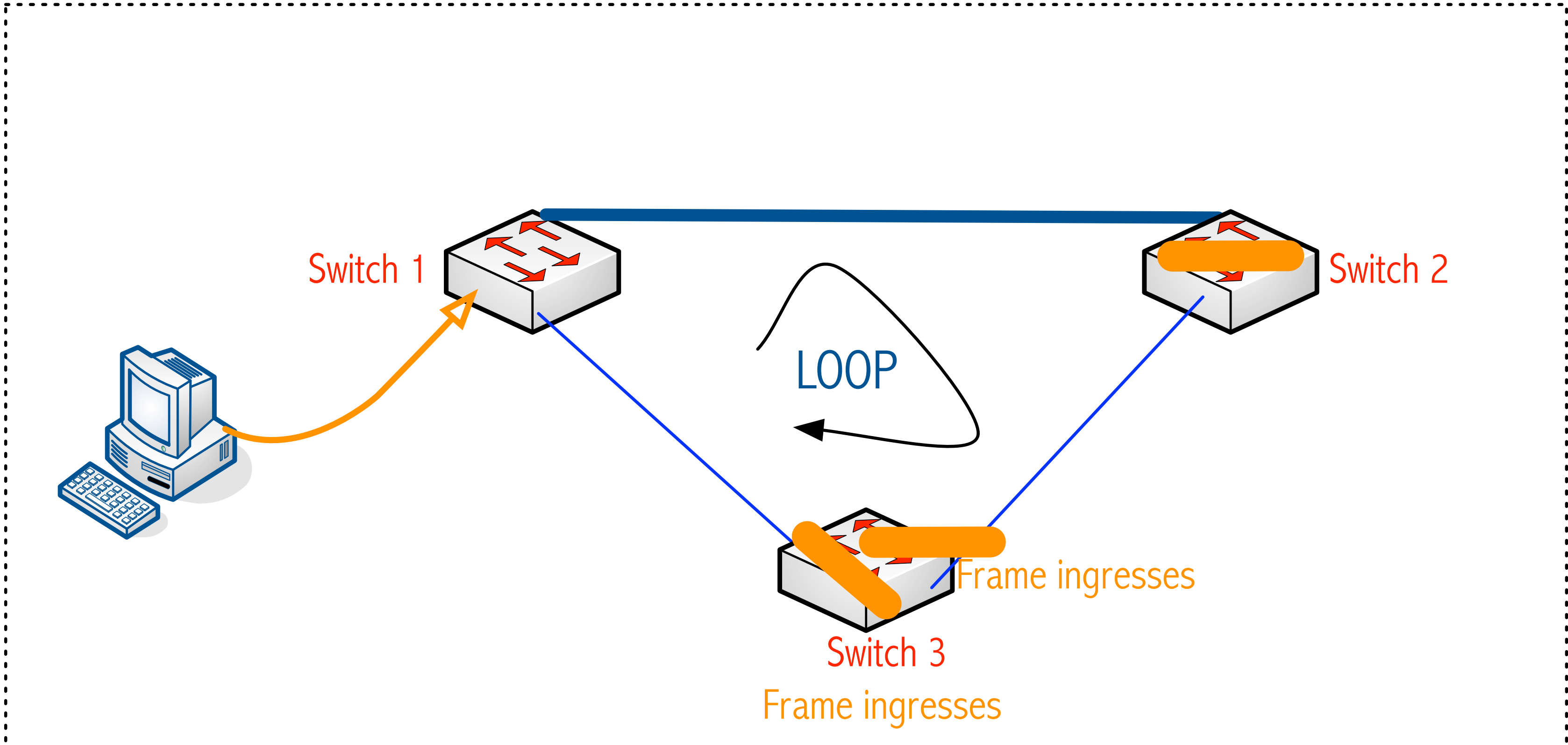
Switched LANs, broadcast and loops

- Switch 2 floods BCAST frame



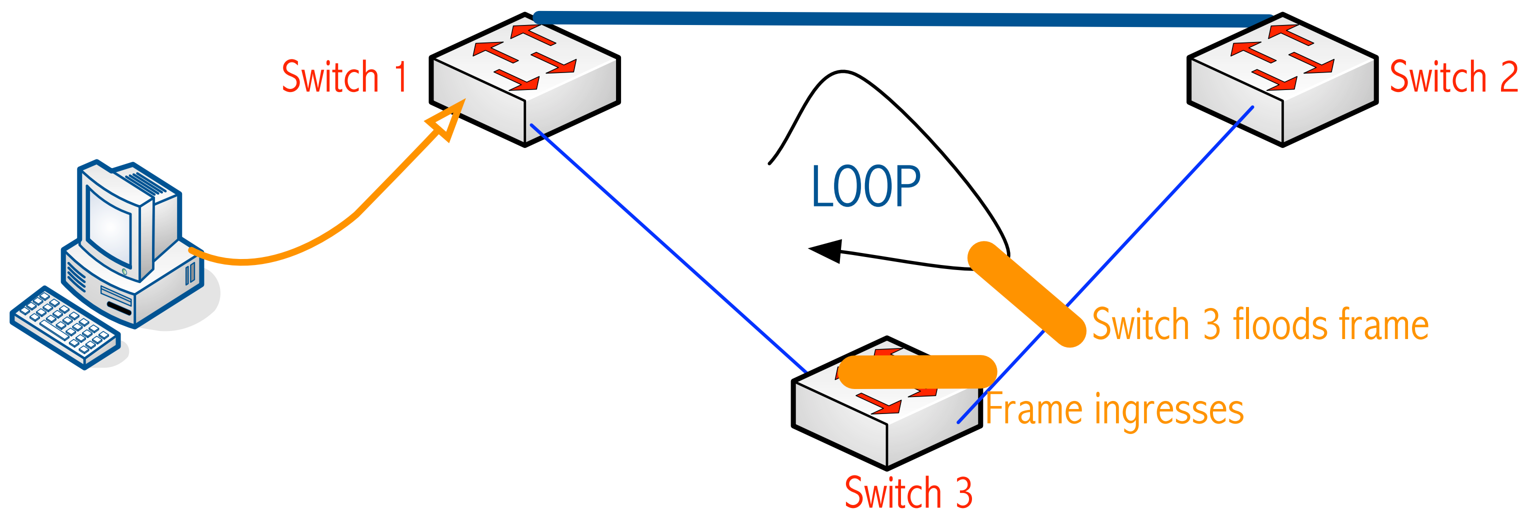
Switched LANs, broadcast and loops

- Switch 3 floods BCAST frame

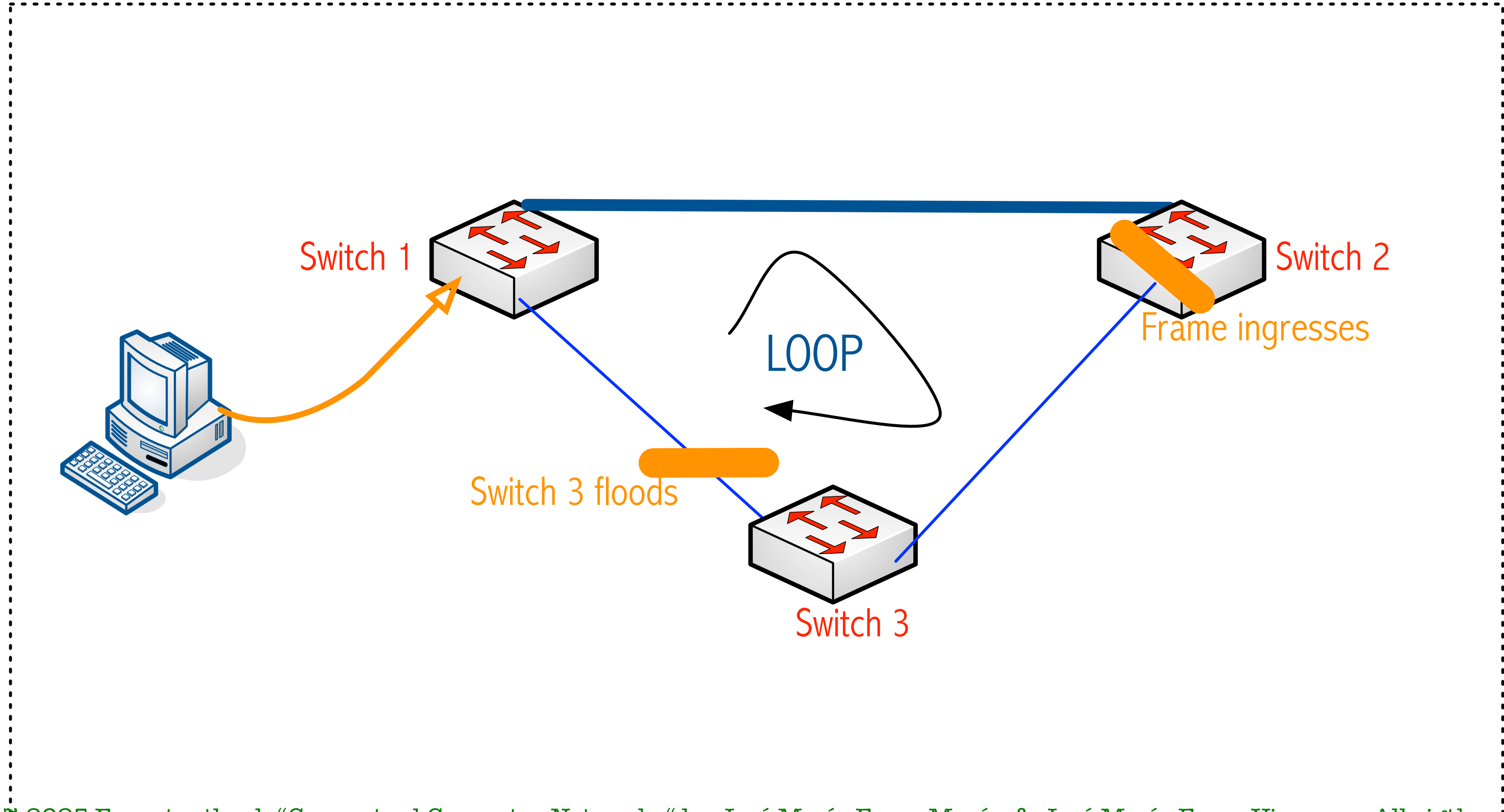


Switched LANs, broadcast and loops

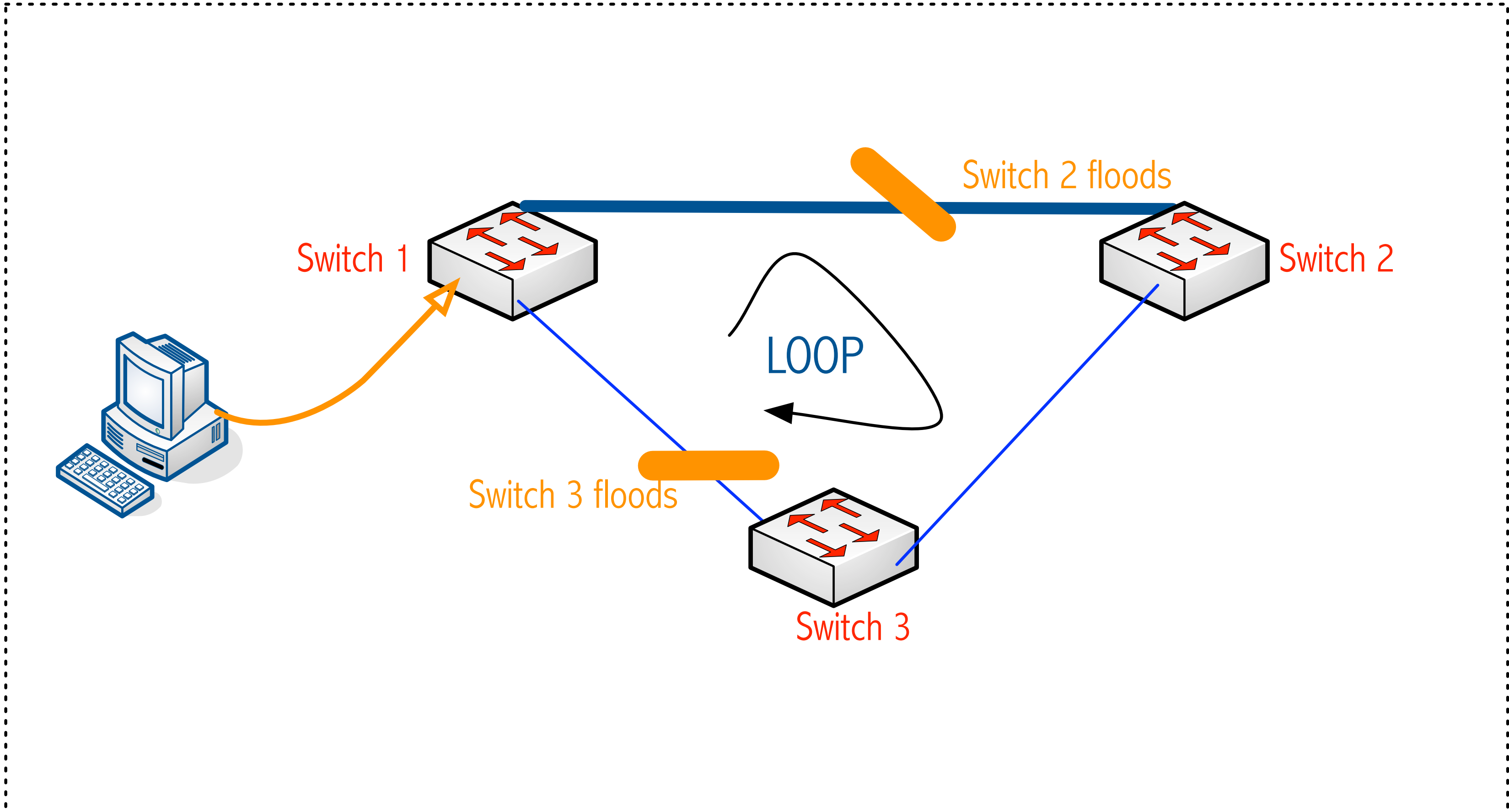
- New copy of frame goes towards Switch 2
- Switch 3 floods the new received copy of frame



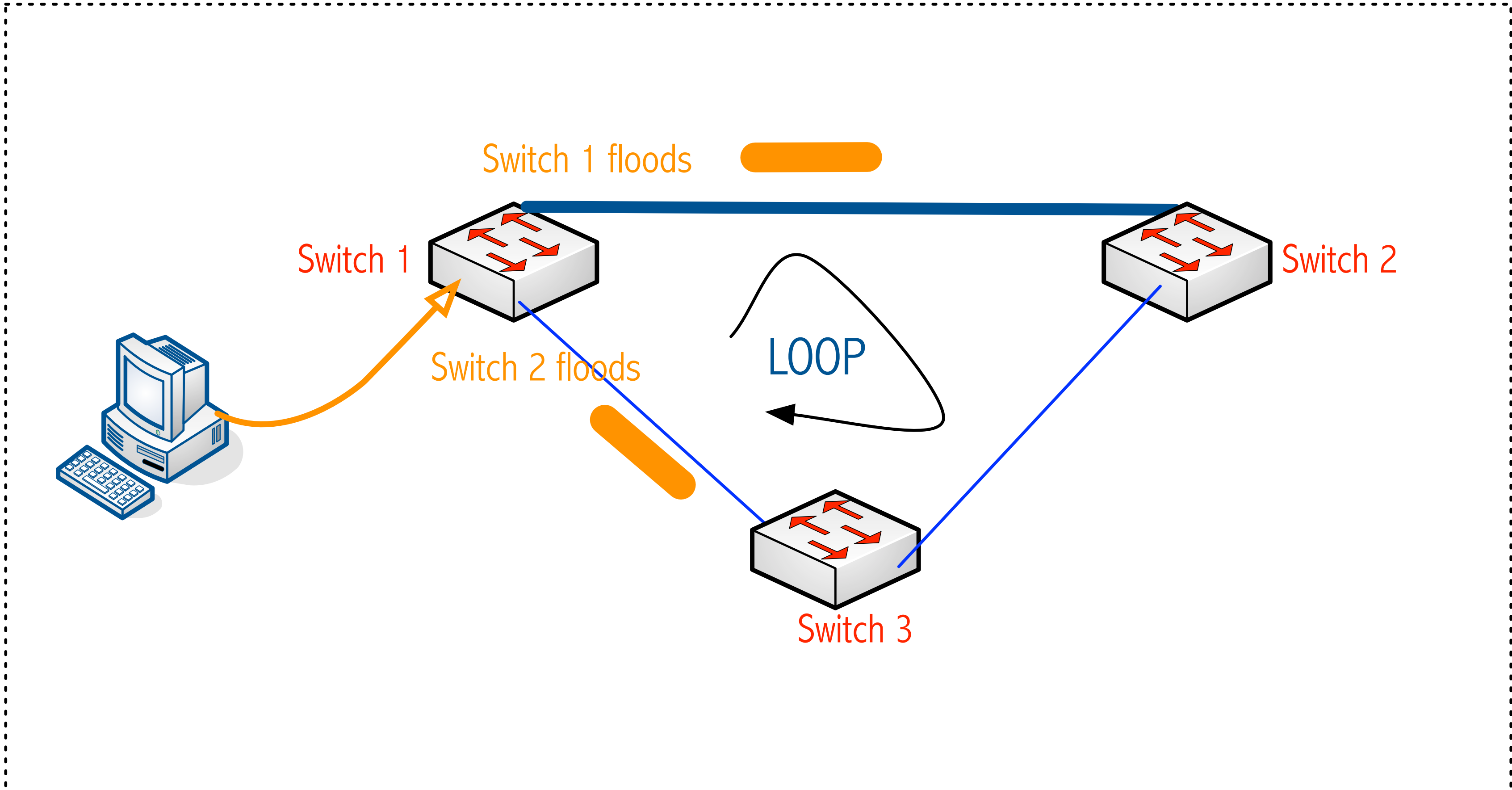
Switched LANs, broadcast and loops



Switched LANs, broadcast and loops

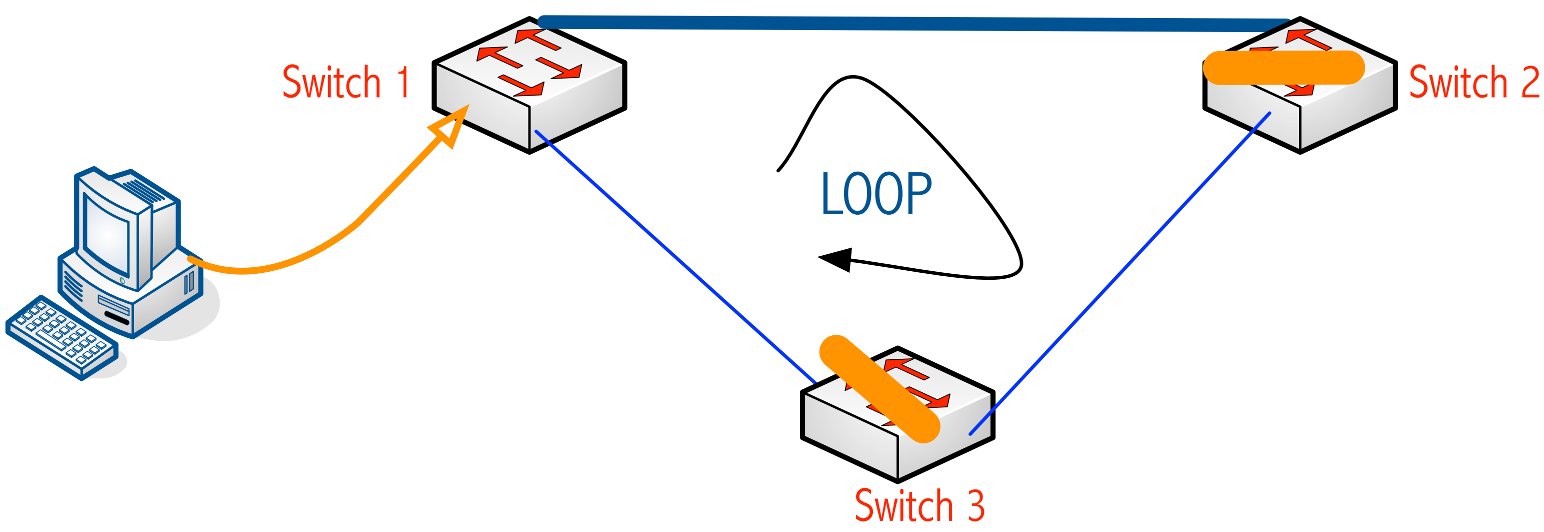


Switched LANs, broadcast and loops



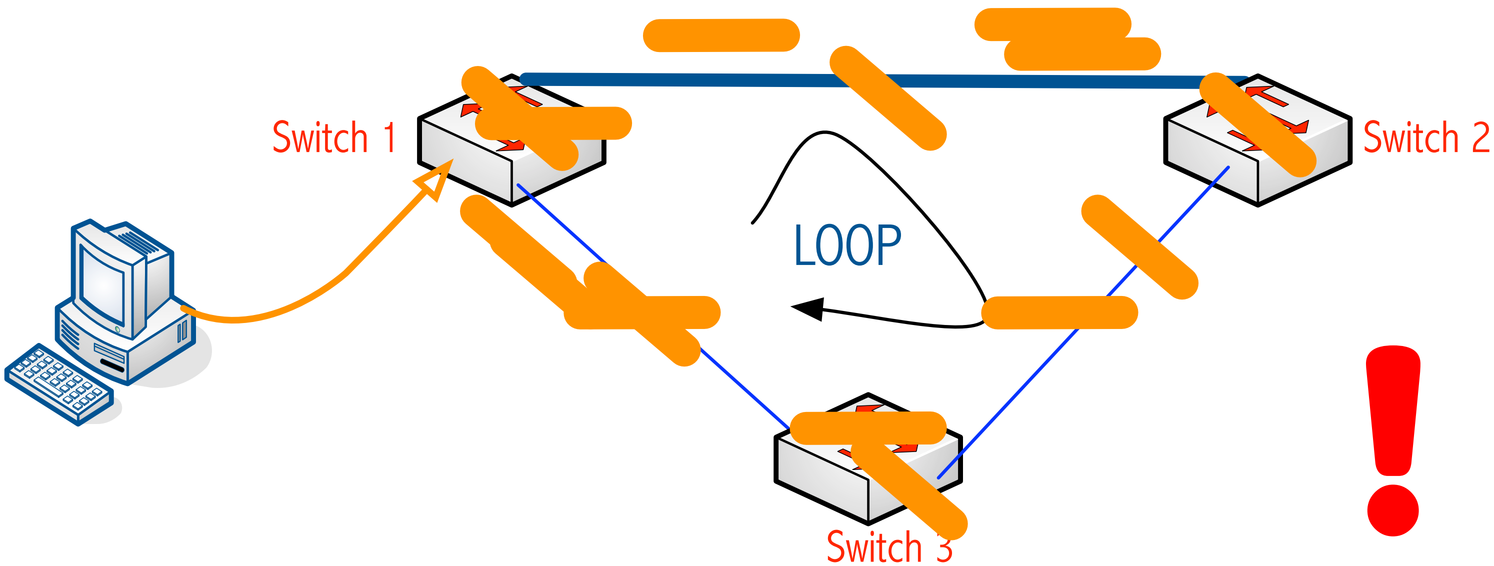
Switched LANs, broadcast and loops

- Frame keeps proliferating
- Endless cycle of receive-copy-send-receive-copy-send
- <u>Millions of copies of frame competing with just a few new, legitimate frames sent by stations</u>



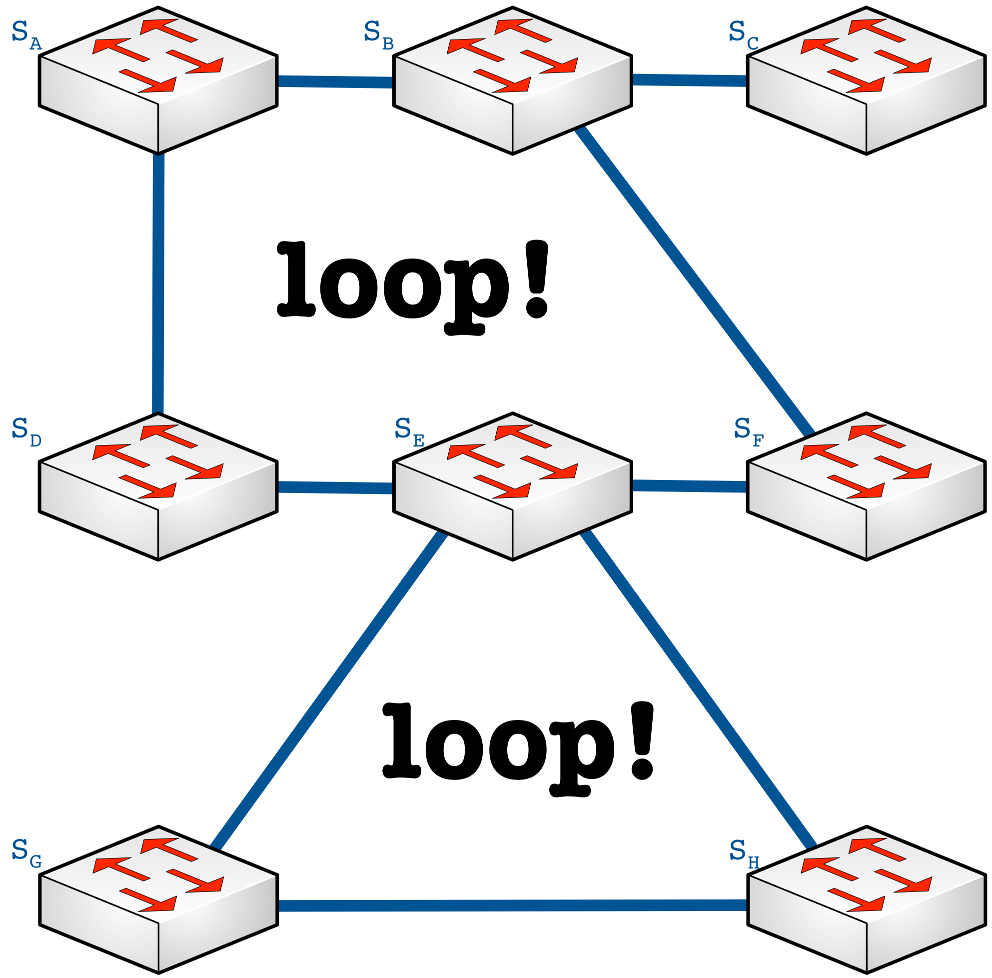
Switched LANs, broadcast and loops

- This process never ends and takes up all the network bandwidth!
- Loops provide redundant network paths in case of failures
- BUT, LOOPS CAUSE BROADCAST STORMS!!!



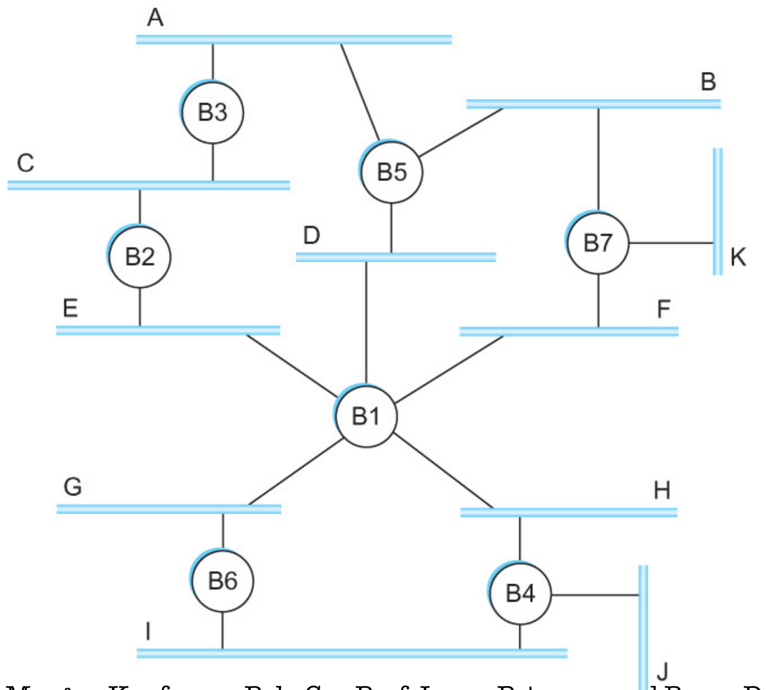
Can flooding live with loops?

- Great having loops and avoiding frame proliferation
- It's possible with:
  - Spanning Tree Protocol
  - STP
  - IEEE 802.1D



Switched Extended LANs

- Example:
  - What is this? A network: <u>*ONE*</u> network
  - B1, B2 … = Bridges
    - Remember: basically a switch
  - A, B, C: LAN segments (Collision domains)
    - Several collision domains
    - One broadcast domain: *The extended LAN*
  - Where are the end-nodes (hosts)?
    - Each end-node is connected to a segment



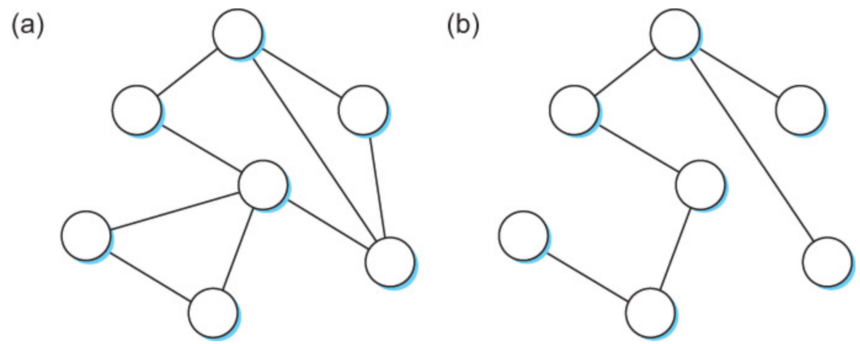
Spanning Tree Protocol (STP)

- Spanning Tree Algorithm is a distributed algorithm
- STP is based on it

Spanning Tree Algorithm

- The extended LAN may contain loops

- A **spanning tree** is a sub-graph of a graph that covers all its vertices but contains no cycles
  - It offers the same –*abstract*- connectivity but with no cycles



Example of (a) a cyclic graph; (b) a corresponding spanning tree.

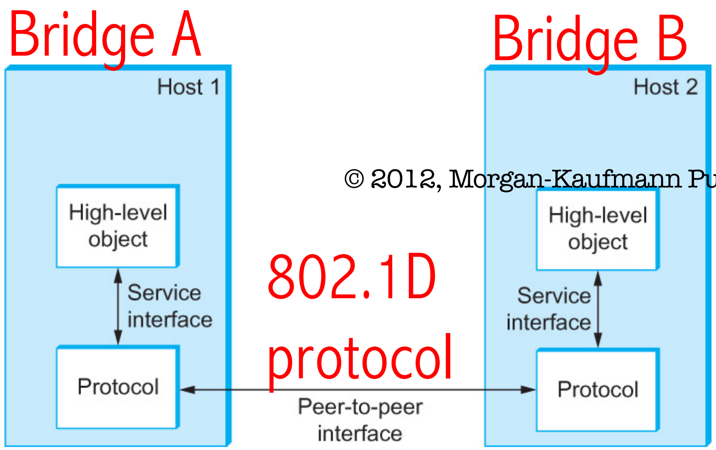
© 2012, Morgan-Kaufmann Pub. Co., Prof. Larry Peterson and Bruce Davie

Spanning Tree Protocol (STP)

- A protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN
  - STP is based on the Spanning Tree Algorithm

- The IEEE 802.1D specification for LAN bridges is based on this algorithm

- Each bridge decides the ports over which it is and is not willing to forward frames
  - By removing ports from the topology the extended LAN is reduced to an acyclic tree
  - It is possible that an entire bridge will not participate in forwarding frames



Spanning Tree Protocol (STP)

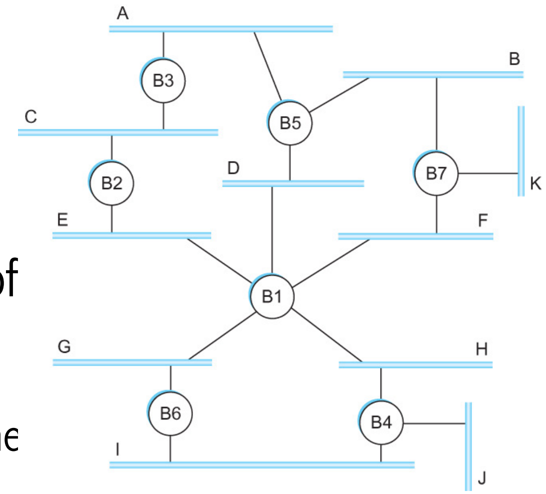
- Spanning Tree is executed in a distributed way (It's a distributed algorithm)
  - It is executed among a set of switches
  - The switches interchange STP messages (Look previous slide)

- The bridges are always ready to reconfigure themselves into a new spanning tree if some bridge or link fails

- Main idea
  - Each bridge selects the ports over which they will forward the frames

Spanning Tree *Algorithm*

- The distributed algorithm selects ports as follows:



© 2012, Morgan-Kaufmann Pub. Co., Prof. Larry Peterson and Bruce Davie

1. Each bridge has a unique identifier

   B1, B2, B3… The Bridge with the smallest id becomes root of

   The root bridge always forwards STP frames out over all of its ports

   Actual switches use as **ID the lowest MAC address** allocated to the ports

2. RP = Root Port:

   Each bridge computes the shortest path to the root and notes which of its ports is on this path

   This port is selected as the bridge's preferred path to the root

3. DBP = Designated Bridge Port:

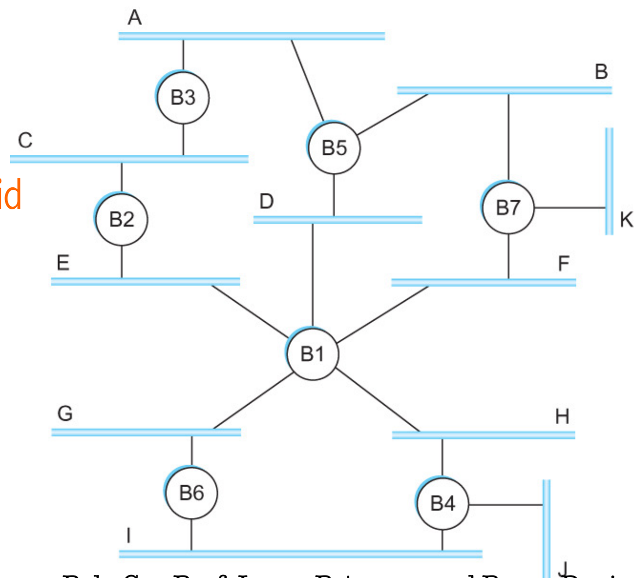
   All bridges connected to a given LAN elect a single DBP

   Responsible for forwarding frames toward the root bridge

Bridge's Root Port

- Each bridge has a **root port (RP)**
  - The closest port to the root
  - Used for communication with the root

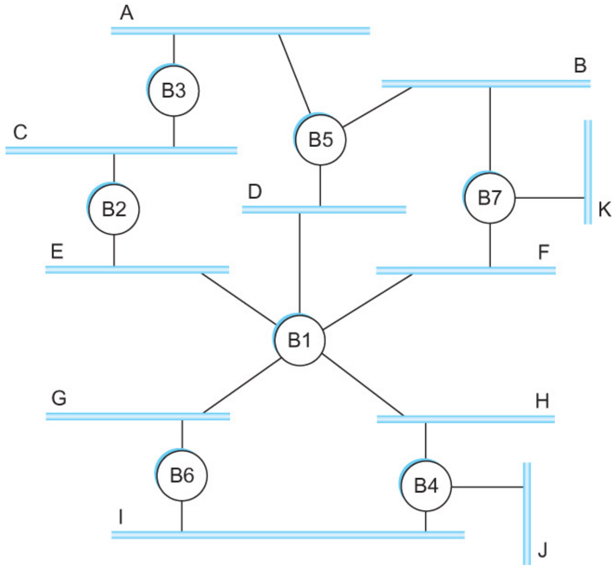
- If two or more ports are equally close to the root
  - *Break ties* by selecting the port with the smallest next-bridge id
  - If still equal cost, then *break ties* by choosing the port with lowest port id



- **Example: Which is B3's root port?**
  - B1 is root
  - Shortest distance from B3 to B1 (The root bridge)
    - Through A = 2
    - Through C = 2
    - Equal, then break ties:
      - A: Next bridge on least-cost path is B5
      - C: Next is B2 which has a lower ID than B5, <u>THEREFORE ROOT of B3 is its port C</u>

LAN's Designated Bridge Port

- Each LAN has a **Designated Bridge Port (DBP)**
  - It's the one that is closest to the root

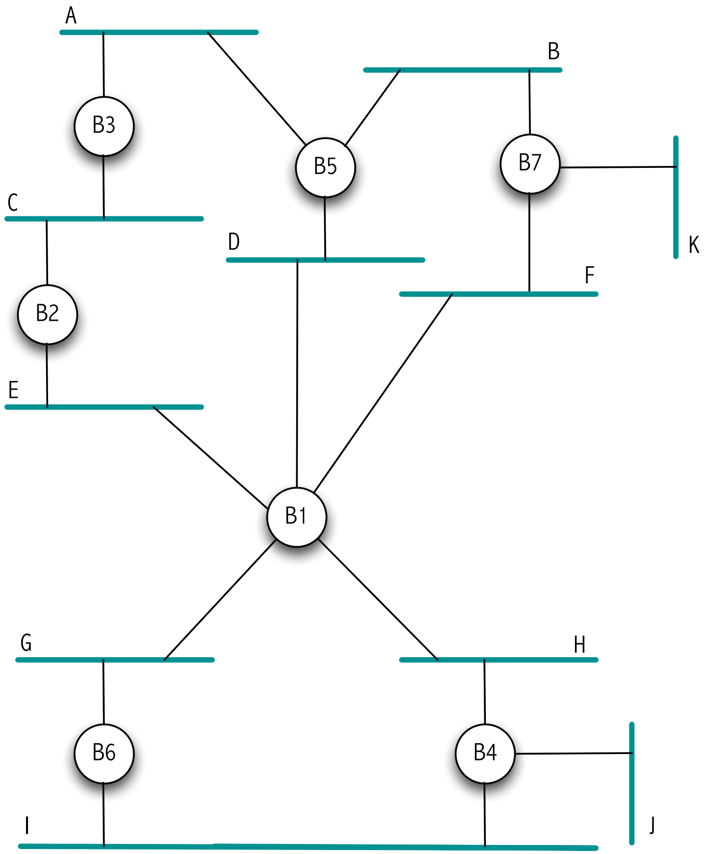


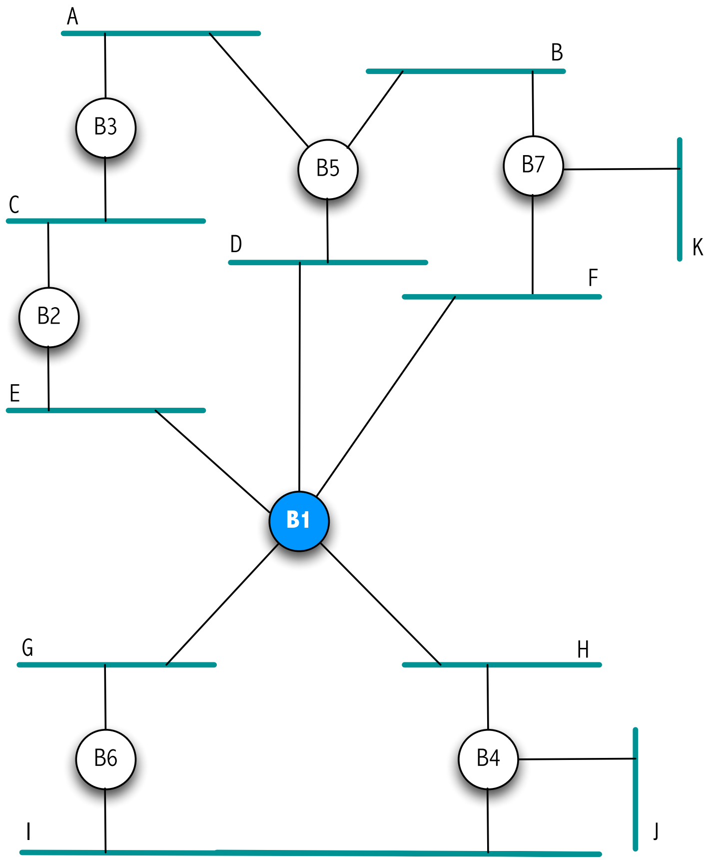
- If two or more bridges are equally close to the root,
  - *Break ties by* selecting the bridge with the smallest bridge id
  - If the bridge selected so far has two or more ports connected to a LAN, choose the port with lowest port id

- **Example: Which is the DBP of LAN B?**
  - Shortest distance from B → B1 (root) is 2 via B5 and via B7
  - Since B5 < B7, we select B5 as the *Designated Bridge on B,* specifically the port on the upper right of B5 is the Designated Bridge Port of LAN B

Exercise, STP

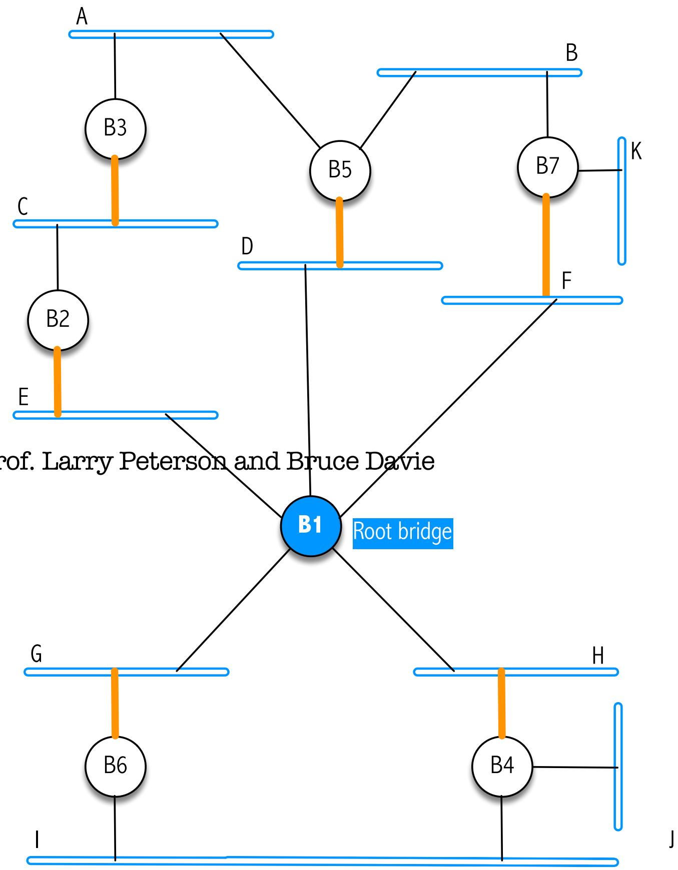
- **Example from textbook pg. 194 (Fig. 3.10, P&D Ed. 5): Extended LAN with loops**
- **Step 1: Root bridge**
  - B1 is the root bridge, the lowest numbered bridge





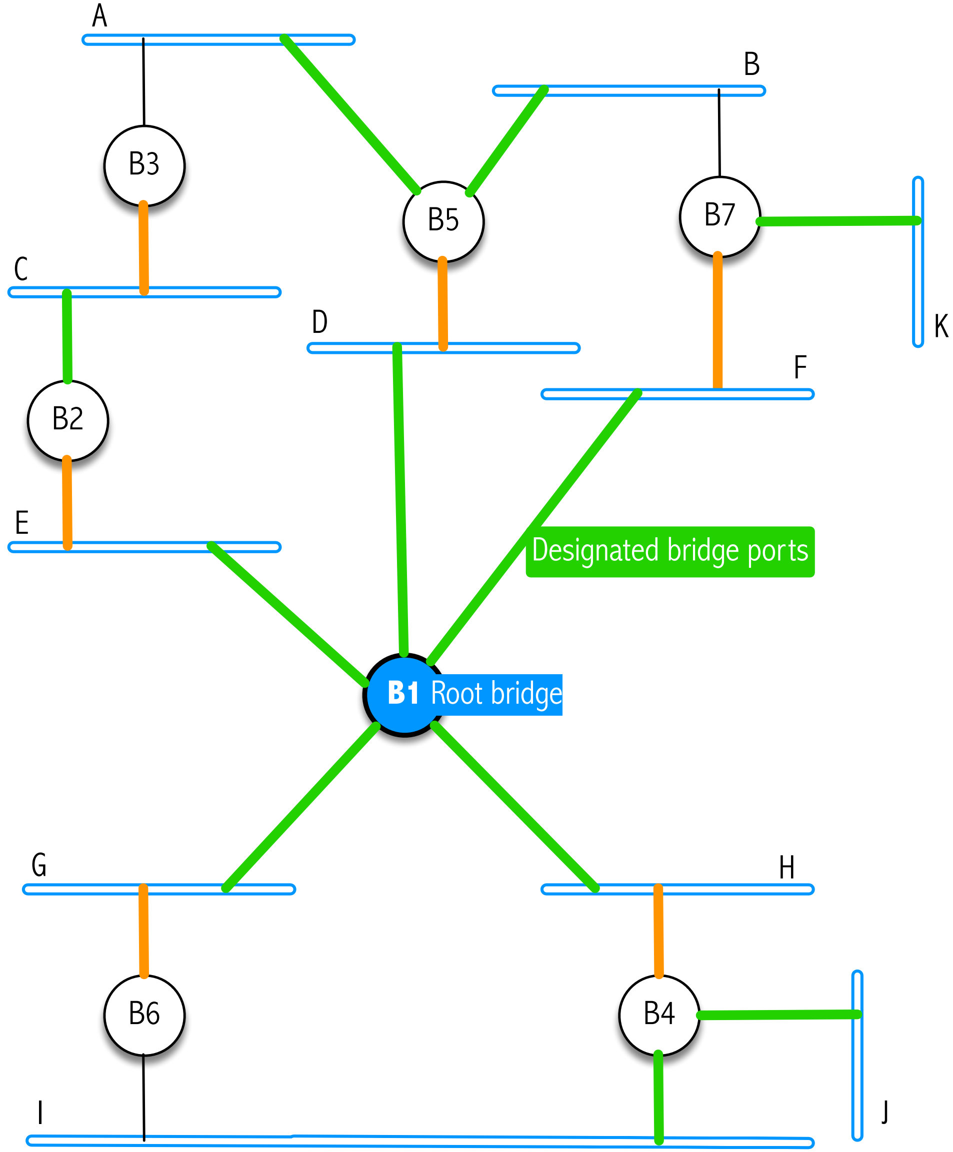
Exercise, STP

- **Textbook pg. 194 (Fig. 3.10; P&D Ed. 5): Extended LAN with loops**
- **Step 2: Root port (RP) of each bridge**
  - B3 least cost to root is 2 (Via A and via C)
    - Break ties by lower label of next bridge: Choose B2 since label is lower numbered, B2 < B5
  - B4 least cost to root is 1 (Via H)
  - Calculate the root port of each bridge



Exercise, STP

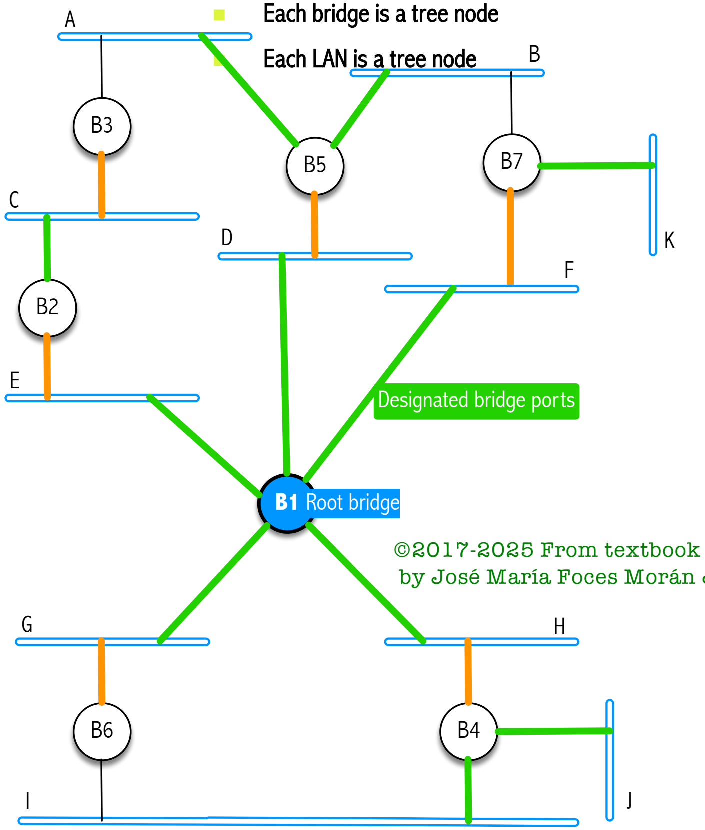
- **Textbook pg. 194 (Fig. 3.10; P&D Ed. 5): Extended LAN with loops**
- **Step 3: Designated Bridge at each LAN**
  - LAN A:
    - Cost to root via B3 = 3
    - Cost to root via B5 = 2
      - Choose bridge that is on the least cost path: B5
  - LAN J:
    - Connected to B4 only: Designated bridge is B4
  - LAN B:
    - Cost to root via B5 = 2
    - Cost to root via B7 = 2
      - Break ties by next bridge label, choose lower: B5 < B7, therefore designated bridge at LAN B is B5

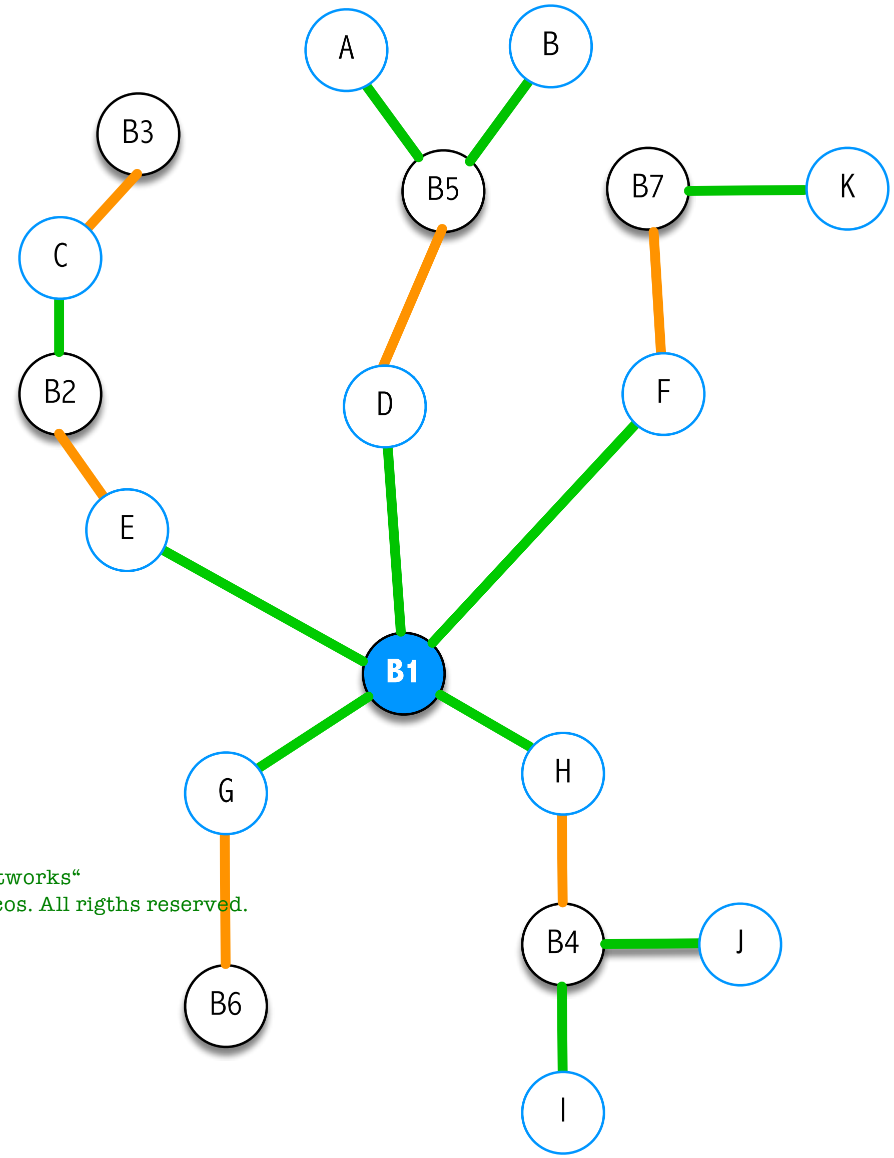


Exercise, STP

Step 4: Obtain the spanning tree

- All ports root or designated port result into active ports
  - Ports not RP nor DBP: disabled
- Spanning tree nodes:
  - Each bridge is a tree node
  - Each LAN is a tree node

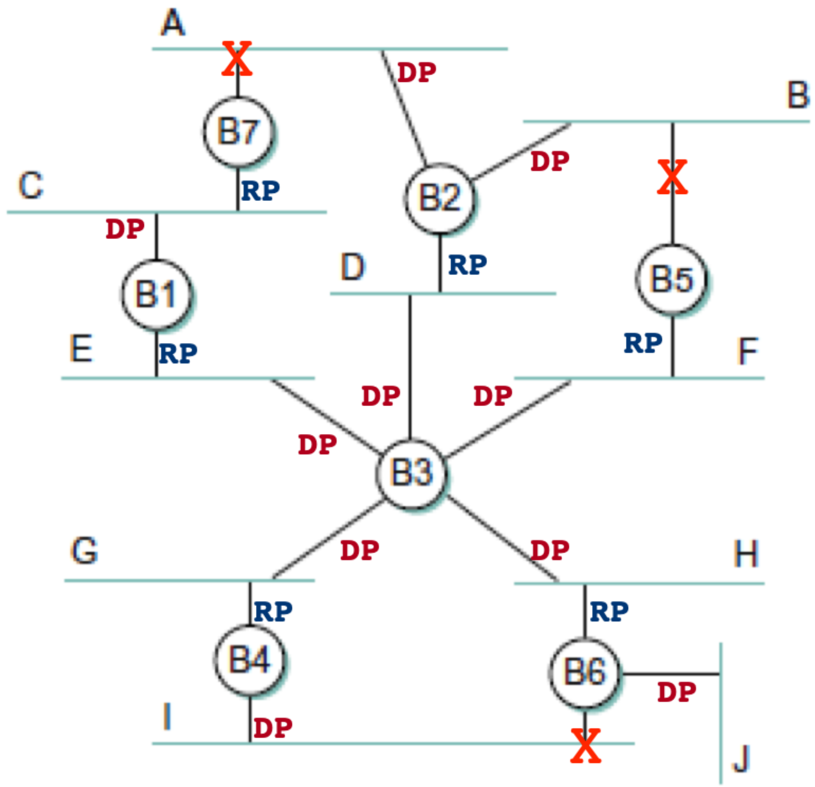




Homework exercises

Peterson and Davie's Exercise 13, Ch. 3.

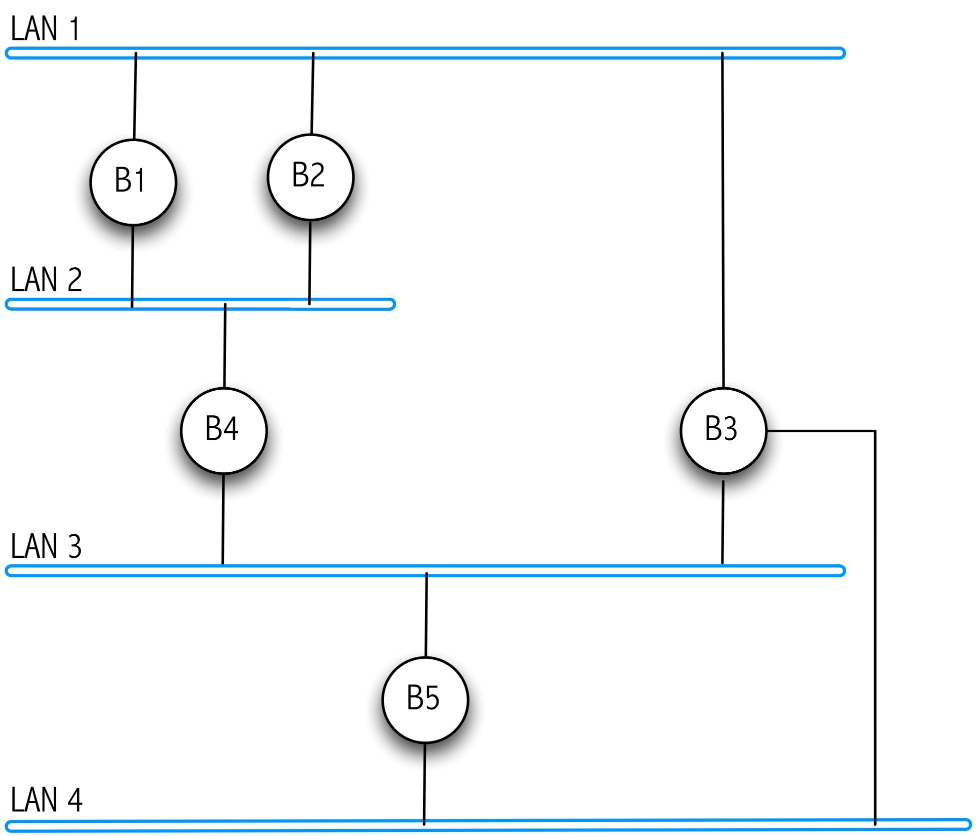
13. Given the extended LAN shown in Figure 3.48, indicate which ports are not selected by the spanning tree algorithm.



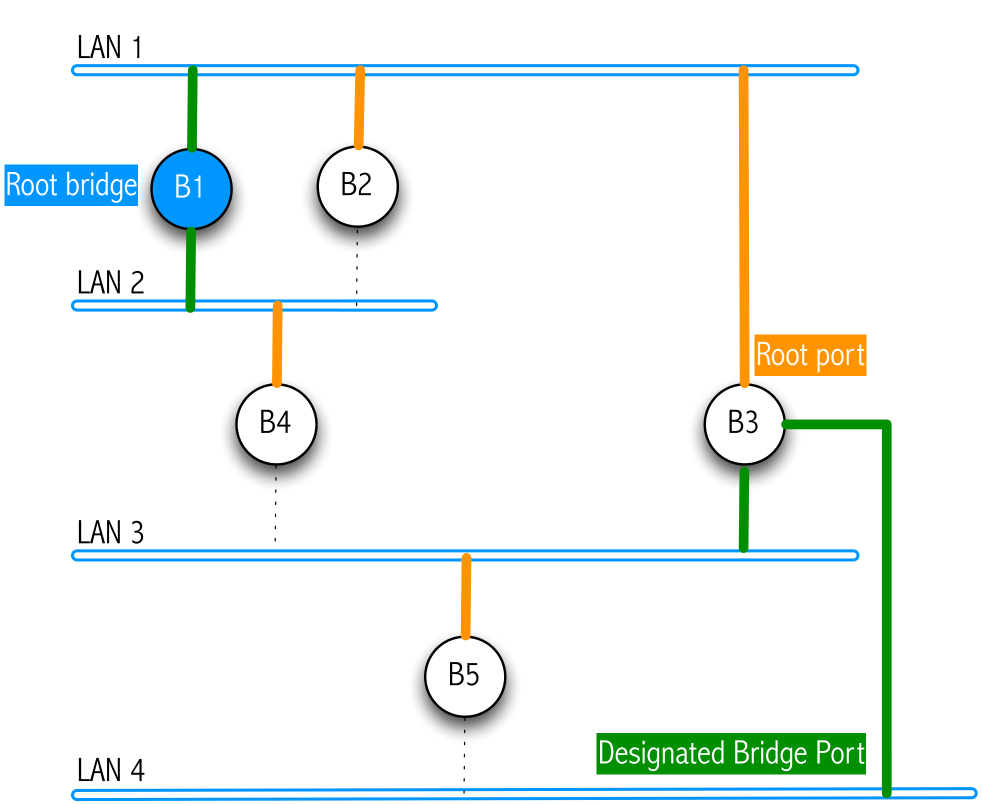
Spanning Tree Algorithm, example

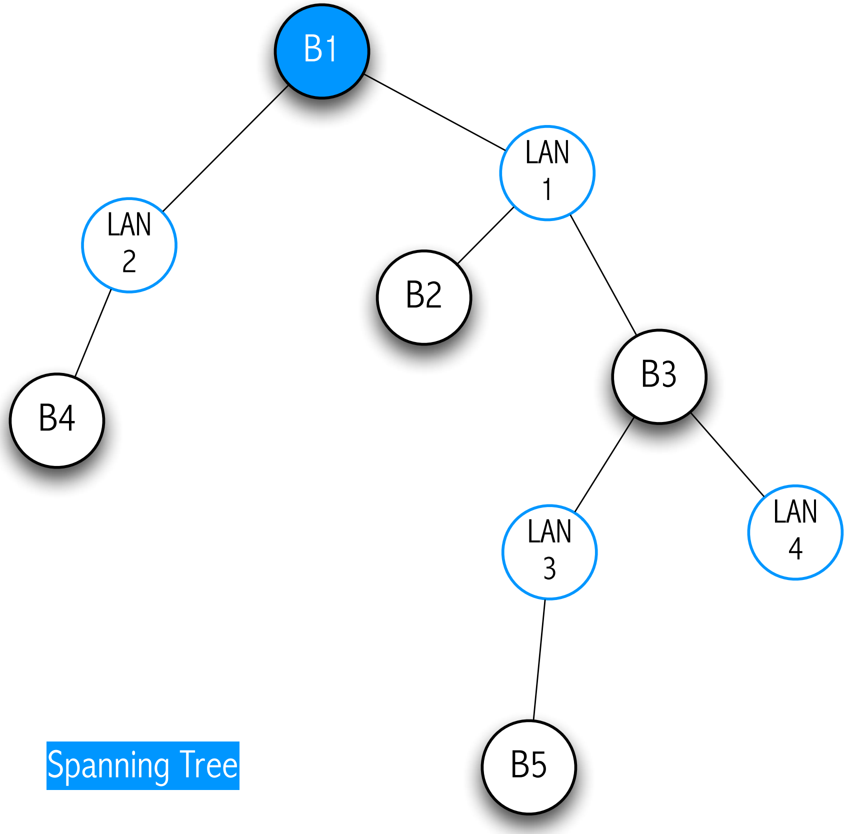
- Obtain the Spanning Tree to the Extended Lan
  - 1. Root bridge
  - 2. RP
  - 3. DBP
  - ST

Extended Lan



Spanning Tree Algorithm, example





Study of STP

STP and other exercises

Weekly Questioannaires and Solutions from past years (For reference only))

- Questionnaire 1
- *Questionnaire 1 Reference Solution*
- Questionnaire 2
- *Questionnaire 2 Reference Solution*
- Questionnaire 3
- Questionnaire 3 Reference Solution
- Questionnaire 4 (Shannon etc.)
- Questionnaire 5 (Datalink-Ethernet, etc.)

- Questionnaire 5 (Datalink-Ethernet/2024)
- Ref. solution to Questionnaire no. 4 (Shannon, etc.)
- Solution to Questionnaire 5 (Datalink-Ethernet, etc.)
- Questionnaire about Error detection AND SOLUTION
- Questionnaire about Exp. Backoff and Switches AND SOLUTION
- Questionnaire about the Spanning Tree Protocol and Intro to IP AND SOLUTION
- Questionnaire about IP forwarding and IP fragmentation
- Questionnaire about IP numbering and IP routing

- http://paloalto.unileon.es/cn/Q/CN-Q-STP-IntroIP-SOL.pdf
- http://paloalto.unileon.es/cn/notes/CN-ExRefSol2013.pdf

The end