Reti di Calcolatori 2019/20
Chapter 3 - Internetworking

Prof. Marino Miculan

Problems

- In Chapter 2 we saw how to connect one node to another, or to an existing network. How do we build networks of global scale?
- How do we interconnect different types of networks to build a large global network?
  ↳ perché se usiamo i cavi, con Ethernet, la distanza è quache distanza di metri. come scaliamo per tutto ie mondo?
- Chapter Outline
  - Switching and Bridging
  - Basic Internetworking (IP)
  - Routing

Chapter Goal

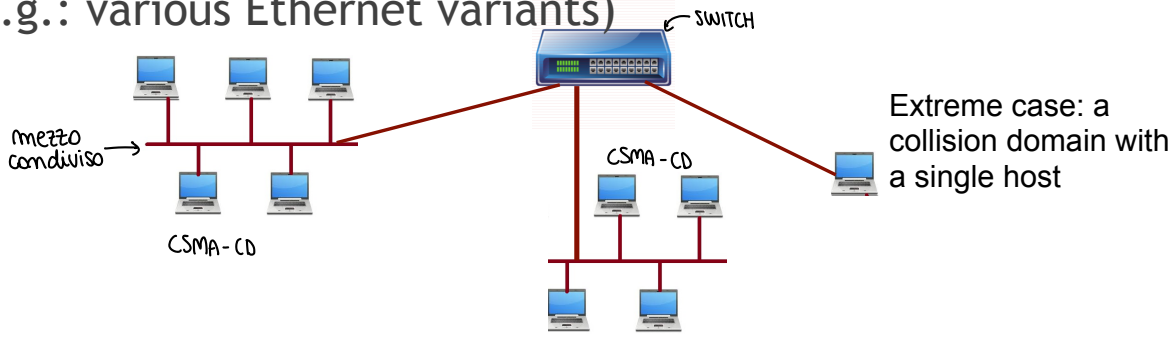
- Understanding the functions of **switches, bridges** and **routers**
  - Connection between different networks
- Discussing **Internet Protocol** (IP) for interconnecting networks
  - Abstraction over the kind of networks
- Understanding the concept of **routing**
  - in an internet, many paths are possible…how to choose, how to follow?

Switching and Forwarding

- Store-and-Forward Switches
- Bridges and Extended LANs
- Cell Switching
- Segmentation and Reassembly

Switching and Forwarding

- **Switch** (commutatore) → interconnette più rami di rete allo scopo di farlo sembrare un'unica rete
  - A mechanism that allows us to interconnect links to form a large network
  - A multi-input, multi-output device which transfers packets from an input to one or more outputs
  - Usually, these links are homogenous: of the same type, with the same address space, but possibly with different technologies (e.g.: various Ethernet variants)



Switching and Forwarding

lo switch prende un frame da una porta e lo inoltra ad un'altra, dopo aver fatto gli appositi controlli d'errore → è un commutatore di liv. 2

- A switch is connected to a set of links and for each of these links, runs the appropriate data link protocol to communicate
- A switch's primary job is to receive incoming packets on one of its links and to transmit them on some other link
  - This function is referred as **switching and forwarding**
  - According to OSI architecture this is the main function of the network layer (3)
  - But can be implemented also at layer 2 (datalink)

riceve il frame, lo analizza e lo ritrasmette al suo destinatario

lo store and forward introduce ritardo (tempo di CONTROLLO + tempo di RITRASMISSIONE)

↳ risolto parzialmente con il CUT THROUGH

STORE AND FORWARD:
1. riceve intero frame da un link
2. lo controlla (CRC, ecc.), se invalido lo droppa senza avviso
3. usando una tabella interna decide a quale link inoltrarlo
4. invia il frame sul link scelto

The star topology

modello SEMPLICE, ogni dispositivo ha il suo dominio e NON CI SONO COLLISIONI

- The use of switches adds the **star topology** to the point-to-point link (PPP, modems), bus (Ethernet), and ring (802.5 and FDDI) topologies

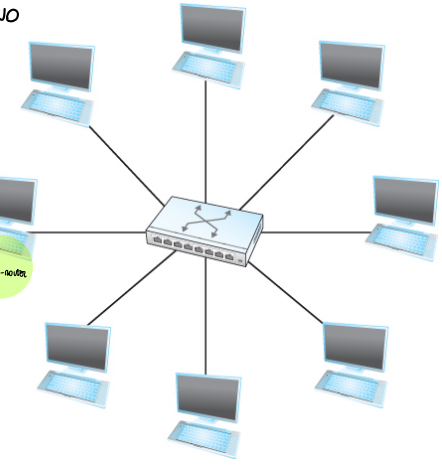
↳ COAX

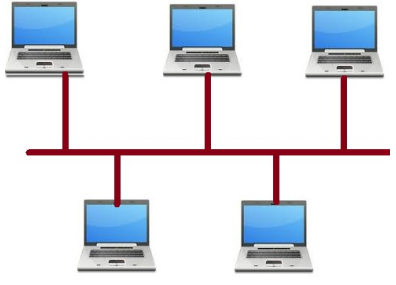
finge di essere un mezzo condiviso

⇒ STAR TOPOLOGY: selettivo → Ethernet "virtuale" (es. switch-pc)
PPP: collegamento singolo → es. tethering o collegamento in fibra router-router
BUS: mezzo condiviso → Ethernet (coax)

- Each spoke of the hub can be seen as a separate collision domain, so no collisions between host and star center
  - In full duplex links, data can be transferred between host and switch in both directions at maximum speed
- can be used to simulate a shared medium

↳ RAMO





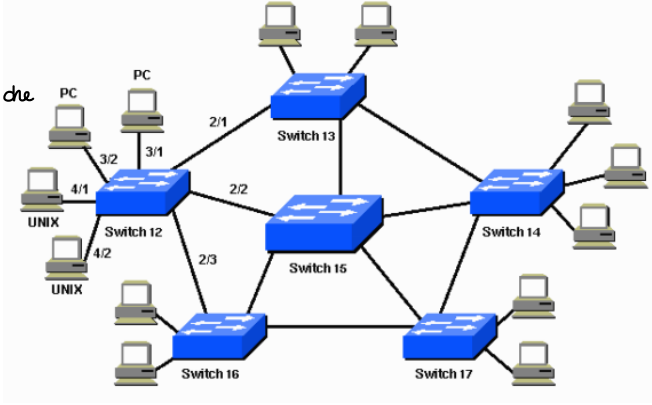
essendo che se si rompe lo switch salta tutto (nelle reti a stella), questi devono essere affidabili

The star topology

- Properties of star topology
  - A switch has a fixed number of inputs and outputs, which limits the number of hosts that can be connected to a single switch (64+)
  - Larger networks can be built by interconnecting a number of switches
  - The whole network appears as a single LAN to each host (although delays and latencies are not uniform)

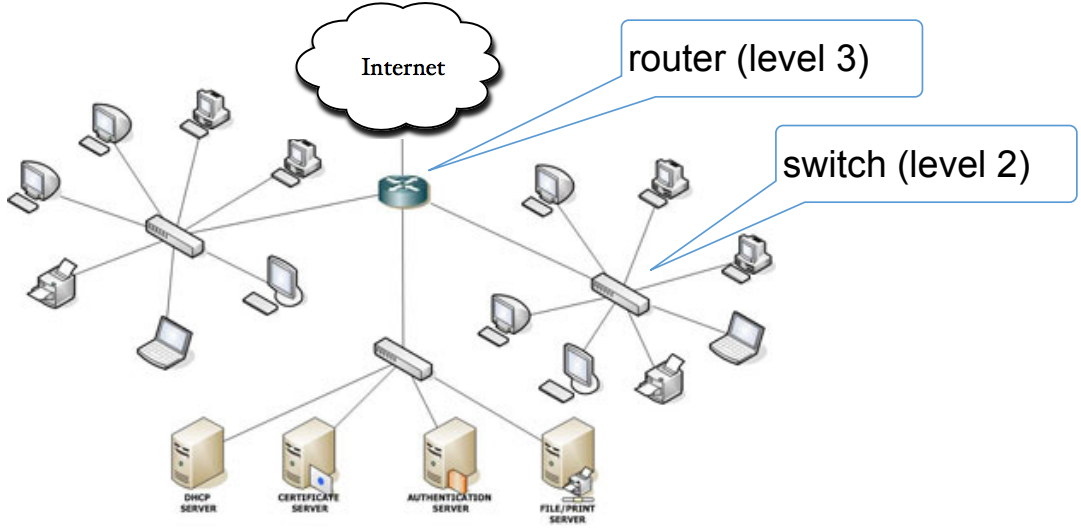
posso collegare SWITCH - SWITCH
SWITCH - HOST

astraggo, è come se virtualmente avessi un unico cavo che collega tutto



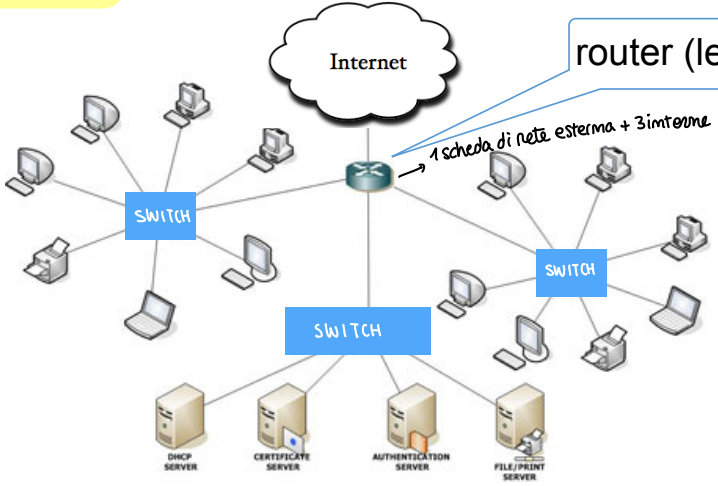
The star topology

- Properties of star topology
  - Different stars/set of stars (i.e., different LANs) can be connected to routers, which are level-3 switches



The star topology → gli SWITCH sono invisibili agli host, loro vedono un'unica lan

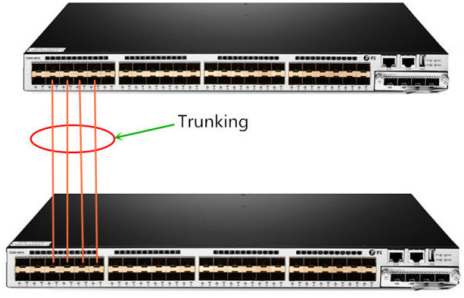
- Properties of star topology
  - From the level-3, switched networks are just another form of local networks



→ 3 reti diverse (3 lan) perché il router non svolge la funzione di fondere le LAN (nella realtà poi è un SWITCH + ROUTER)

The star topology

- Properties of star topology (cont.)
  - hosts/other stars can be connected by point-to-point links
  - These can be normal links on normal ports (called *uplinks*), possibly tied together to appear as a single link (*trunking*)
  - Not to be confused with *stackable switches*, whose backplanes can be connected together via dedicated, proprietary high-speed links
    - Stacked switches act as a single one, not as separate units



The star topology

- Properties of star topology (cont.)
  - networks of large geographic scope can be built
    - MANs/WANs can be built using optic fiber or ATM switches
    - ISPs use xDSL switches (DSLAM) to connect end users over telephonic lines

ATM switch

Optic switch

DSLAM

The star topology

→ es. router con 24 porte con 1 Gbps a porta ⇒ 24 Gbps totale in full duplex (quindi sia TX che RX)
↳ velocità in LAN
in realtà un pò meno per l'overhead

- Properties of star topology (cont.)

→ es. 1 Gbps frame = 14+4+8+1 = 27 byte = 216 bit
frame rate = $\frac{10^9}{216} = 4{,}6 \cdot 10^6$ pacchetti/s e x 24 ≅ 10^8 pacchetti/s = 100 Mbps

  - Adding a new host to the network by connecting it to a switch does not necessarily mean that the hosts already connected will get worse performance from the network
    - Every host on a switched network has its own link to the switch, so it is possible for many hosts to transmit at the full link speed (bandwidth)
    - The limit is given by the *internal maximum switching speed* ("backplane bandwidth") - in the order of tens of giga packets per second (Gpps) for high-end devices
  - This is not true for shared media networks
    - Two hosts on the same Ethernet cannot transmit continuously at full speed because they share the same transmission medium

Switching and Forwarding

- How does the switch decide which output port to place each packet on?
  - It looks at the header of the packet for an identifier that it uses to make the decision
  - Three approaches → l'importante è che all'interno della rete tutti gli switch usino lo stesso approccio
    - *Datagram* or *Connectionless* approach → switch conoscono l'intera mappa della RETE → ogni frame contiene tutti gli indirizzi perché ogni pacchetto è inviato come singolo
    - *Virtual circuit* or *Connection-oriented* approach → ogni frame contiene il VCI e basta in quanto tutto il circuito è stato creato prima dagli switch (Solo il pacchetto di richiesta di connessione richiede gli indirizzi completi)
    - *Source routing* (less common) → ogni volta che A deve trasmettere a B viene creata una connessione "diretta"
  - All the switches of the network must adhere to the same method
- Exchanged units: Frames (level 2), Packets (level 3), Datagrams (level 4), Messages (level 7),…

Switching and Forwarding

- Assumptions
  - Each host has a globally unique address
    - level 2: MAC address
    - level 3: IP address, etc.
  - There is some way to identify the input and output ports of each switch
    - We can use numbers (one for each port)
    - We can use names (of the nodes the port is connected to)
    - For sake of simplicity, let us use numbers

Switching and Forwarding: Connectionless

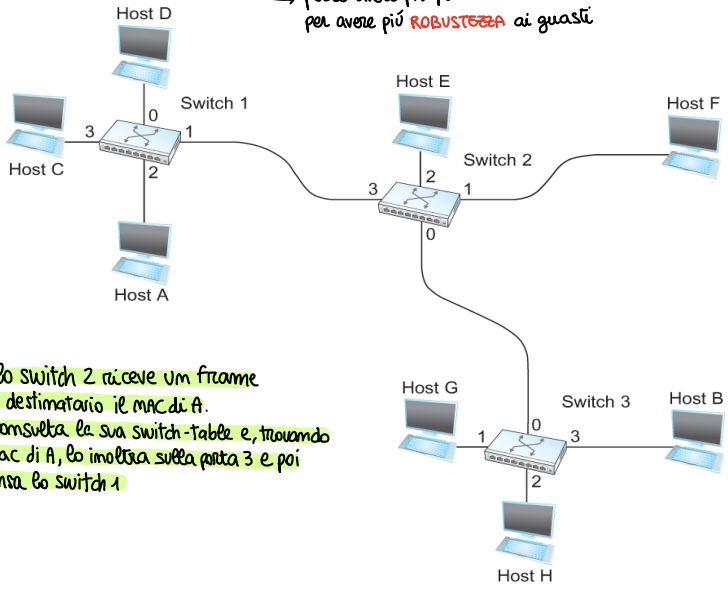
↳ ogni singolo pacchetto trattato in autonomia, senza ricordarsi passato e futuro
ogni pacchetto deve avere abbastanza informaz. per essere indirizzato, lo switch non si ricorda cosa è passato prima

- Key Idea
  - Every packet contains enough information to enable any switch to decide how to get it to destination
    - Every packet contains the complete destination address
  - Each switch forwards each packet just by looking at its header
  - no state is kept in switches

Switching and Forwarding: Connectionless

- An example network is aside
- To decide how to forward a packet, a switch consults a **forwarding table** (sometimes called a **routing table**)
- Eg, Forwarding Table for Switch 2:

→ posso avere più percorsi alternativi per avere più ROBUSTEZZA ai guasti



MAC addr.

Dest.	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

=> il problema è tenere aggiornate le forwarding table

→ es. lo switch 2 riceve un frame con destinatario il MAC di A. Lui consulta la sua switch-table e, trovando il mac di A, lo inoltra sulla porta 3 e poi ci pensa lo switch 1

Switching and Forwarding: Connectionless

- Characteristics of Connectionless (Datagram) Network
  - A host can send a packet anywhere at any time, since any packet that turns up at the switch can be immediately forwarded (assuming a correctly populated forwarding table)
  - When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running
  - Each packet is forwarded independently of previous packets that might have been sent to the same destination.
    - Thus two successive packets from host A to host B may follow completely different paths
  - A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and update the forwarding table accordingly

Switching and Forwarding: Connectionless

- Characteristics of Connectionless (Datagram) Network (cont)
  - We expect the forwarding tables to be updated and correct
  - Who fills the tables?
    - Human: Viable only for small networks
    - Automatic: Routing algorithms running in the background, checking for link status and automatically updating tables
      - "Distributed autonomic system"

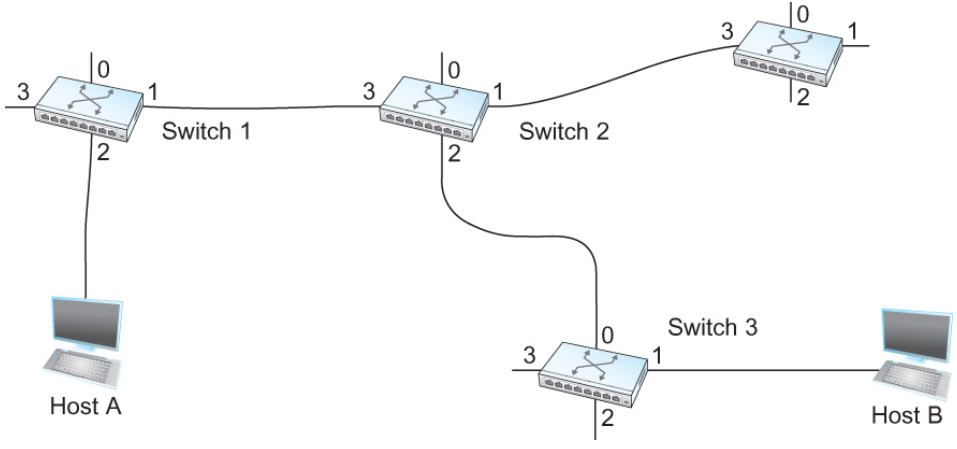
Switching and Forwarding: Virtual Circuit

- Virtual Circuit Switching
  - Widely used technique for packet switching
  - Uses the concept of virtual circuit (VC)
  - Also called a **connection-oriented model** ≠ TCP
  - First set up a virtual connection from the source host to the destination host and then send the data

↳ prima di inviare i dati, devo creare una "connessione virtuale" tra A e B
e poi mi basta avere il VCI nei pacchetti "non primi" per trovare l'indirizzo su cui spedire

Switching and Forwarding: Virtual Circuit

- Host A wants to send packets to host B
- Two-stage process
  - Connection setup
  - Data Transfer



Switching and Forwarding: Virtual Circuit

- Connection setup
  - Establish "connection state" in each of the switches between the source and destination hosts
  - The connection state for a single connection consists of an entry in the "VC table" in each switch through which the connection passes
- The semantics for one such entry is:
  - If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then the packet should be sent out the specified outgoing interface with the specified outgoing VCI value first having been placed in its header

Switching and Forwarding: Virtual Circuit

- One entry in the VC table on a single switch contains
  - A virtual circuit identifier (VCI) that uniquely identifies the connection at this switch and that will be carried inside the header of the packets that belong to this connection (cont.)
  - An incoming interface on which packets for this VC arrive at the switch
  - An outgoing interface in which packets for this VC leave the switch
  - A potentially different VCI that will be used for outgoing packets

ha valore identificativo solo sulla corrente linea, dato che non è globale, infatti INCOMING VCI OUTGOING VCI SONO DIVERSI

"se dall'interfaccia di ingresso designata arriva un pacchetto che contiene nella propria intestazione il VCI designato, allora il pacchetto deve essere inviato all'interfaccia di uscita specificata dopo che il VCI specificato per l'uscita è stato inserito nella sua intestazione"

Switching and Forwarding: Virtual Circuit

- The combination of the VCI of the packets as they are received at the switch and the interface on which they are received *uniquely identifies* the virtual connection
- There may be many virtual connections established in the switch at one time
- Incoming and outgoing VCI values are not generally the same
  - VCI is not a globally significant identifier for the connection; rather it has significance only on a given link
- Whenever a new connection is created, we need to assign a new VCI for that connection on each link that the connection will traverse
  - We also need to ensure that the chosen VCI on a given link is not currently in use on that link by some existing connection.

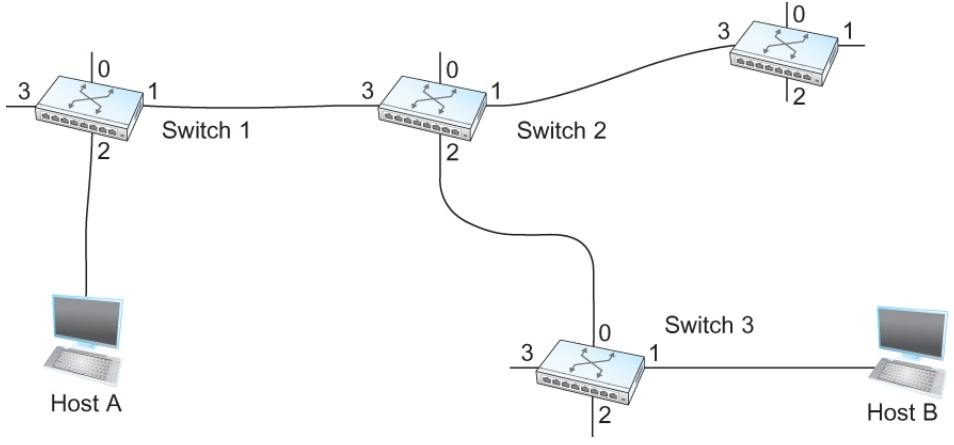
Switching and Forwarding: Virtual Circuit

2 MODALITÀ di INSTAURAMENTO DELLA CONNESSIONE:

- Two broad classes of approach to establishing connection state
  - Network Administrator will configure the state
    - The virtual circuit is *permanent* (PVC)
    - The network administrator can delete this
    - Can be thought of as a long-lived or administratively configured VC
  - A host can send messages into the network to cause the state to be established
    - This is referred as *signalling* and the resulting virtual circuit is said to be switched (SVC)
    - A host may set up and delete such a VC dynamically without the involvement of a network administrator

Switching and Forwarding: Virtual Circuit

- Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B
  - First the administrator identifies a path through the network from A to B



Switching and Forwarding: Virtual Circuit

- The administrator then picks a VCI value that is currently unused on each link for the connection
  - For our example,
    - Suppose the VCI value 5 is chosen for the link from host A to switch 1
    - 11 is chosen for the link from switch 1 to switch 2
    - So the switch 1 will have an entry in the VC table

Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
2	5	1	11

Switching and Forwarding: Virtual Circuit

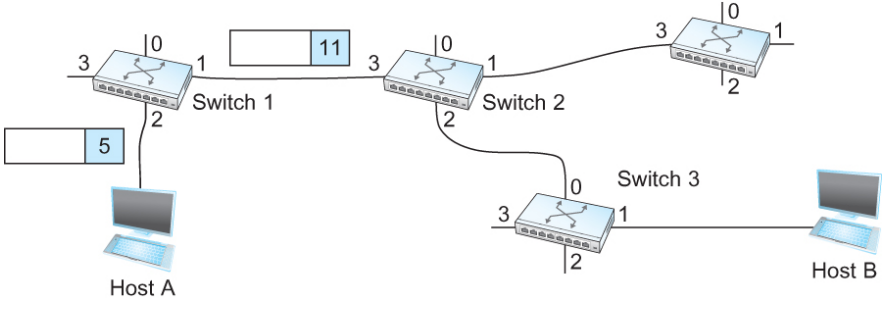
- Similarly, suppose
  - VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3
  - VCI of 4 is chosen for the link from switch 3 to host B
  - Switches 2 and 3 are configured with the following VC table

Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
3	11	2	7

Incoming Interface	Incoming VC	Outgoing Interface	Outgoing VC
0	7	1	4

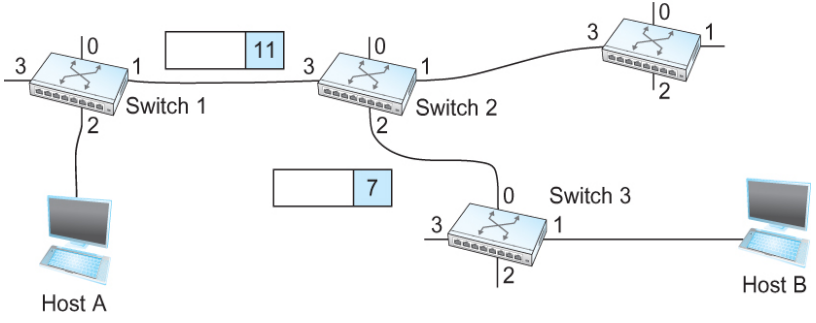
Switching and Forwarding: Virtual Circuit

- For any packet that A wants to send to B, A puts the VCI value 5 in the header of the packet and sends it to switch 1
- Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry.
- The table entry on switch 1 tells the switch to forward the packet out of interface 1 and to put the VCI value 11 in the header



Switching and Forwarding: Virtual Circuit

- Packet will arrive at switch 2 on interface 3 bearing VCI 11
- Switch 2 looks up interface 3 and VCI 11 in its VC table and sends the packet on to switch 3 after updating the VCI value appropriately
- This process continues until it arrives at host B with the VCI value of 4 in the packet
- To host B, this identifies the packet as having come from host A



Switching and Forwarding: Virtual Circuit

- In real networks of reasonable size, the burden of configuring VC tables correctly in a large number of switches would quickly become excessive
  - Thus, some sort of *signalling* is almost always used, even when setting up “permanent” VCs
  - In case of Permanent VCs, signalling is initiated by the network administrator
  - SVCs are usually set up using signalling by one of the hosts

Switching and Forwarding: Virtual Circuit

↳ CREAZIONE del CIRCUITO tra due host usando un ALGORITMO

- How does the signalling work
  - To start the signalling process, host A sends a setup message into the network (i.e. to switch 1)
    - The setup message contains (among other things) the complete destination address of B.
    - The setup message needs to get all the way to B to create the necessary connection state in every switch along the way
    - It is like sending a datagram to B where every switch knows which output to send the setup message so that it eventually reaches B
    - Assume that every switch knows the topology to figure out how to do that
  - When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its VC table for this new connection
    - The entry is exactly the same shown in the previous table
    - Switch 1 picks the value 5 for this connection

Switching and Forwarding: Virtual Circuit

- How does the signalling work (contd.)
  - When switch 2 receives the setup message, it performs the similar process and it picks the value 11 as the incoming VCI
  - Similarly switch 3 picks 7 as the value for its incoming VCI
    - Each switch can pick any number it likes, as long as that number is not currently in use for some other connection on that port of that switch
  - Finally the setup message arrives at host B.
  - Assuming that B is healthy and willing to accept a connection from host A, it allocates an incoming VCI value, in this case 4.
    - This VCI value can be used by B to identify all packets coming from A

Switching and Forwarding: Virtual Circuit

- Now to complete the connection, everyone needs to be told what their downstream neighbour is using as the VCI for this connection
  - Host B sends an acknowledgement of the connection setup to switch 3 and includes in that message the VCI value that it chose (4)
  - Switch 3 completes the VC table entry for this connection and sends the acknowledgement on to switch 2 specifying the VCI of 7
  - Switch 2 completes the VC table entry for this connection and sends acknowledgement on to switch 1 specifying the VCI of 11
  - Finally switch 1 passes the acknowledgement on to host A telling it to use the VCI value of 5 for this connection

Switching and Forwarding: Virtual Circuit

- When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1
- The switch 1 removes the relevant entry from its table and forwards the message on to the other switches in the path which similarly delete the appropriate table entries
- At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed

Switching and Forwarding: Virtual Circuit

- Characteristics of VC
  - Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one RTT of delay before data is sent
  - While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link.
    - Thus the per-packet overhead caused by the header is reduced relative to the datagram model

Switching and Forwarding: Virtual Circuit

- Characteristics of VC (cont.)
  - If a switch or a link in a connection fails, the connection is broken and a new one will need to be established.
    - Also the old one needs to be torn down to free up table storage space in the switches
  - The issue of how a switch decides which link to forward the connection request on has similarities with the function of a routing algorithm

Switching and Forwarding: Virtual Circuit

- Good Properties of VC
  - By the time the host gets the go-ahead to send data, it knows quite a lot about the network-
    - For example, that there is really a route to the receiver and that the receiver is willing to receive data
  - It is also possible to allocate resources to the virtual circuit at the time it is established

Switching and Forwarding: Virtual Circuit

- Example: X.25 network (an old connection-oriented packet-switched network) employs the following three-part strategy
  - Buffers are allocated to each virtual circuit when the circuit is initialized
  - The sliding window protocol is run between each pair of nodes along the virtual circuit, and this protocol is augmented with the flow control to keep the sending node from overrunning the buffers allocated at the receiving node
  - The circuit is rejected by a given node if not enough buffers are available at that node when the connection request message is processed

Switching and Forwarding: Virtual Circuit

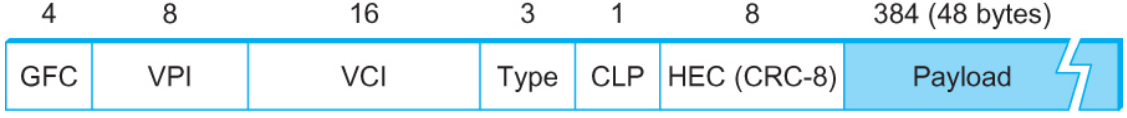
- Comparison with the Datagram Model
  - Datagram network has no connection establishment phase and each switch processes each packet independently
  - Each arriving packet competes with all other packets for buffer space
  - If there are no buffers, the incoming packet must be dropped
- In VC, we could imagine providing each circuit with a different quality of service (QoS)
  - The network gives the user some kind of performance related guarantee
    - Switches set aside the resources they need to meet this guarantee
      - For example, a percentage of each outgoing link's bandwidth
      - Delay tolerance on each switch

Switching and Forwarding

- Most popular examples of VC technologies are Frame Relay and ATM
- ATM (**Asynchronous Transfer Mode**)
  - Connection-oriented packet-switched network
  - Packets are called **cells**
    - Fixed size: 5 byte header + 48 byte payload
  - Fixed length packets are easier to switch in hardware
    - Simpler to design
    - Enables parallelism

Switching and Forwarding

- ATM
  - User-Network Interface (UNI)
    - Host-to-switch format
    - GFC: Generic Flow Control
    - VCI: Virtual Circuit Identifier
    - Type: management, congestion control
    - CLP: Cell Loss Priority
    - HEC: Header Error Check (CRC-8)

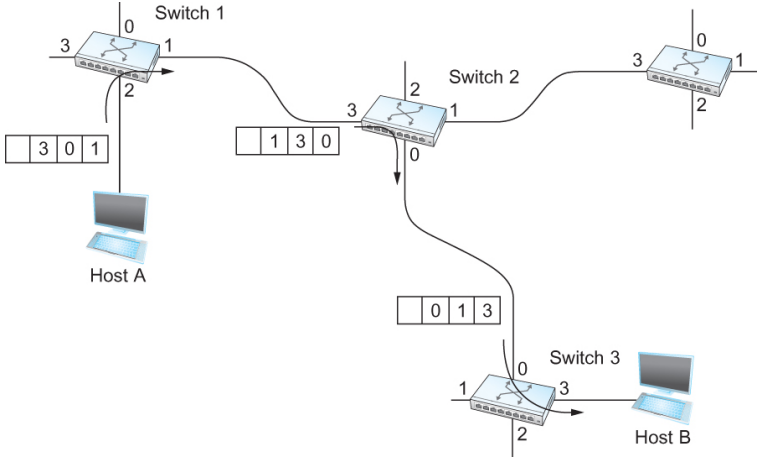


  - Network-Network Interface (NNI)
    - Switch-to-switch format
    - GFC becomes part of VPI field

Switching and Forwarding: Source Routing

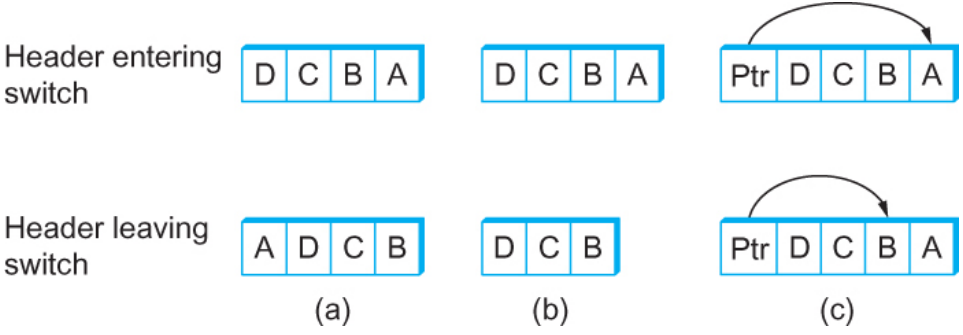
Scrivo nell'intestazione il percorso che deve fare le pacchetto (a priori)
La sorgente programma il percorso, gli switch sono stupidi

- Source Routing
  - All the information about network topology that is required to switch a packet across the network is provided by the source host



Switching and Forwarding: Source Routing

- Other approaches in Source Routing



Bridges and LAN Switches

- Bridges and LAN Switches
  - Class of switches that is used to forward packets between shared-media LANs such as Ethernets
    - Known as **LAN switches**
    - Referred to as *Bridges*
    - usually by “switch” we intend LAN (i.e. level 2) switches, even if there can be switches of any level

Bridges and LAN Switches

Repeater collega 2 rami di Ethernet e passa i bit da un ramo all'altro
non fa controlli, è di liv. 1 → modo iniziale con cui si collegavano le varie Ethernet

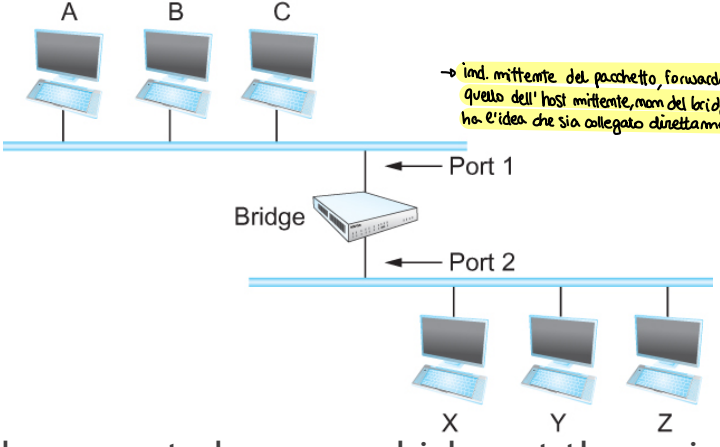
- Suppose you have a pair of Ethernets that you want to interconnect
  - One approach is put a repeater in between them
    - A repeater just decodes/encodes bits
    - It might exceed the physical limitation of the Ethernet
      - No more than four repeaters between any pair of hosts
      - No more than a total of 2500 m in length is allowed
  - An alternative would be to put a node between the two Ethernets and have the node forward frames from one Ethernet to the other
    - This node is called a *Bridge*
    - A collection of LANs connected by one or more bridges is usually said to form an *Extended LAN*

Bridges and LAN Switches

- Simplest Strategy for Bridges: accept LAN frames on their inputs and forward them out to all other outputs
  - Used by early bridges (*hubs*), now obsolete
- **Learning Bridges**
  - Observe that there is no need to forward to everyone all the frames that a bridge receives
  - instead, each frame can be forwarded only to the relevant host(s)

Bridges and LAN Switches

- Consider the following figure
  - When a frame from host A that is addressed to host B arrives on port 1, there is no need for the bridge to forward the frame out over port 2.



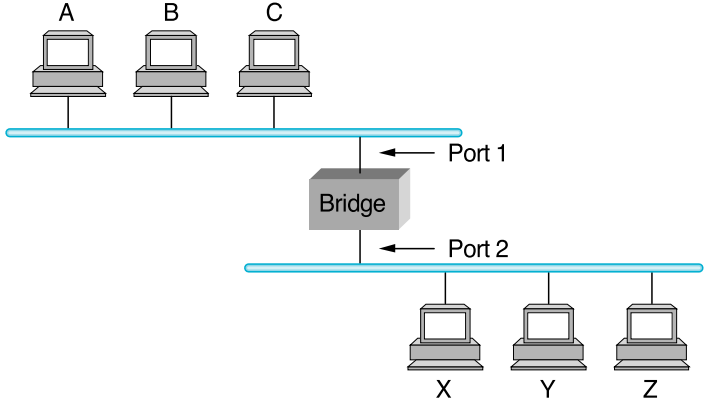
→ ind. mittente del pacchetto, forwardato dal bridge, è quello dell'host mittente, non del bridge così il destinatario ha l'idea che sia collegato direttamente

- How does a bridge come to learn on which port the various hosts reside?

↳ sa dove sono posizionati gli host → come fa a saperlo? - lo fai a mano
quando passano i frame lui osserva - apprenda automaticamente osservando il traffico

Bridges and LAN Switches

- Solution: Download a table into the bridge



Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

  - Who does the download?
    - Human? Too much work for maintenance

Bridges and LAN Switches

- The bridge can learn this information by *inspecting* the source address in *all the frames it observes at its ports*
- Record the information at the bridge and build the table
- When a bridge first boots, this table is empty
  - Entries are added over time → aggiungiamo una (host) entry quando questa passa
  - A timeout is associated with each entry
  - The bridge discards the entry after a specified period of time, to protect against the situation in which a host is moved from one network to another → anche per evitare situazioni in cui un host si spegne
- If the bridge receives a frame that is addressed to host not currently in the table, forwards the frame out on all other ports

Bridges and LAN Switches

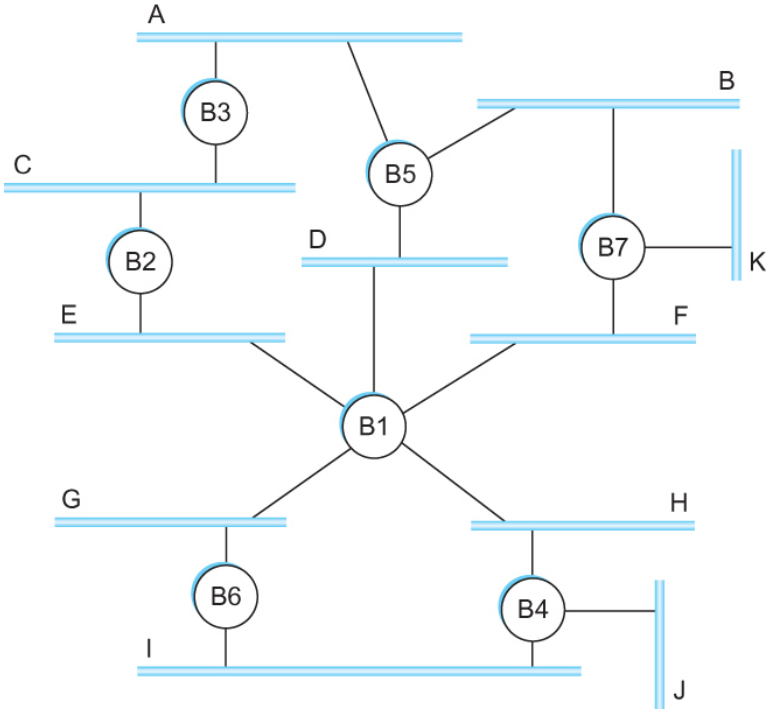
- Broadcast:
  - Forward all broadcast frames to all ports (apart the port where the frames come from)
- Multicast
  - Simple strategy: forward all multicast frames to all ports (apart the port where the frames come from), as broadcast
  - More sophisticated: Learn when no group members are associated to a port
    - Accomplished by having each member of group G send a frame to bridge multicast address with G in source field

↳ Simulato facendo il flooding

↳ Lo fanno gli switch economici, poi sarà l'host a scartare

Bridges and LAN Switches

- This strategy works fine as long as the extended LAN does not have a loop in it
  - Frames potentially loop through the extended LAN forever
  - Bridges B1, B4, and B6 form a loop



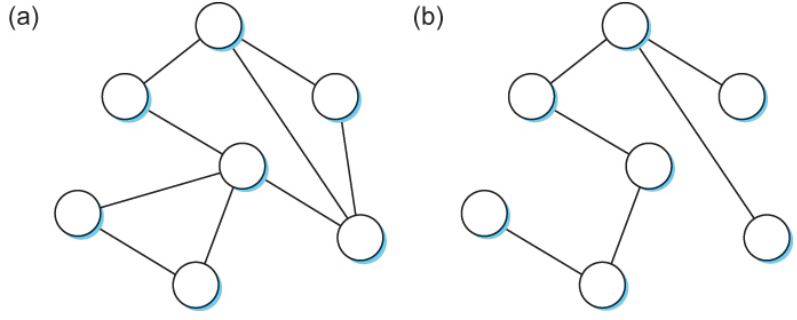
Bridges and LAN Switches

- How does an extended LAN come to have a loop in it?
  - Network is managed by more than one administrator
    - For example, it spans multiple departments in an organization
    - It is possible that no single person knows the entire configuration of the network
      - A bridge that closes a loop might be added without anyone knowing
  - Loops are built into the network to provide redundancy in case of failures → es. se si guasta uno switch posso usare un altro percorso
- Solution: **Distributed Spanning Tree Algorithm** → implementato in tutti gli switch

Spanning Tree Algorithm

- Think of the extended LAN as being represented by a graph that possibly has loops (cycles)
  - Nodes can be bridges and simple LANs
- A spanning tree is a sub-graph of this graph that covers all the vertices (of type simple LANs) but contains no cycles
  - Spanning tree keeps all the vertices of the original graph but throws out some of the edges
  - Example of (a) a cyclic graph; (b) a corresponding spanning tree.

→ gli switch automaticamente spengono alcune porte (che creano ridondanze) per trasformare in un albero



Spanning Tree Algorithm

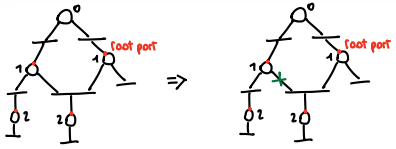
- "Rapid Spanning Tree": a protocol used by a set of bridges to agree upon a spanning tree for a particular extended LAN
- Developed by Radia Perlman at Digital (1985)
- IEEE 802.1D (2004) specification for LAN bridges is based on this algorithm
- Used in almost all switches, nowadays
  - (but beware that in some networks there can be old switches without the Spanning Tree)

Spanning Tree Algorithm

→ ALG. DISTRIBUITO: non c'è MASTER, è P2P. È dinamico, quando stacco e attacco un cavo fa tutto lui da solo

- Main idea: each bridge decides the ports over which it is and is not willing to forward frames
  - By disabling ports from the topology, the extended LAN is reduced to a tree
  - It is even possible that an entire bridge will not participate in forwarding frames
- Algorithm is dynamic
  - The bridges are always prepared to reconfigure themselves into a new spanning tree if some bridges fail

Spanning Tree Algorithm



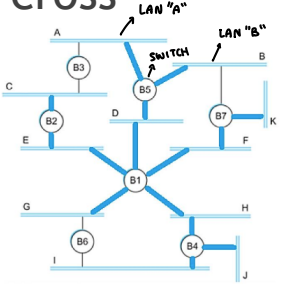
- Algorithm selects ports as follows:
  - Each bridge is given a unique identifier (usually the lowest id of its ports)
    - B1, B2, B3,…and so on
- **The bridge with the smallest id will be elected as the root of the spanning tree**
  - MAC ADDRESS + PRIORITY
- The root bridge always forwards frames out over all of its ports, like a hub
- 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, hence is called **root port**
    - per CIASCUNA LAN si seleziona UNO SWITCH (quello + vicino a root) e gli altri si disabilitano
- Finally, all the bridges connected to a given LAN elect a single designated bridge that will be responsible for forwarding frames upwards and downwards

Spanning Tree Algorithm

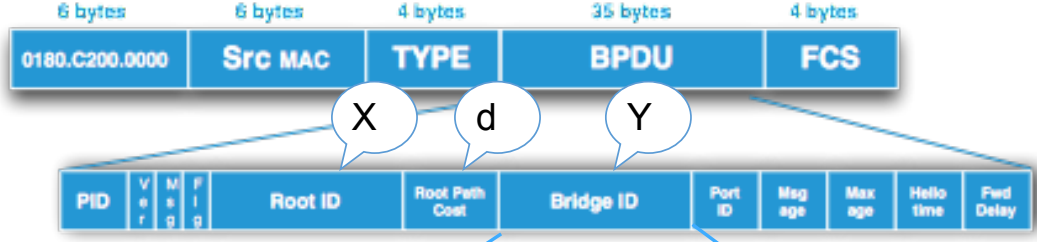
- For each LAN, the bridge which is closest to the root is the designated bridge
  - If two or more bridges are equally close to the root, select bridge with the smallest id
- Each bridge may be connected to more than one LAN
  - So it participates in the election of a designated bridge for each LAN it is connected to.
  - Each bridge decides if it is the designated bridge relative to each of its ports
  - The bridge forwards frames over only those ports for which it is the designated bridge

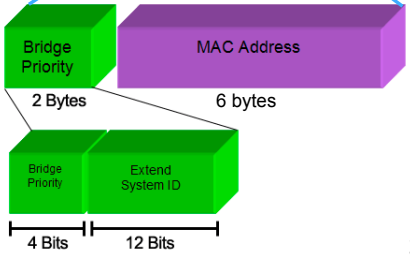
Spanning Tree Algorithm

- Bridges reach this agreement by exchanging simple configuration messages (called BPDU) between immediate neighbours (i.e. over the shared segments)
- Each message is a triple (X,d,Y) where
  - Y is the identifier of the bridge generating the message
  - X is the identifier of the root bridge, according to Y
  - d is the distance from X to Y, in number of LANs to cross
- In reality, these messages are defined by 802.1D, carr other data (like bridge priority, costs, etc) and are ins frames with a special destination address



Spanning Tree Algorithm



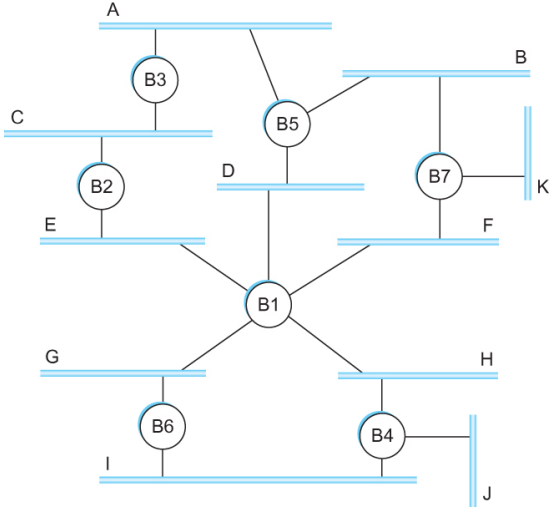


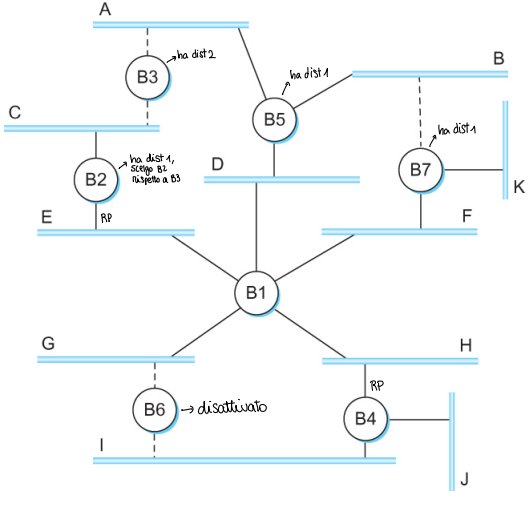
- Each bridge ID is composed by the MAC address and a 16-bit bridge priority (of which, only the first 4 bits are the real priority, the others indicate the VLAN - not relevant here)
- Default priority is 32769 = 1000 000000000001
- It can be changed by the administrator, if he wants a specific bridge to become the root.
  - E.g., by putting 4097 = 0001 000000000001

Spanning Tree Algorithm

- B1 is the root bridge
- B3 and B5 are connected to LAN A, but B5 is the designated bridge
- B5 and B7 are connected to LAN B, but B5 is the designated bridge

se due switch hanno la stessa distanza, vince quello che ha nome più basso





Spanning Tree Algorithm

- Initially each bridge X thinks it is the root, so it sends a configuration message (X, 0, X) on each of its ports identifying itself as the root and giving a distance to the root of 0
- Upon receiving a configuration message (Y, d, Z) over a particular port, the bridge checks to see if the new message is better than the current best configuration message recorded for that port
- The new configuration is better than the currently recorded information if
  - It identifies a root with a smaller id or
  - It identifies a root with an equal id but with a shorter distance or
  - The root id and distance are equal, but the sending bridge has a smaller id

es. B4 (B4,0,B4) riceve (B2,0,B2) e vede se la distanza è migliore / la root è migliore → alfabeticamente B1 < B4 ⇒ ora, per B4 la root è B2. e quindi modifica la sua dist. dalla root (B1) a 1 ⇒ (B2,1,B4)
↳ ID che è MAC+PRIORITY
Si guarda prima l'id e poi la distanza. altrimenti per ogni modo ad avere C'dall'attuale bridge root

Spanning Tree Algorithm

- If the new message is better than the currently recorded one,
  - The bridge discards the old information and saves the new information
  - It first adds 1 to the distance-to-root field

↳ quando non è ROOT

- When a bridge receives a configuration message indicating that it is not the root bridge (that is, a message from a bridge with smaller id)
  - The bridge stops generating configuration messages on its own
  - Only forwards configuration messages from other bridges after 1 adding to the distance field

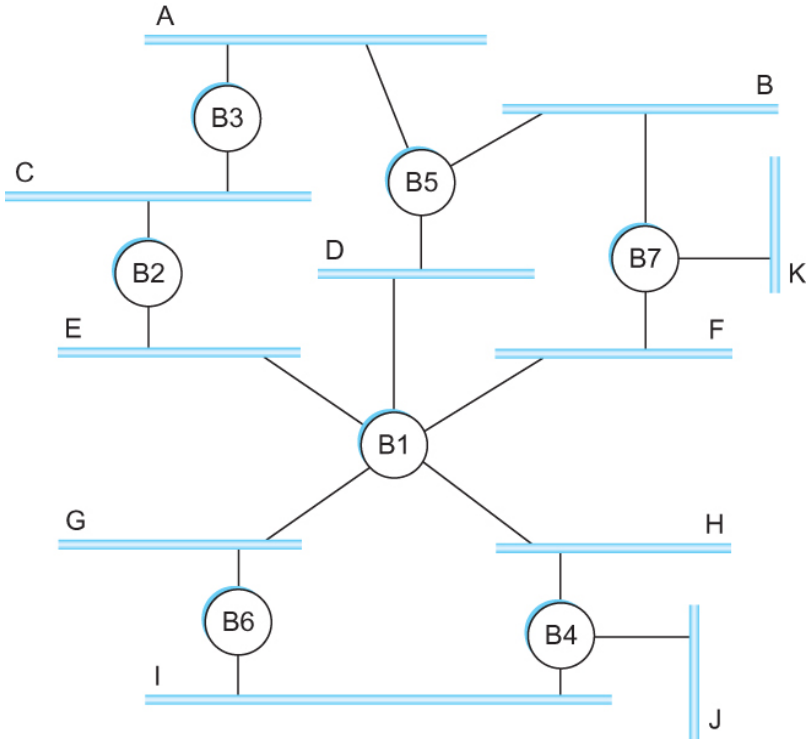
Spanning Tree Algorithm

↳ quando non è designato, ovvero deve spegnersi

- When a bridge receives a configuration message that indicates it is not the designated bridge for that port
  - i.e., a message from a bridge that is closer to the root or equally far from the root but with a smaller id
    - The bridge stops sending configuration messages over that port
- When the system stabilizes,
  - Only the root bridge is still generating configuration messages.
  - Other bridges are forwarding these messages only over ports for which they are the designated bridge

Spanning Tree Algorithm

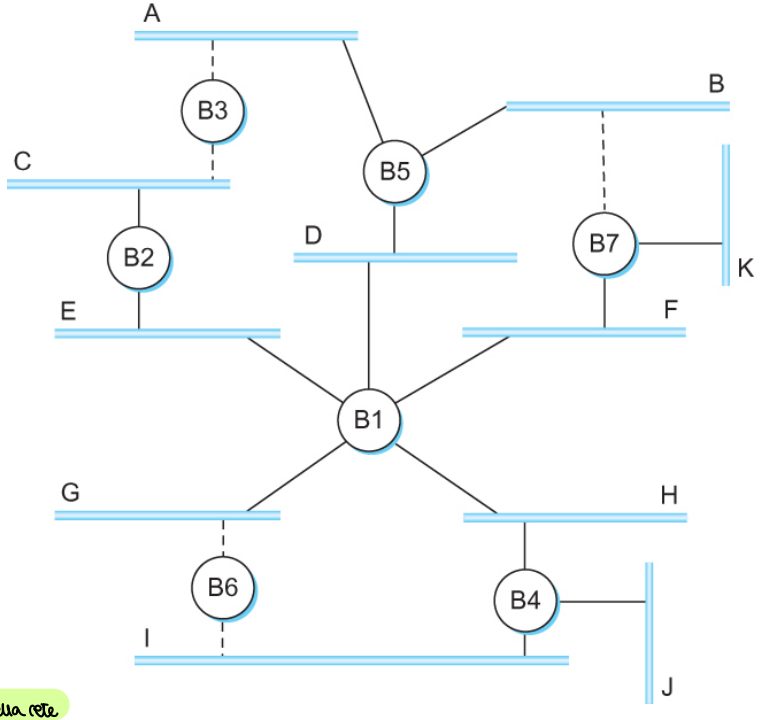
- Consider the situation when the power had just been restored to the building housing the following network
- All bridges would start off by claiming to be the root
- Denote a configuration message from node X in which it claims to be distance d from the root node Y as (Y, d, X)



Spanning Tree Algorithm

→ è sempre deterministico, vince chi ha l'id più basso (MAC + PRIORITY)

- Consider the activity at node B3
- B3 receives (B2, 0, B2)
- Since 2 < 3, B3 accepts B2 as root
- B3 adds 1 to the distance advertised by B2 and sends (B2, 1, B3) to B5
- Meanwhile B2 accepts B1 as root because it has the lower id and it sends (B1, 1, B2) toward B3
- Similarly B5 accepts B1 as root and sends (B1, 1, B5) to B3
- B3 accepts B1 as root and it notes that both B2 and B5 are closer to the root than it is.
  - Thus B3 stops forwarding messages on both its interfaces
  - This leaves B3 with both ports not selected



es. B7 non disattiva la sua porta sulla rete di K perché non sente messun altro PDU su quella rete

Spanning Tree Algorithm

- Even after the system has stabilized, the root bridge continues to send configuration messages periodically
  - Other bridges continue to forward these messages
- When a bridge fails, the downstream bridges will not receive the configuration messages
- After waiting a specified period of time, they will once again claim to be the root and the algorithm starts again

Spanning Tree Algorithm

- Limitations of this algorithm
  - Although the algorithm is able to reconfigure the spanning tree whenever a bridge fails, it is not able to forward frames over alternative paths for the sake of routing around a congested bridge → algoritmo che non considera le carico degli switch
  - Moreover, disabling ports in switches leads to longer paths between two LANs (e.g. consider the path between LANs B and K)

Limitation of Bridges

- Do not scale
  - Spanning tree algorithm does not scale (linear time, does not take advantage of hierarchy)
  - Broadcast does not scale: too much traffic on a large scale (nor we need it)
- Do not accommodate heterogeneity
  - all parts of the network must have the same kind of address, and similar features (e.g. broadcast)
- What if two segments admit different MTUs?
  - e.g. Ethernet (802.3): 1500 byte; WiFi (802.11): 2346 byte
  - in one direction, the bridge has to *fragment* a frame into 2 or more frames, which must be recomposed at the datalink level by receiver (*PAF fragmentation*)
- To overcome these limitations, we need to move to level 3 — Internetworking