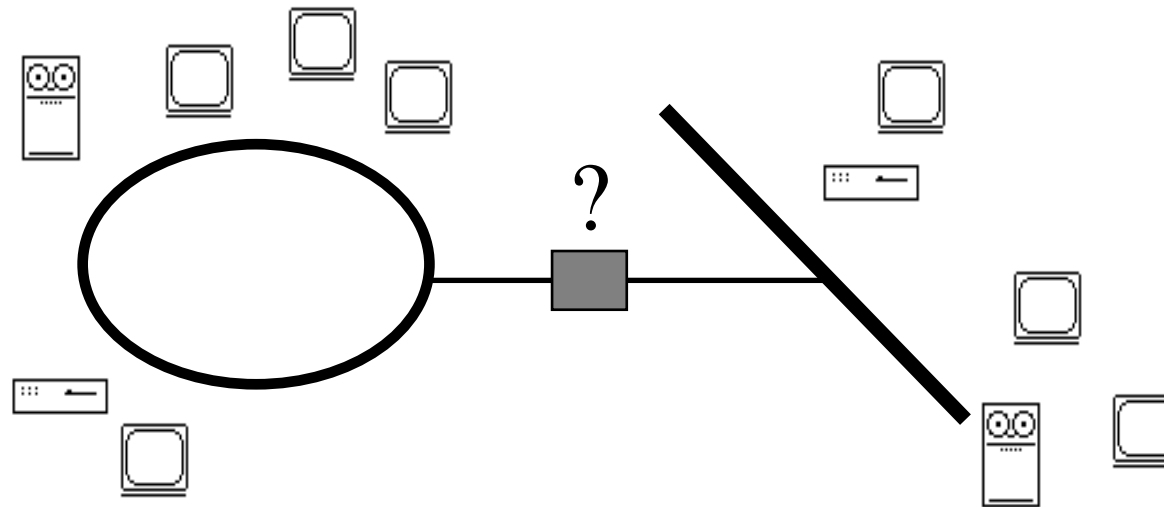


LAN Interconnexion

(/udd/bcousin/ITI-Caire/Cours/2.Bridging.fm- 18 April 2002 19:21)



PLAN

- Presentation
- General architecture
- Physical layer interconnection
- Bridging
- Transparent bridging
- Source routing
- Conclusion

1. Presentation

Networking is heterogeneous:

- many different Network access methods, network infrastructures, etc.

Heterogeneity is permanent:

- old standards
- improvement of current standards
- new technologies

Solutions

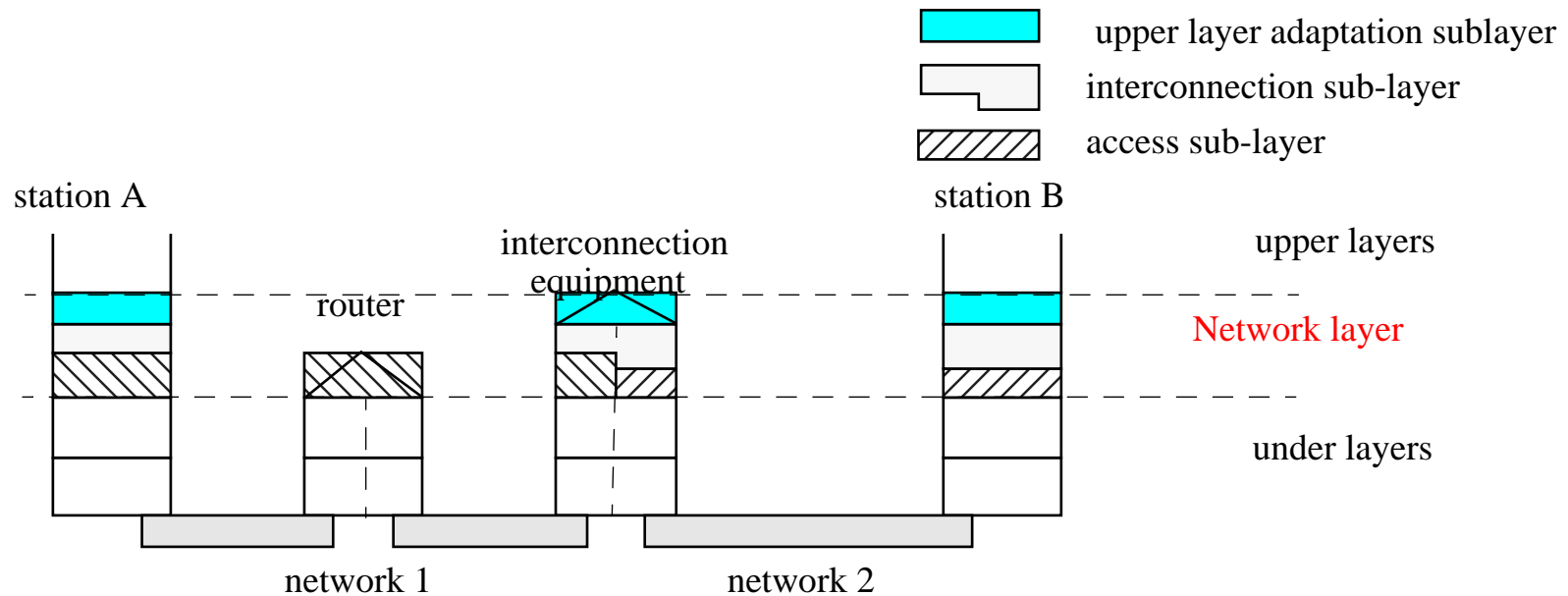
- to interconnect heterogeneous networks
- with interconnection equipments

2. General architecture

2.1. ISO interconnection architecture

3 sub-layers:

- upper-layer adaptation sub-layer
- interconnection sub-layer
- access sub-layer



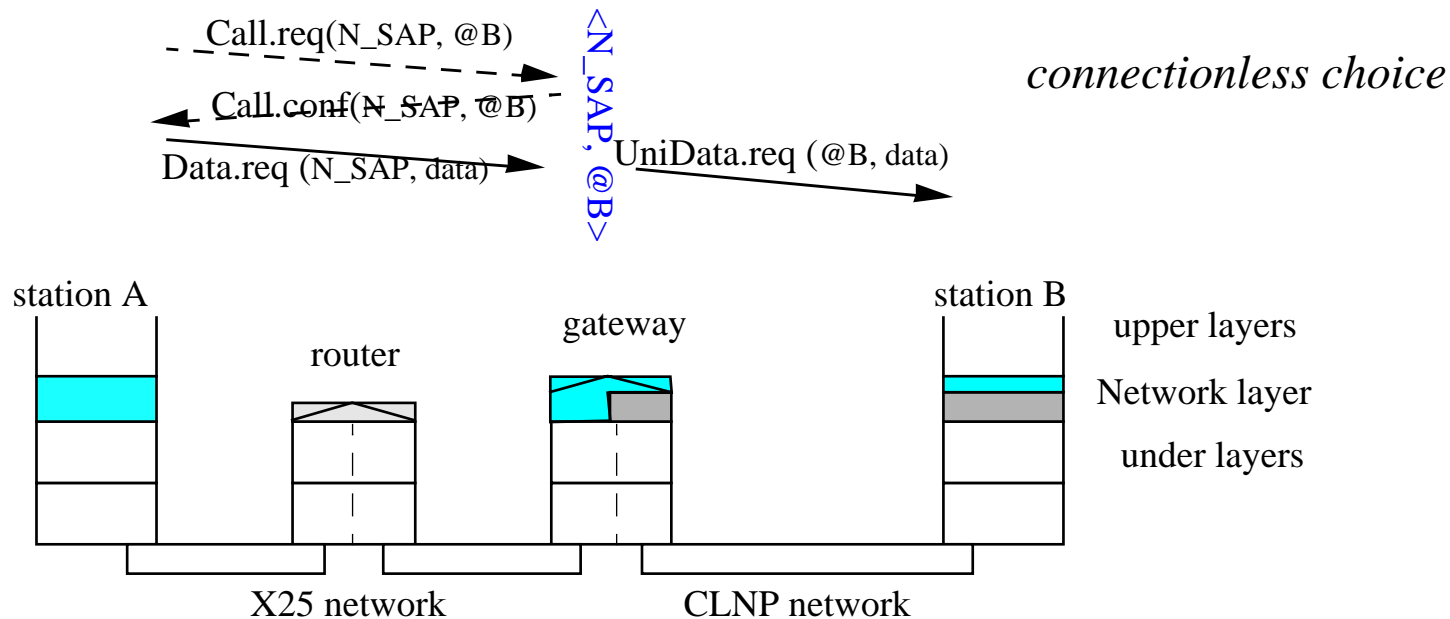
2.2. Interconnection example

Interconnection between a connectionless network and a connection oriented network with (e.g. CNLP -“connectionless network protocol”- and X25.3).

- choice of the interconnection mode:

- . connected mode
- . connectionless mode

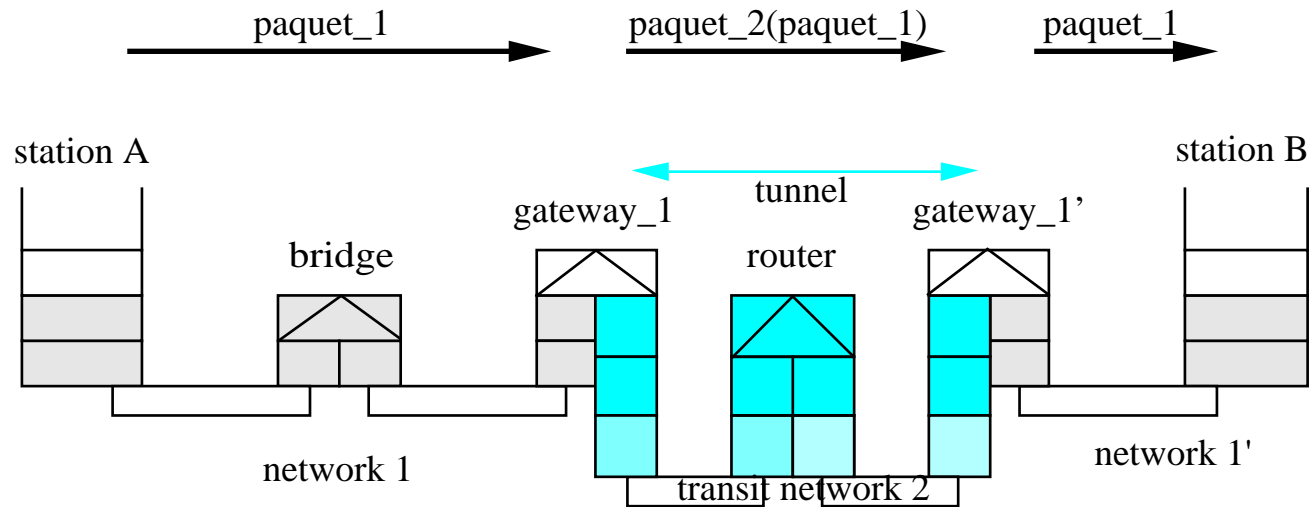
Many problems to solve: address/SAP, management of connection context, connection messages, etc.



2.3. Tunnelling

Interconnection of 2 equivalent networks through a transit network:

- packet transmission through transit network is transparent
- example:
 - . LAN interconnection through Internet = LAN emulation
 - . multicast islands interconnection through unicast network = Mbone

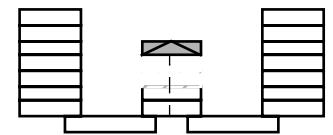
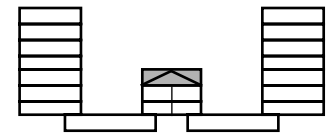
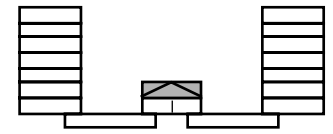
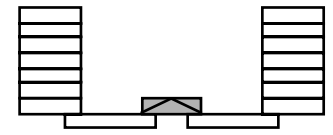


- Packet encapsulation
 - transmitted packets are put in the data field (encapsulated) of transit packets

2.4. Interconnection levels

Interconnection function can be done at every layer:

- Layer 1 (physical): modem, repeater, multiport
 - modulation adaptation to physical support
 - e.g.: opto-electronic repeater between 2 Ethernet segments
- Layer 2 (Data link): hub, bridge
 - translation between heterogeneous Medium access method
 - .e.g: interconnection between Ethernet LANs
- Layer 3 (Network): router
 - provided for!
- Upper layers: gateway, relay, protocol translator
 - .e.g.: SMTP<=>X400 message system



3. Physical layer interconnection

3.1. Introduction

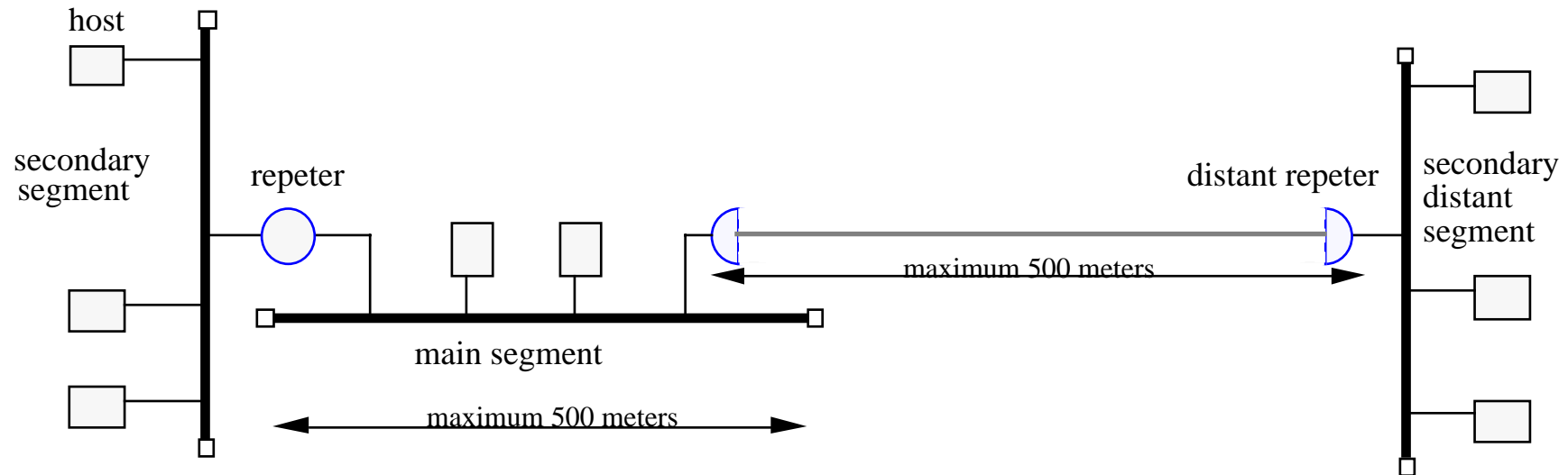
Signal adaptation due to physical support restrictions

- no bit nor frame semantic
- Adaptation between two modulation technics
 - e.g. modem
- Extension of the network area
 - e.g. repeater
- Splitting of one access point to many
 - e.g. multiport
- Combination of any above functions:
 - e.g. (repeater + multiport)

3.2. Ethernet repeater

Historic Ethernet LAN use repeaters for its segments interconnection

- signal amplification, collision enforcement, beaconing propagation
- repeater between homogeneous or heterogeneous segments
- 1 main segment, any number of secondary (distant or not) segments
 - . due to signal transmission delay, collision detection, frame minimum length

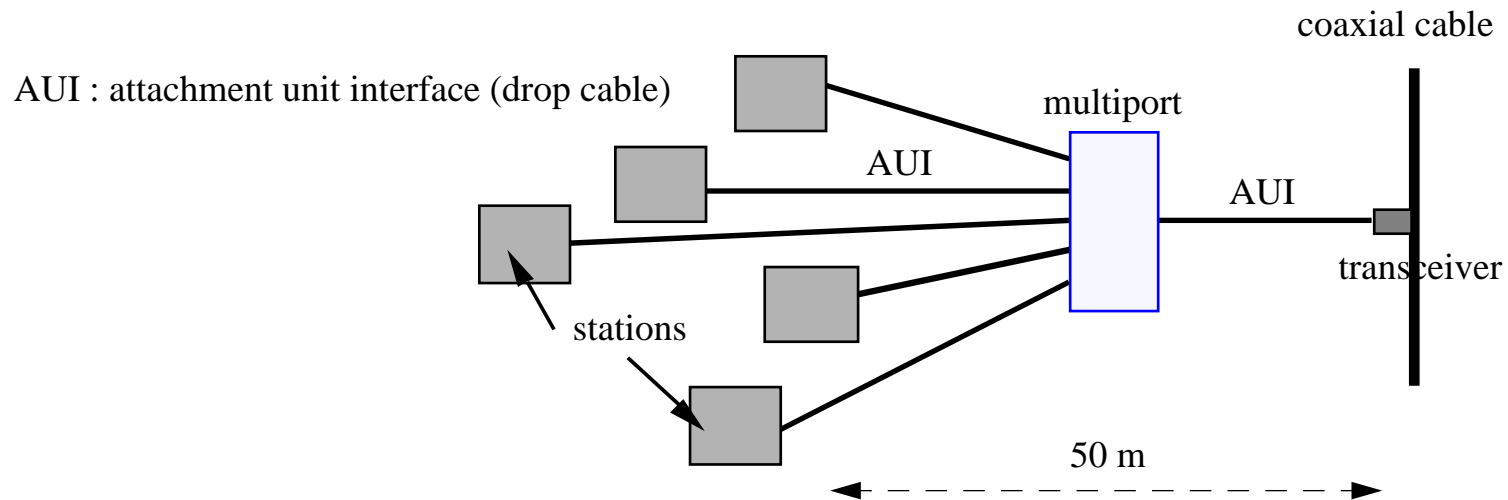


- don't get confuse
 - one LAN segment / one LAN / interconnection of LANs / IP subnet

3.3. Multiport, fan-out, concentrator

Used by historical Ethernet (10base5)

- minimum distance requirement between 2 plugs on coaxial cable
 . 2.5 meter (signal loss due to plug insertion)
- star topology eases central management



3.4. Hub

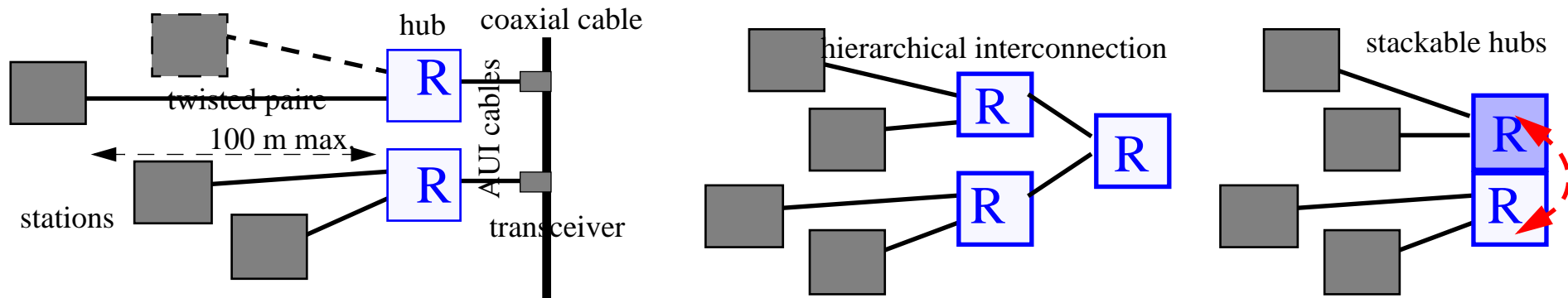
Actual Ethernet architecture: 10xx Base T

Hub = multiport repeater with integrated transceivers

- star placement, hierarchical
- each host station link is a LAN ==> hub is a bridge!
 - no more frame multiplexing <==> full duplex frame transmission

Intelligent equipment:

- failure detection and isolation,
- distant management (SNMP aware)
- security function (frame coding against natural broadcasting of LAN)
- stackable, medium-less: internal bus interconnection



4. Bridging

4.1. Introduction

LAN interconnection (Data link level)

- All LAN build up a larger (virtual) LAN
- Bridge receives data frames from LAN and forwards them on (an)other LAN(s)
- To be forwarded, frames have to get access to the network (medium access control)
 - delay ==> frame buffer
- Addressing
 - IEEE 802 address (e.g. MAC, Ethernet address):
 - . large format and universal address (6 bytes)
- Transparency
 - applications ignore if network uses bridging or not

4.2. Services

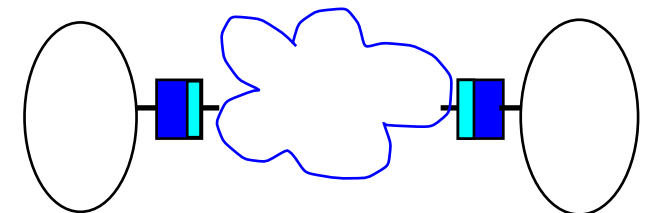
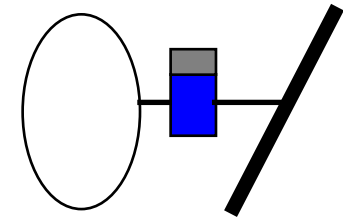
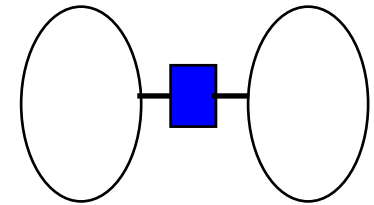
- Interconnection service
 - lowest common denominator between LAN
- Extension of the network **area**
 - direct interconnection
 - indirect interconnection through a transit network: LAN emulation
- Increasing of the **bandwidth**
 - local traffic on one LAN is local
 - potential bandwidth addition
- **Security** management
 - LAN protection:
 - . firewall
 - frame coding
- VLAN (layer 2+)
 - QoS
 - . frame priority
 - tagging
 - . traffic isolation

4.3. Bridging types

2 orthogonal characteristics:

- homogeneous or heterogeneous interconnection
- distant (and indirect) or local interconnection

- interconnected networks are identical
 - frame format and semantic are compatible
- interconnected networks are different
 - frame format translation is required
 - . e.g. maximum length: Ethernet frame \leq FDDI frame
 - . segmentation?
 - service conservation
 - . e.g. Token Ring priorities in Ethernet!
 - . lowest service
- distant interconnection
 - tunnelling technique



4.4. Bridging basis

Flooding

- The most usual bridging function
- On frame reception, bridge forwards the frame to every output link except the input link

Backward learning

- Received frame has information which could be used to route the frames
- Transparent bridging bridge memorizes Input link and Source address of received frame in bridging table entry, for later routing use.

Source routing

- Header frame contains a routing list which enables the selection by the bridge of the next route step to the destination
- The source host puts the routing list into the frame header

5. Transparent bridging

5.1. Introduction

- No station modification
 - stations are unaware of bridge existence
- Bridge auto-configuration
 - bridge default algorithm:
 - . broadcasting (flooding)
 - self-learning:
 - . backward learning
 - cache optimization
- Promiscuous mode
 - bridge processes every frame from every network to which it is connected
 - bridge is potentially a security problem point

5.2. Bridging table and bridge port

Every transparent bridging bridge has a bridging table:

- Table entry = (destination address, port identifier)
- The host identified by the address could be reached (directly or indirectly) from the bridge port
- Limited timelife is associated to each entry

Bridge port:

- each bridge interface giving direct access to a LAN is identified by a port identifier
 - . 2 first bytes: port priority (0 = high priority)
 - => management of port redundancy by network manager
 - . 6 next bytes: address IEEE 802 from the network interface card

Bridge identifier (bridge address)

- bridges are identified by their lowest port identifier

5.3. Transparent bridging algorithm

A frame is received on an input port:

Extraction of the **destination address** then address look up in the bridging table
 if no table entry exists then

 frame flooding to every bridge port except the input port [1]

else

 if entry port \neq input port then

 frame forwarding to the entry port [2]

 [3]

Extraction of the **source address** then address look up in the bridging table

if no table entry exists then

 a new entry is created with input port and source address [1']

else

 entry port = input port [2' and 3']

5.4. Consequences

- Number of entry (destination) could be large
 - lookup time could be too long
- Limited size of bridging table
 - preemption of least recently used entry
- Entry time-out (20 s)
 - inactive entry detection
 - incorrect entry deletion (host failure or transfer)
- LAN can be interconnected through several bridges
 - redundancy against failure
 - incorrect architecture management
- Flooding on cyclic graph
 - multiple frame copy

5.5. Spanning tree

Acyclic graph building: a **tree**

- by desactivation of some bridge ports
- The tree spans all the bridges in the network: total spanning tree
- Many spanning trees exist! We just want to build any of them!

The spanning tree algorithm is an election algorithm based on a value (a 4-uplet):

- the root of the tree is the bridge with **lowest address**
- bridges are selected from the **shortest path** to root
- in case of path length equality, bridge with the lowest address is selected
- in case of dual port bridge, port with the lowest port identifier is selected

5.6. Ring election algorithm

Properties

- each host bids a number
- the elected hosts are the hosts with the lowest number
- every host agree on the winning number

Principle

- each host sends downstream a message with its number,
- on reception, message with lowest number are forwarded and the local host number is replaced by the lowest number
- on reception, message with higher number are discarded

Proof

- the only message which is always forwarded is the message with the lowest bid
- this message turns around the ring, passing through all hosts.

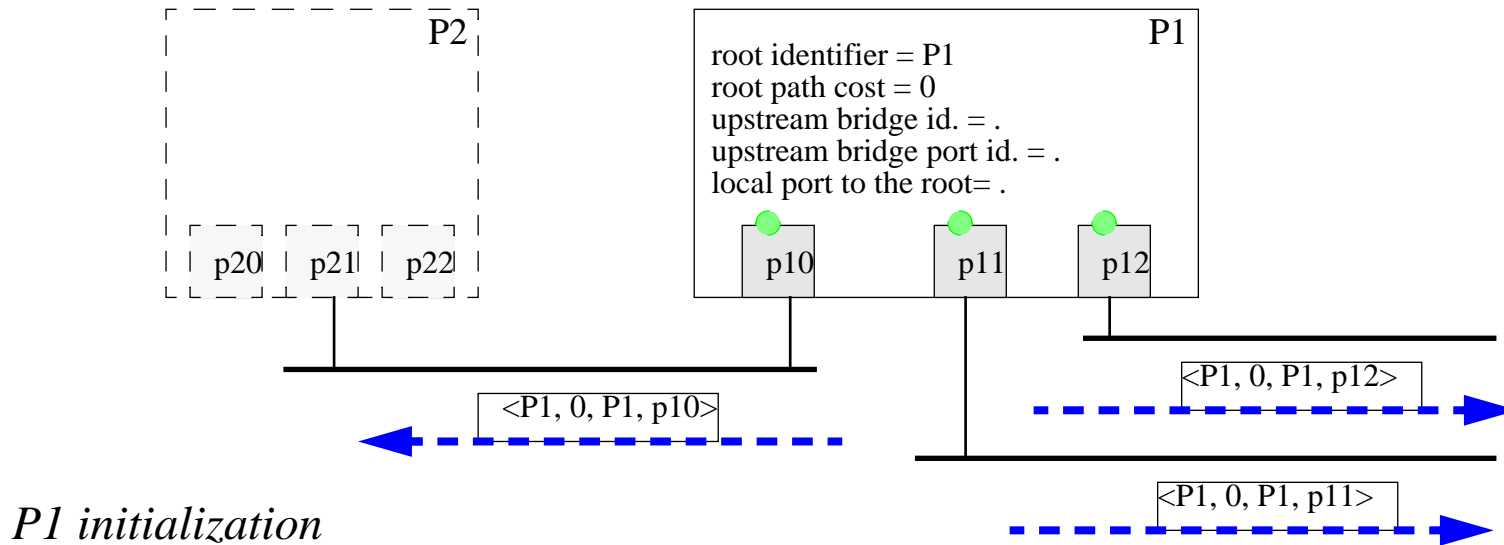
5.7. Spanning tree algorithm

5.7.1 Initialization

On initialization, every bridge chooses itself as the root and broadcasts this value in a message

Message format:

- $\langle \text{root address, path length, upstream bridge, bridge port} \rangle$.

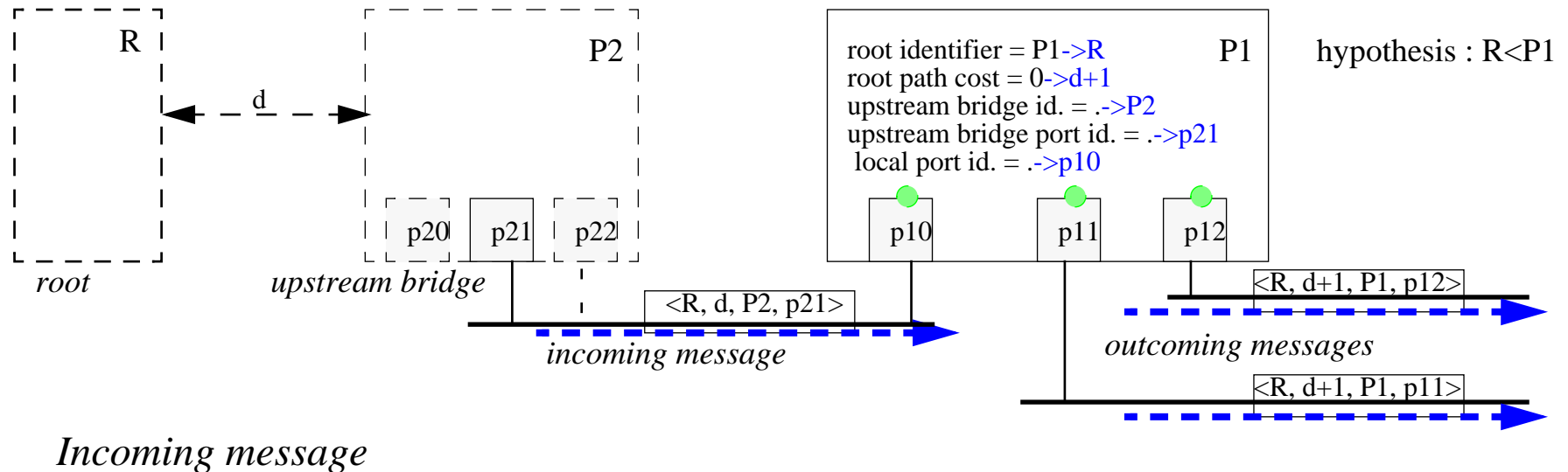


5.7.2 Message processing

Each bridge memorizes the current lowest value.

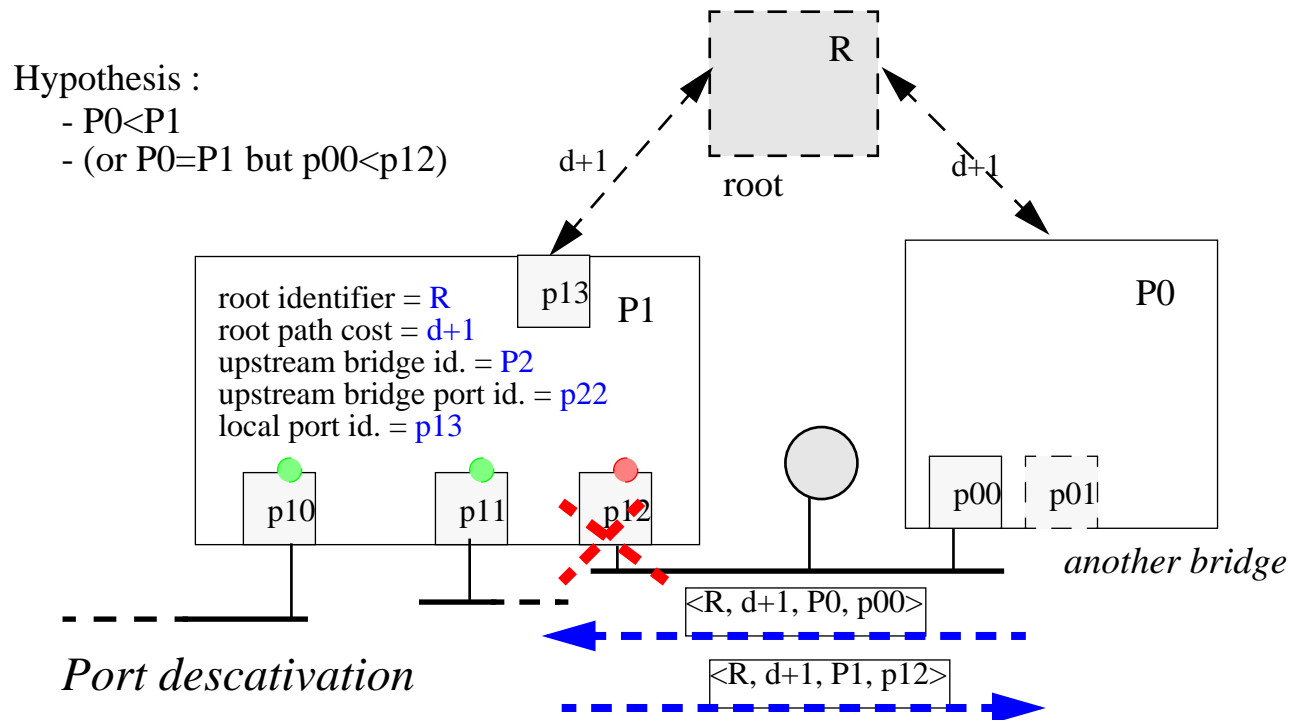
When a bridge receives a message, it compares the local value and the message value (with an incrementation of the path length). If the message value is lower:

- the message value is writing in the bridge memory
- a new message is broadcast with the appropriated value, except to the upstream port



5.7.3 Port desactivation

Any bridge port which receives a message which has a **higher** value than the value of the local bridge and which has a **lower** value if the path incrementation is not done, is disactivated.



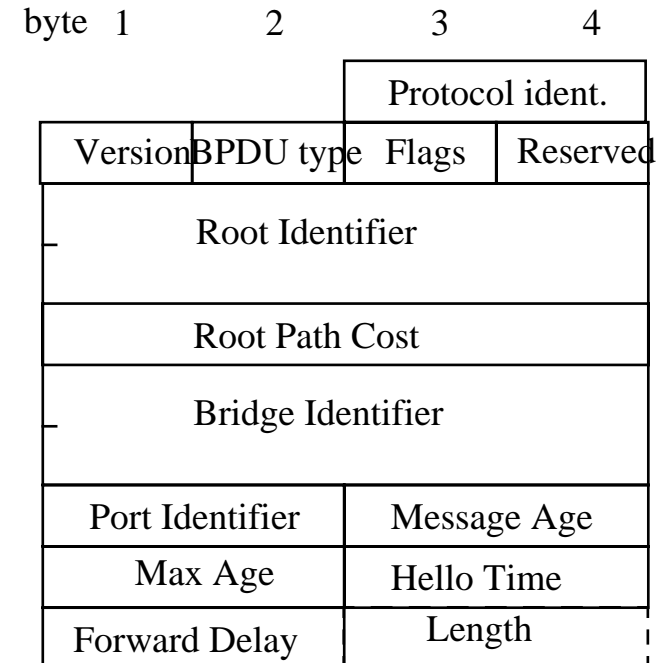
When a bridge has several ports on the same LAN, all the ports on the same LAN are desactivated except the port with the lowest identifier.

5.7.4 Tree monitoring

- The port desactivation and the broadcasting of new information are delayed:
 - default value of Forward Delay field = 15 s
 - to allow bridges to reach stable states, and messages to reach all network bridges
- Bridge failures have to be monitored
 - root sends messages periodically (
 - . default value of Hello Time field = 4 s
- each message is stamped
 - Message Age field
 - older message is discard
- Sent information have a limited lifetime:
 - old information are discarded
 - default value of Max Age field = 20 s.

5.8. Message format

- . Protocol Identification (= 0)
- . Protocol Version (= 0, or 1)
- . BPDU type (= 0)
- . Flags
- . Root Identifier
- . Root Path Cost
- . Bridge Identifier: upstream bridge to the root
- . Port Identifier: output port of the upstream bridge
- . Message Age [unit: $1/256^{\text{th}}$ s]
- . Max Age [unit: $1/256^{\text{th}}$ s]
- . Hello Time [unit: $1/256^{\text{th}}$ s]
- . Forward Delay [unit: $1/256^{\text{th}}$ s]
- . Length (for version 1 message format BPDU)

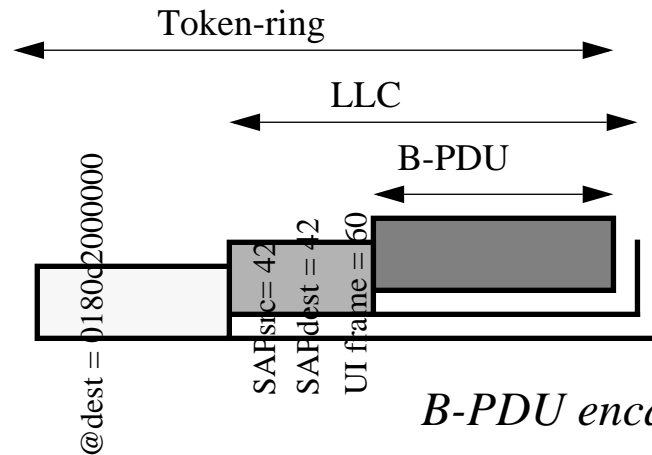


Bridge-PDU

5.9. Frame encapsulation example

B-PDU message could be encapsulated by LLC and Token-Ring frames:

- MAC address for Bridge group: 0180c2000000
- Bridge management SAP: 42



B-PDU encapsulation in Token ring frame

5.10. Conclusion

Transparent bridging algorithm is simple, and transparent to host.

- **easy** to install and maintain

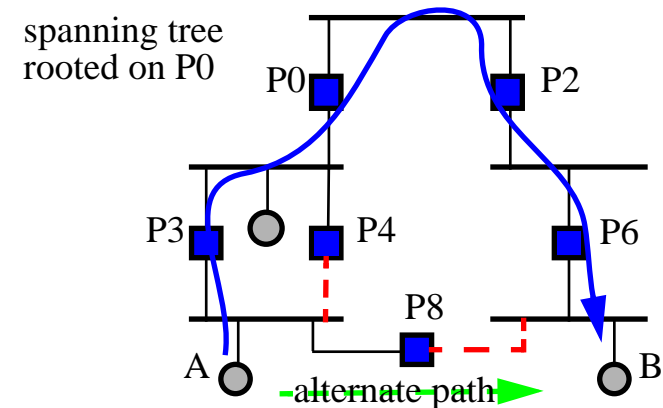
Requires **Spanning Tree** algorithm.

Transparent bridging proposes

- by default **flooding** process,
- then efficient forwarding owing to **backward learning**.

Transparent bridging drawbacks:

- root and links leading to the root are **congestion points**
- **under-use of redundancy** links and bridges
- **promiscuous mode burdens** the bridge



6. Source Routing

6.1. Introduction

Proposed by IBM for Token Ring networks.

- Optimum
 - Use the shortest path between hosts
- Efficiency
 - Alleviated bridge burden
 - ... processing is move to hosts

 The routing is done by the Source

6.2. Source Routing principle

Identification:

- a number, unique to the network, identifies each LAN
- a number, unique to every LAN to which it is connected, identifies each bridge

Each host has a path cache:

- the path description to every destination, the host is interested on.
- path = list of entries <LAN identifier, bridge number>
 - . LAN identifier: 12 bits
 - . bridge number: 4 bits
- bridges connected to different LAN may have the same number
- Bridge port identifier: <LAN identifier, bridge number>

Bridged frames are marked: most significant bit of source address = 1

6.3. Source Routing bridging

When a **host sends** a data frame (frame preparing),

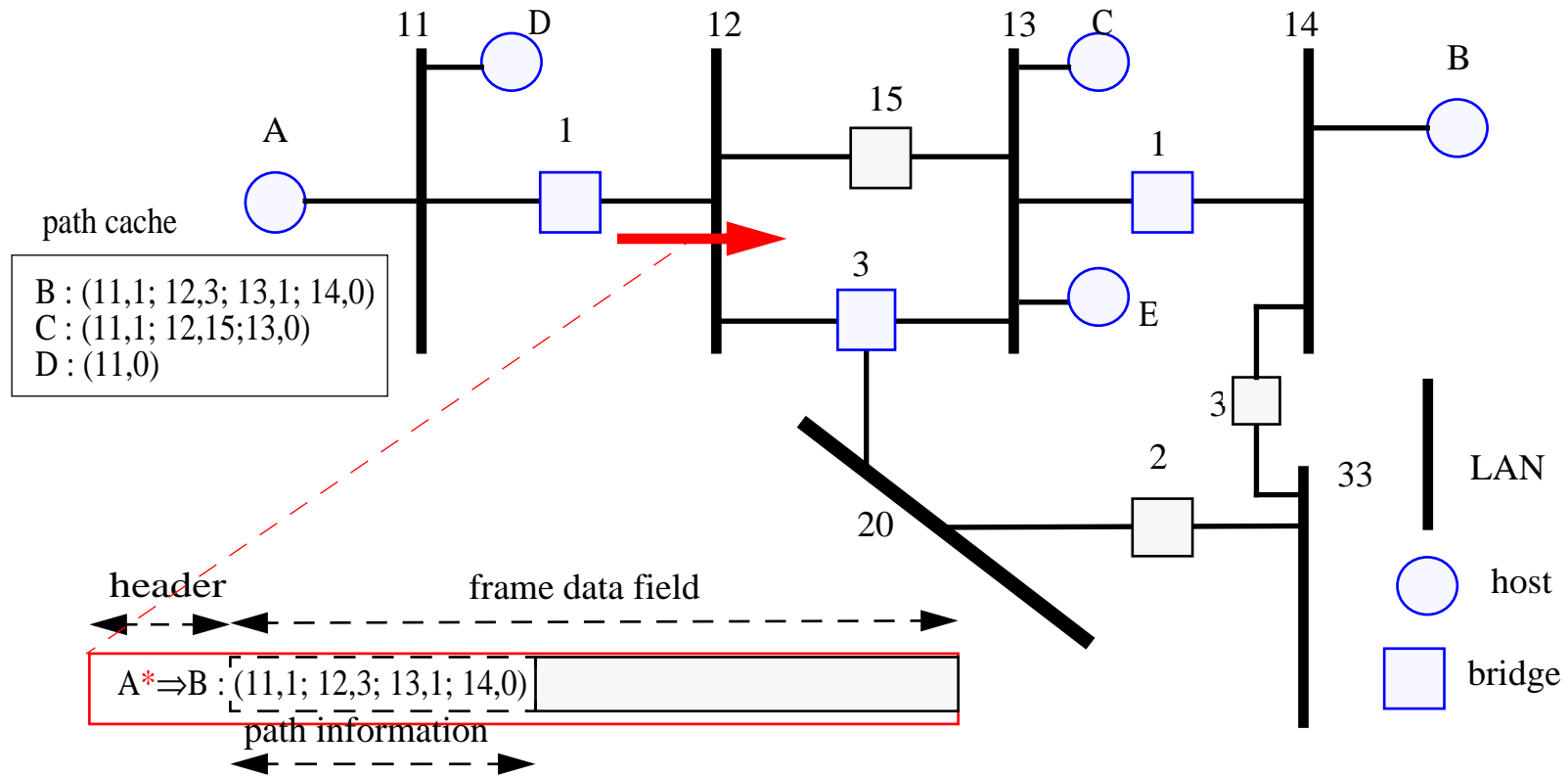
- path lookup into path cache
 - . entry hit: the path description is placed into the header frame, the frame is sent
 - . entry miss: path exploration process to the destination is launched

When a **bridge receives** a data frame (just following the path instruction),

- Bridges process the marked frames only.
 - path extraction from the frame header
 - if the path list holds an entry with the input LAN
 - . the frame is forwarded to the entry bridge
 - . else the frame is discarded

6.4. Source Routing example

A sends a frame to B:



6.5. Path explorer

The source host broadcasts a Path Explorer message when it looks for a path to a destination

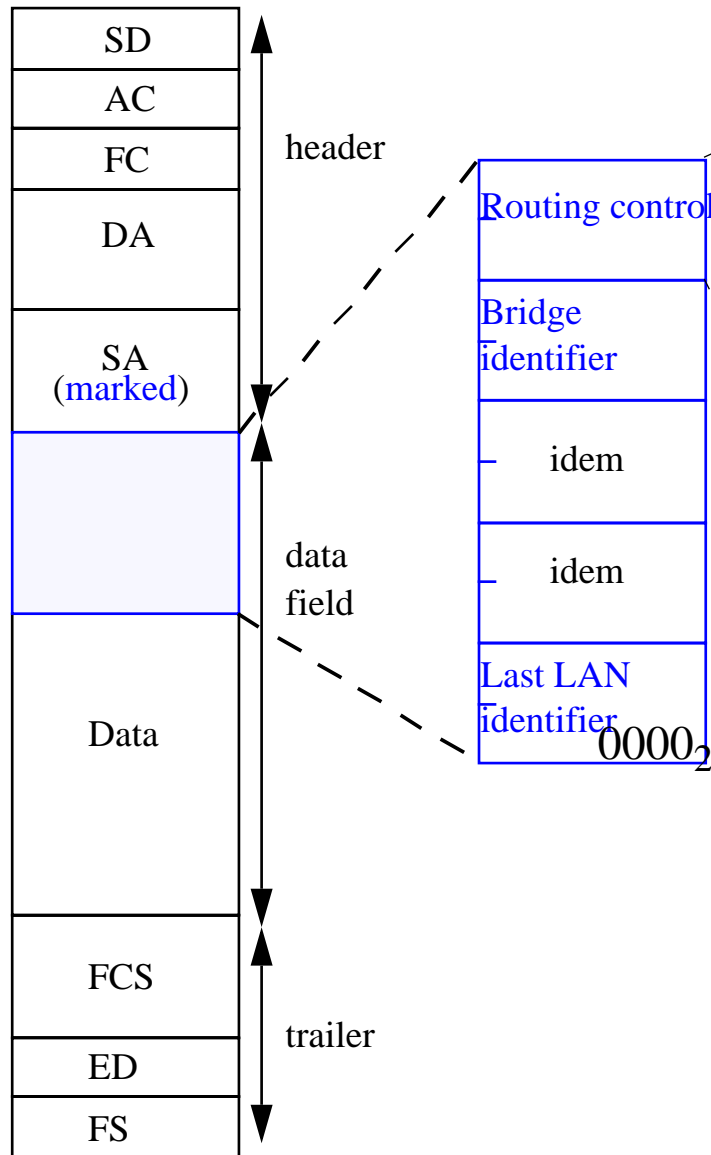
- the destination address field of the frame is the address of the destination
- the path description in the frame is restricted to the output LAN identifier
- **Bridges, when receiving a Path Explorer** message
 - lookup for the bridge port identifier in the path description inside the message
 - . a miss: the bridge broadcasts the message, appending the output bridge port identifier to the path description of message (output LAN id. + bridge number)
 - . a hit: the message is discarded
- **Destination, when receiving a Path Explorer** message,
 - sends the message (with direction bit inversed) back to source
- **Source, when receiving a Path Explorer** message, and when the new path is shorter than the current one,
 - inserts the path description into the path cache
 - sends the waiting frames to the destination

6.6. Path explorer types

2 path explorer types:

- **all path explorer:**
 - message broadcasting = message flooding
 - . forwarding to every output port except the input port
- **spanning tree explorer**
 - source to destination message broadcasting = ST broadcasting
 - . forwarding to every output port belonging to the ST tree, except the input port
 - when a spanning tree explorer message is received:
 - . **host** discards the message, if the input port has not been validated during the ST building
 - . **destination** sends back an All Path Explorer message to the source

6.7. Frame format



Routing control (16 bits)

- Frame type (3 bits) :
 - . 000₂ = Data frame
 - . 010₂ = Spanning Tree explorer frame
 - . 100₂ = All Path explorer frame
- List length in bytes (5 bits)
 - . maximum : 15 bridges on each LAN
- List reading direction (1 bit)
 - . 1 = inverse direction
- MTU (3 bits):
 - . coding of the most current lengths
 - . e.g. 011₂ = 2052 bytes

6.8. Management

- Path monitoring
 - a timer is associated to every entry of the path cache
 - when the time expires the entry is discarded
- Message loss
 - message is periodically sent until
 - . either a response is obtained
 - . or the number of consecutive backoff exceeds the maximum attempt number
- Free optimization
 - when source looks for a path to a destination
 - the destination freely learns the path to the source
 - . when the path explorer message reaches it

6.9. Conclusion

Hosts learn paths to destinations through Path Explorer process

Hosts select the best path to each destination

Source puts into the frame the path they must follow

Bridges follow the instructions find inside the frame

Caching optimization

Broadcasting is not very scalable: broadcast storms

7. Conclusion

Interconnection function could be find in any layer

- in many types of equipment: hub, bridge, router, gateway, etc.

Forwarding + translation + tunneling process

Source routing versus Transparent bridging

- bridge versus host oriented
- Complementary approaches:
 - when the frame is marked
 - . Source routing
 - . else Transparent bridging

Bridging is adapted to LAN interconnection (natural broadcasting)

- ✔ multi protocol
- ✔ easy network management
- ✘ frame length adaptation (segmentation)
- ✘ no scalability
- ✘ no error management