Repeaters, Hubs, Bridges, and Routers

Class 5

LAN Extension and Switching

Topics

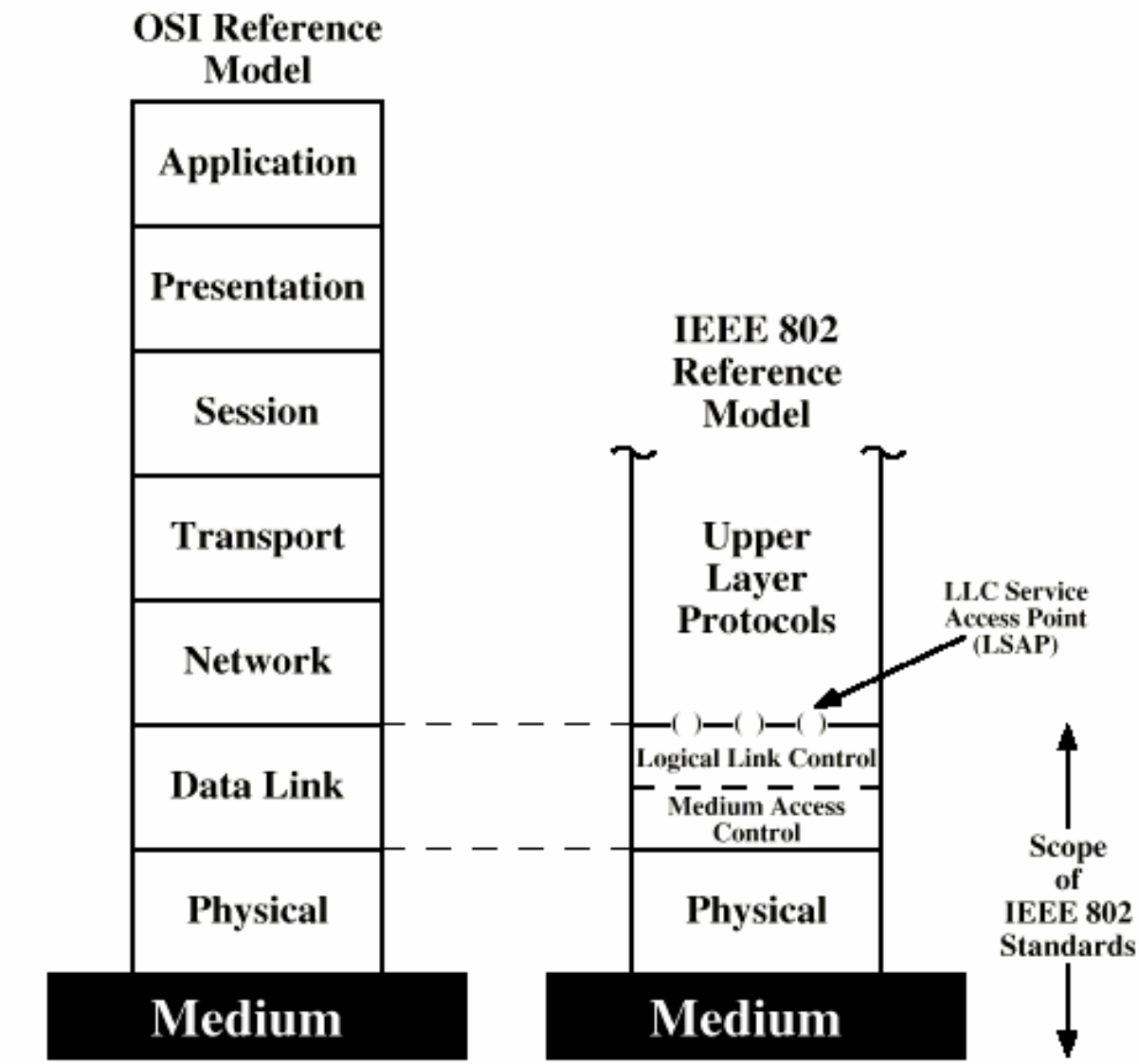
- LAN Protocols
- Hubs/Repeaters
- Bridges
- Routers

LAN Protocols – Basics

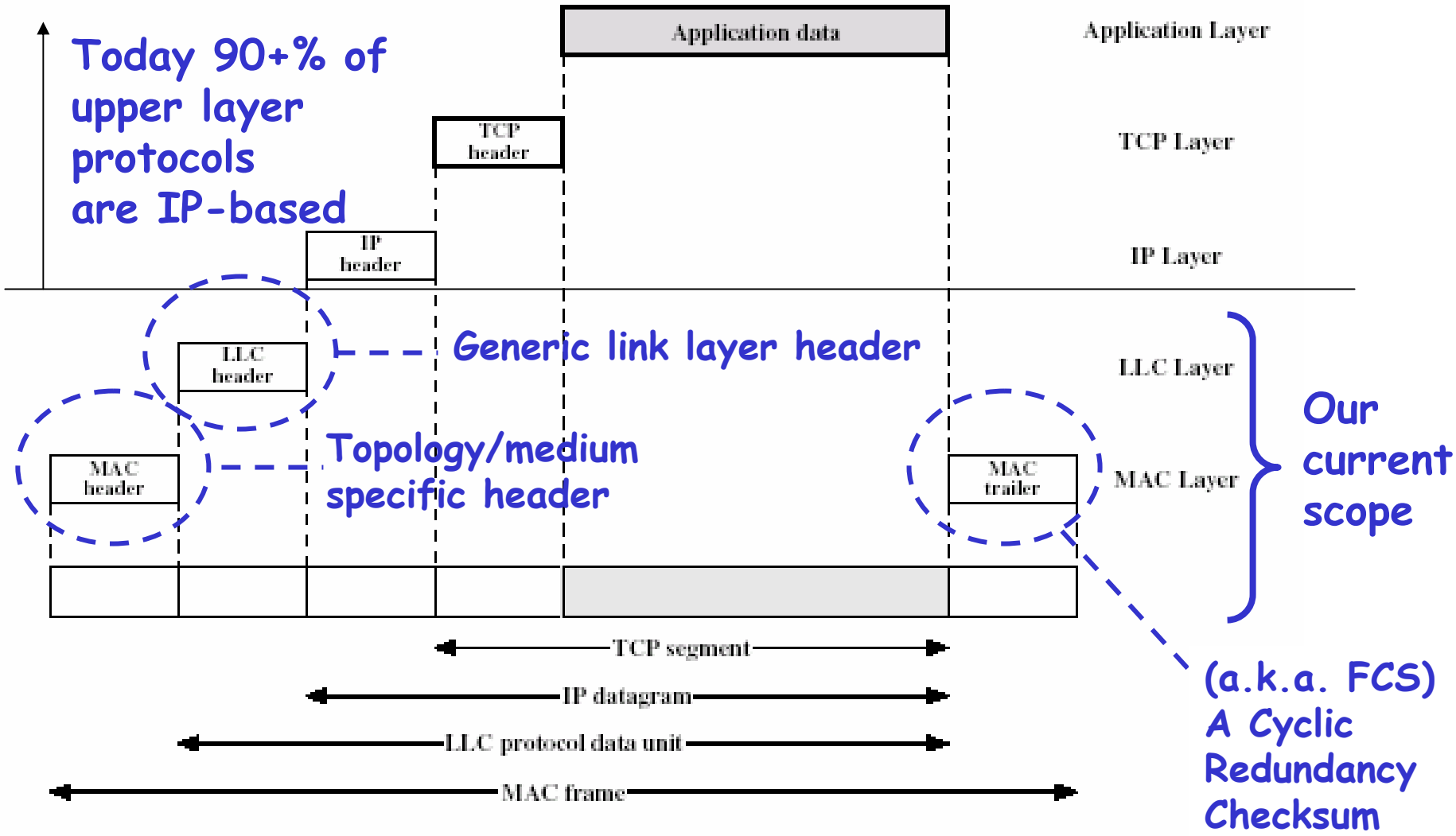
IEEE 802 reference model

- Logical Link Lontrol (LLC)
  - Interfaces to higher levels
  - Handles *flow* and *error* ***control***
- Media Access Control (MAC)
  - Governs access to transmission medium
  - Assembles data into frames with *address* and *error detection* fields
  - Disassembles frame – Address recognition / *Error* ***detection***
- Physical
  - Encoding/decoding
  - Preamble generation/removal (synchronization)
  - Bit transmission/reception
  - Transmission medium dependent

LAN Protocols – ISO Reference



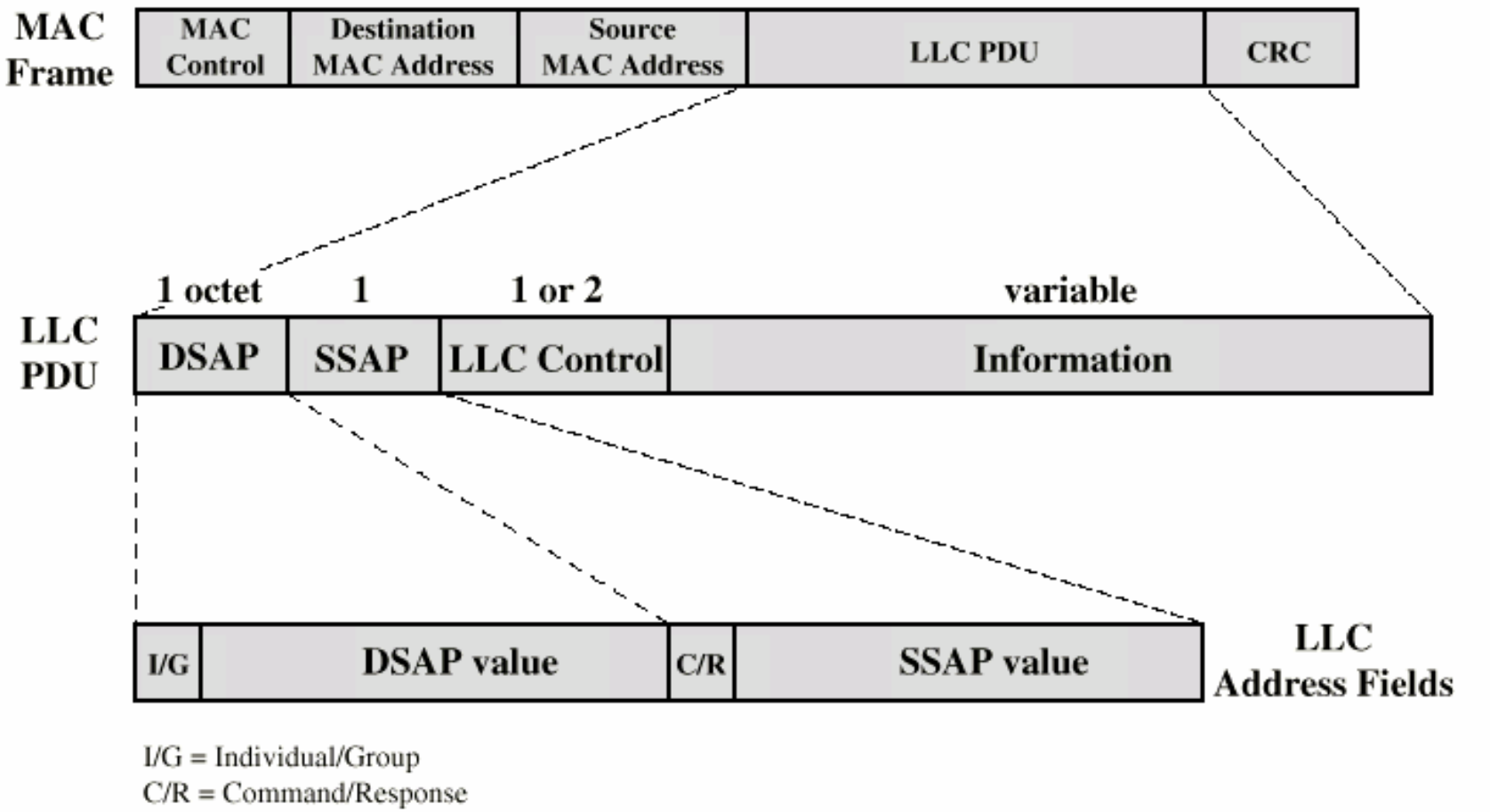
LAN Protocols – In Context



Media Access Control – Operation and Frame Format

- Operation
  - MAC layer ***receives data*** from LLC layer
  - MAC layer ***detects errors*** and discards frames
  - LLC ***optionally retransmits*** unsuccessful frames
- Frame format
  - MAC control
  - Destination MAC address
  - Source MAC address
  - LLC data
  - CRC

MAC and LLC – Typical Frame Formats



LAN Extension – Basics

Challenge – how do we add more nodes and increase distance between nodes on a LAN

Solution

- ***Expand LAN*** beyond a single LAN (Segment)
- ***Interconnect*** to other LANs/WANs

Method

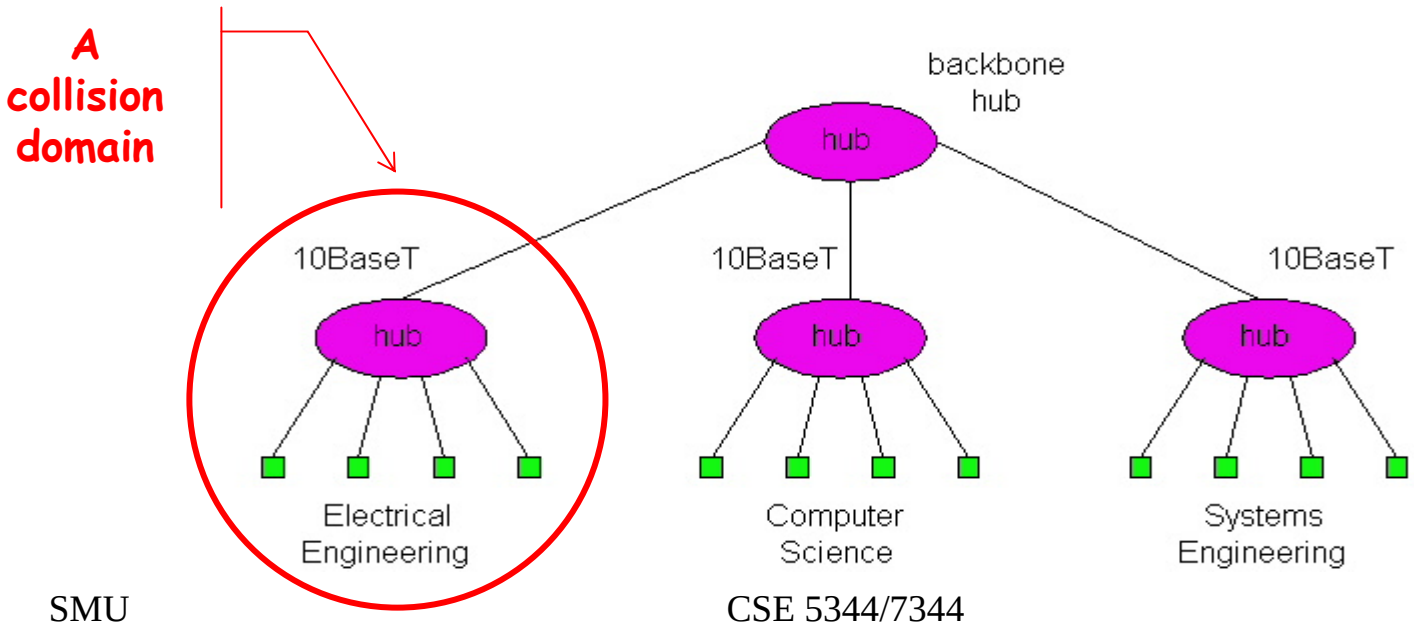
- Use bridge or router
- Bridge is simpler
  - Connects *similar* LANs
  - *Identical protocols* for physical and link layers
  - Minimal processing (in simple case)
- Router is more general purpose
  - Interconnects *various* LANs and WANs

Interconnecting LAN segments

- Hubs / Repeaters → Physical layer
- Bridges → MAC layer
- Routers → Network layer

Interconnecting with Hubs

- **Physical layer device**
- *Extends max distance* between nodes
- Backbone Hub (Repeater) *interconnects* **LAN segments**
- Can't interconnect 10BaseT & 100BaseT
- The individual ***segment collision domains*** become ***one large collision domain***
  - if a node in CS and a node in EE transmit at same time: collision

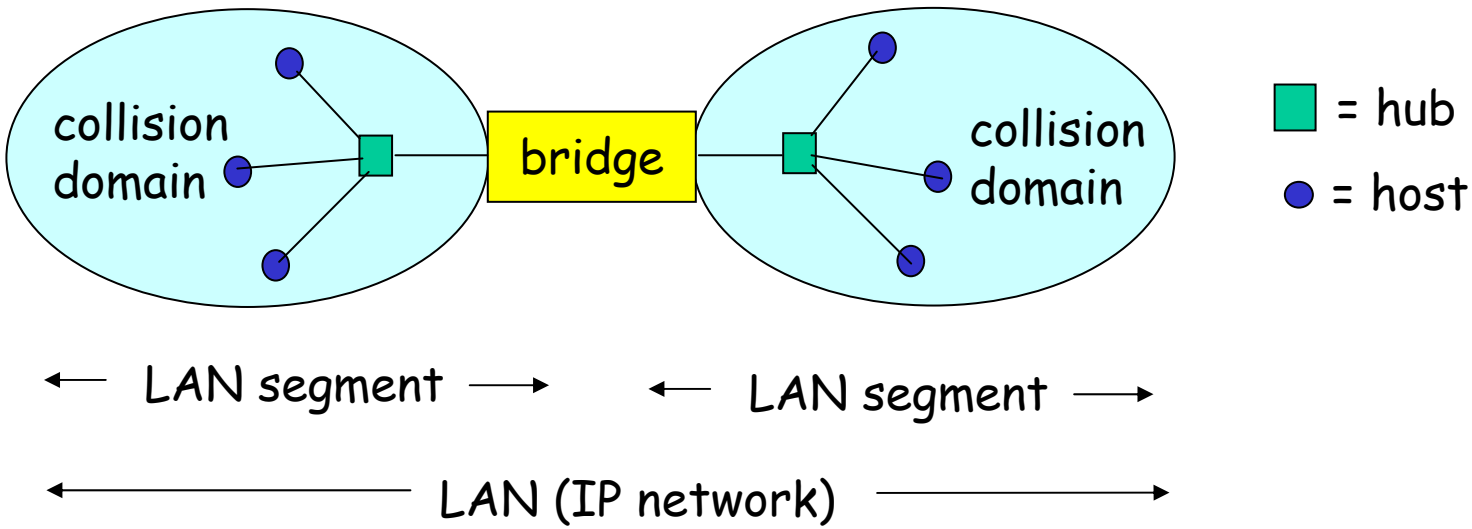


Bridges

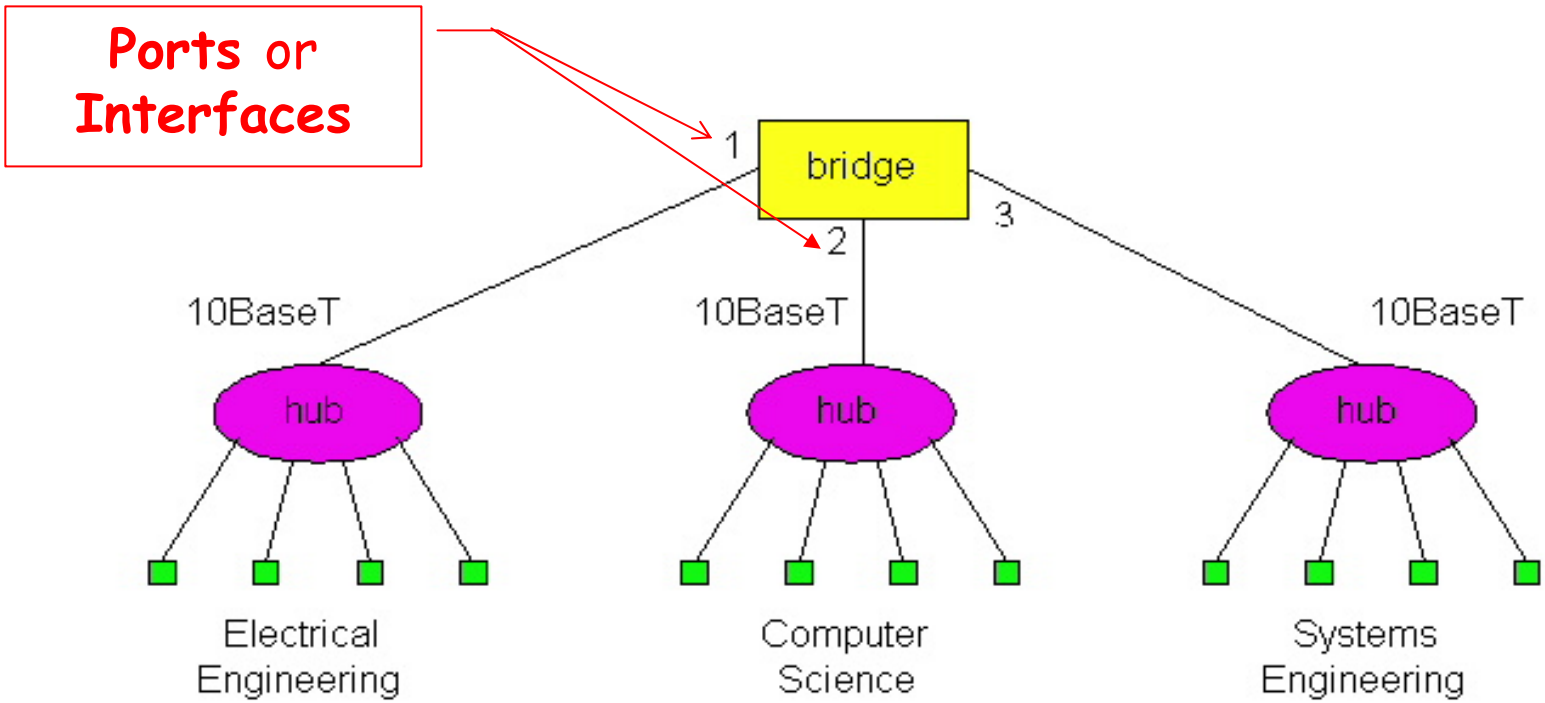
- **MAC layer (Data Link Layer – L2) device**
  - stores and forwards Ethernet frames
  - examines frame header and ***selectively*** forwards frame based on **MAC dest** address
  - when frame is to be forwarded on a segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of bridges
- plug-and-play, self-learning
  - bridges do not need to be configured

Bridges: traffic isolation

- Bridge installation breaks LAN into **LAN segments**
- Bridges *filter* packets:
  - same-LAN-segment frames are not (usually) forwarded onto other LAN segments
  - segments become ***separate*** collision domains



Forwarding



How does the bridge determine to which LAN segment to forward frame?

- Looks like a routing problem...

Self learning

- A bridge has a Bridge Table
- Entry in Bridge Table:
  - (Node LAN Address, Bridge Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- Bridges *learn* which hosts can be reached through which interfaces
  - when frame received, bridge "learns" location of sender: incoming LAN segment
  - records sender/location pair in Bridge Table

Filtering/Forwarding

When bridge ***receives*** a frame:

index Bridge Table using MAC **dest** address

if entry is found for destination

then{

if dest is on segment from which frame arrived

then *drop* the frame

else *forward* the frame on interface indicated

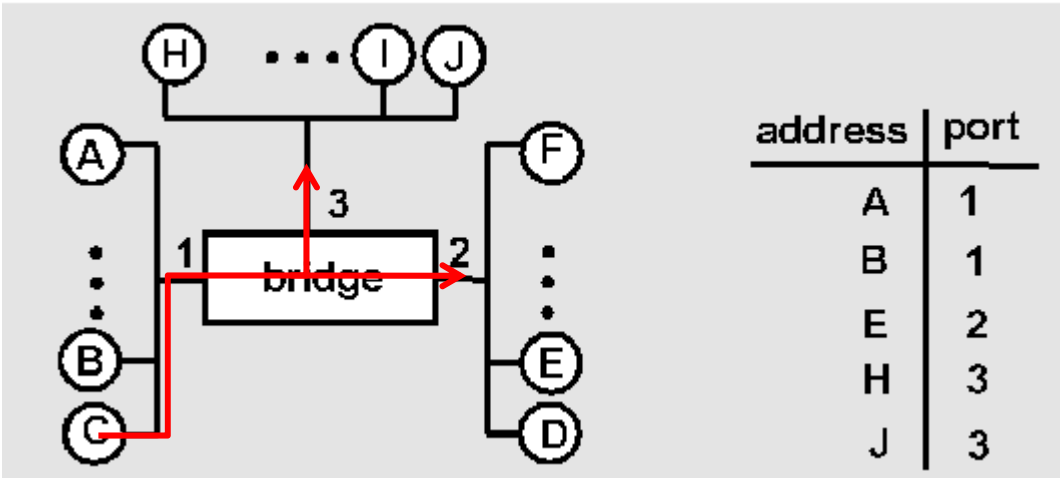
}

else *flood*

forward on all but the interface on which the frame arrived

Bridge example

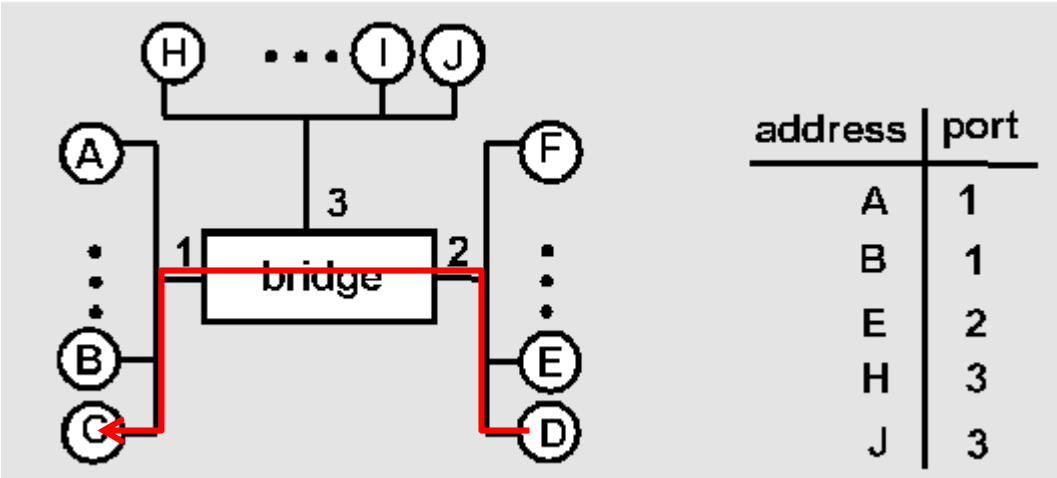
Suppose C sends frame to D and D replies back with frame to C.



address	port
A	1
B	1
E	2
H	3
J	3

- Bridge receives frame from C
  - notes in Bridge Table that C is on interface 1
  - because D is not in table, bridge sends frame to interfaces 2 and 3
- frame received by D

Bridge Learning: example



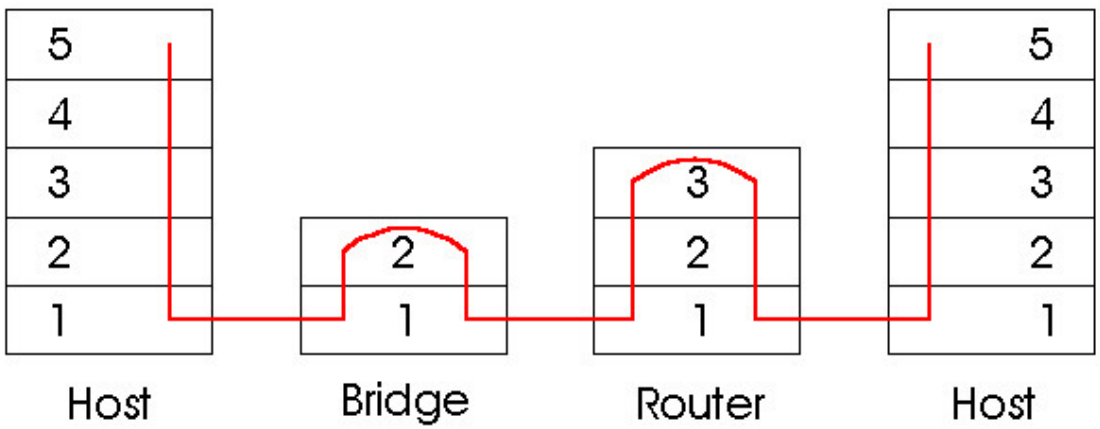
- D generates frame for C and sends it
- bridge receives frame
  - notes in Bridge Table that D is on interface 2
  - bridge knows C is on interface 1, so *selectively* forwards frame to interface 1

Some Bridge Features

- Isolates collision domains resulting in higher total max throughput
- Limitless number of nodes and geographical coverage
- Can connect different Ethernet types
- Transparent ("plug-and-play"): no configuration necessary

Bridges vs. Routers

- Both are Store-and-Forward devices
  - ***routers***: network layer devices (examine ***network layer*** **headers**)
  - ***bridges*** are **link layer** devices
- Routers maintain ***Routing Tables***
  - implement routing algorithms
- Bridges maintain ***Bridge Tables***
  - implement filtering, learning and spanning tree algorithms



Routers vs. Bridges

Bridges + and -

+ Bridge operation is simpler requiring less packet processing
+ Bridge tables are self learning
- All traffic confined to spanning tree, even when alternative bandwidth is available
- Bridges do not offer protection from broadcast storms

Routers vs. Bridges

Routers + and -

- \+ arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
- \+ provide protection against broadcast storms
- \- require IP configuration (not plug and play)
- \- require higher packet processing

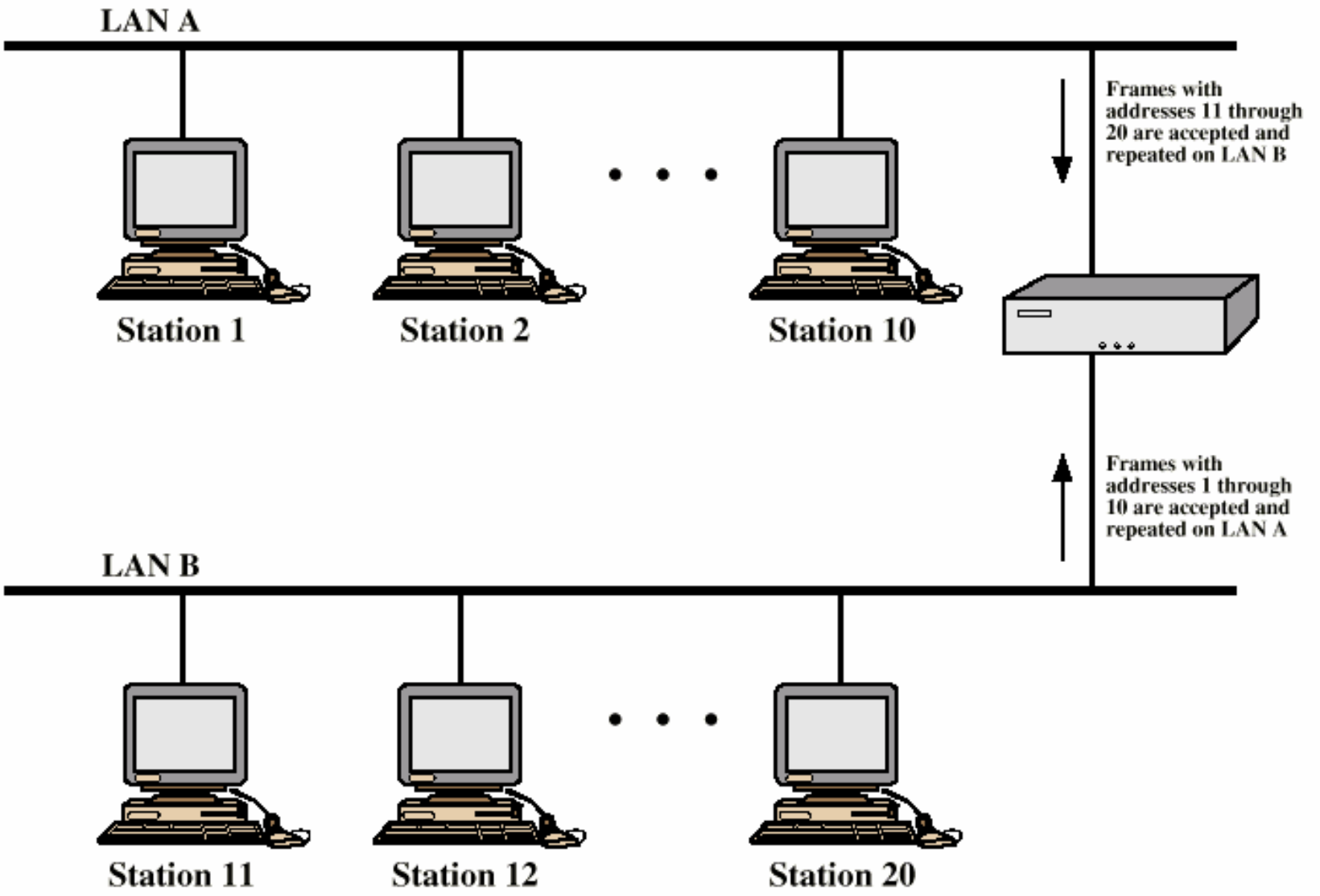
- bridges do well in small networks (few hundred hosts)
- *routers* are used in *large* networks (thousands of hosts)

Summary comparison

	hubs	bridges	routers
traffic isolation	no	yes	yes
plug & play	yes	yes	no
optimal routing	no	no	yes

More Detailed Focus on Bridge Routing

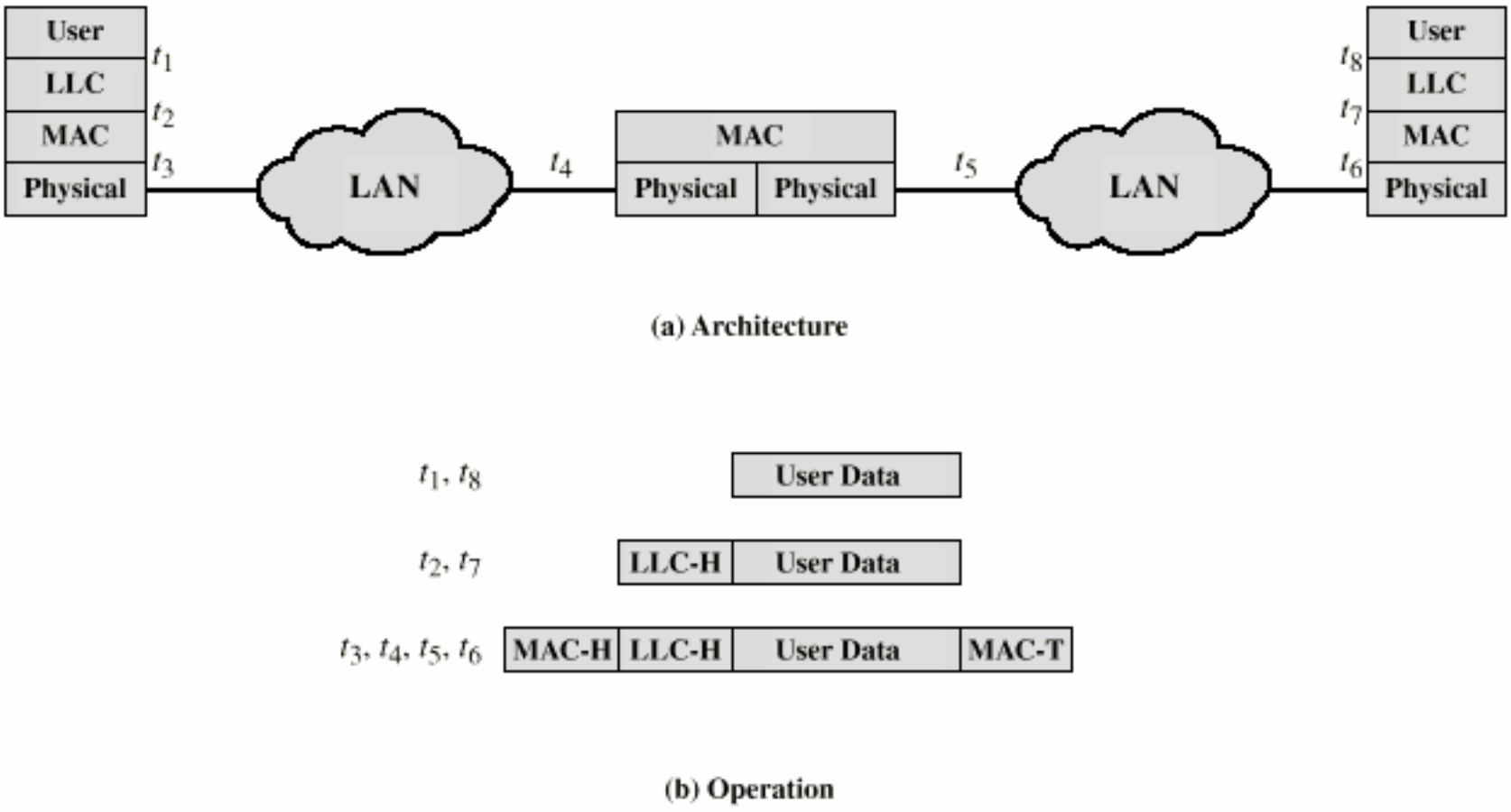
Bridges – Operation



Bridges – Design Aspects

- ***No modification*** to content or frame format
- ***No encapsulation*** – Exact bitwise copy of frame
- ***Minimal buffering*** to meet peak demand
- IEEE 802.1D
- MAC level – Station address is at this level
- Does not need LLC layer – relays MAC Frames
- Contains *routing* and *address* intelligence
- May connect more than two LANs
- Bridging is *transparent to stations*

Bridges – Connection of Two LANs



Routing with Bridges

- Fixed routing
- Spanning tree algorithm (transparent)
- Source routing

Bridges – Fixed Routing

- Complex large LANs need alternative routes
  - Load balancing
  - Fault tolerance
- Bridge must *decide whether* to forward frame
- Bridge must *decide on which* LAN to forward frame

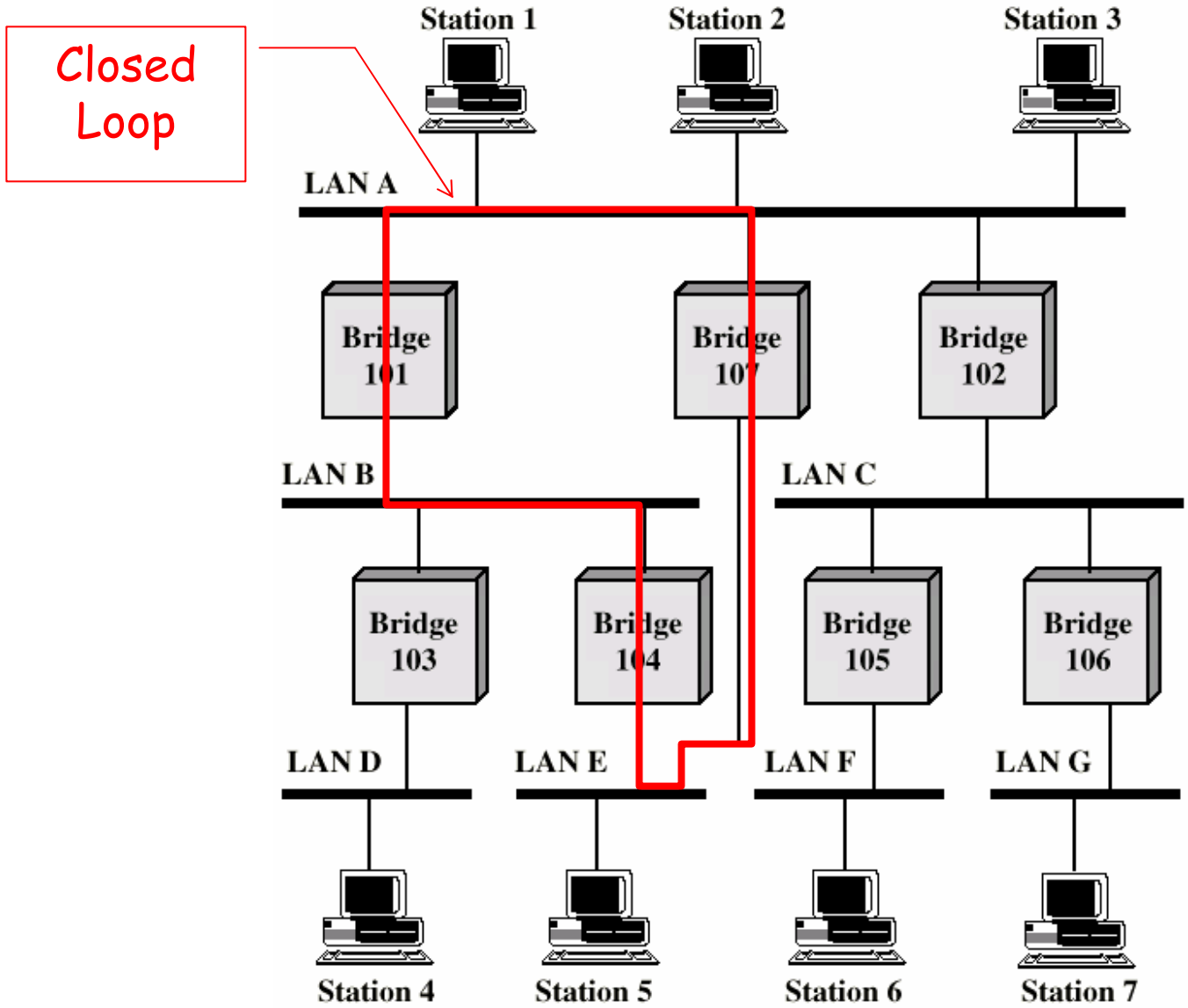
Fixed Routing (cont)

Routing is selected for ***each source-destination pair*** of LANs

- Done during configuration (manual)
- Usually least-hop route
- Only changed when topology changes

- Based on a central routing matrix
- Simple, minimal processing
- Suitable for small LANs
  - large internets need more dynamic strategies

Bridges – Multiple LANs



Bridges – Spanning Tree

- Bridge automatically develops routing table
- Automatically updates in response to changes

- ***Frame forwarding***
- ***Address learning***
- ***Spanning tree Algorithm / Loop resolution***

Bridges – Frame Forwarding

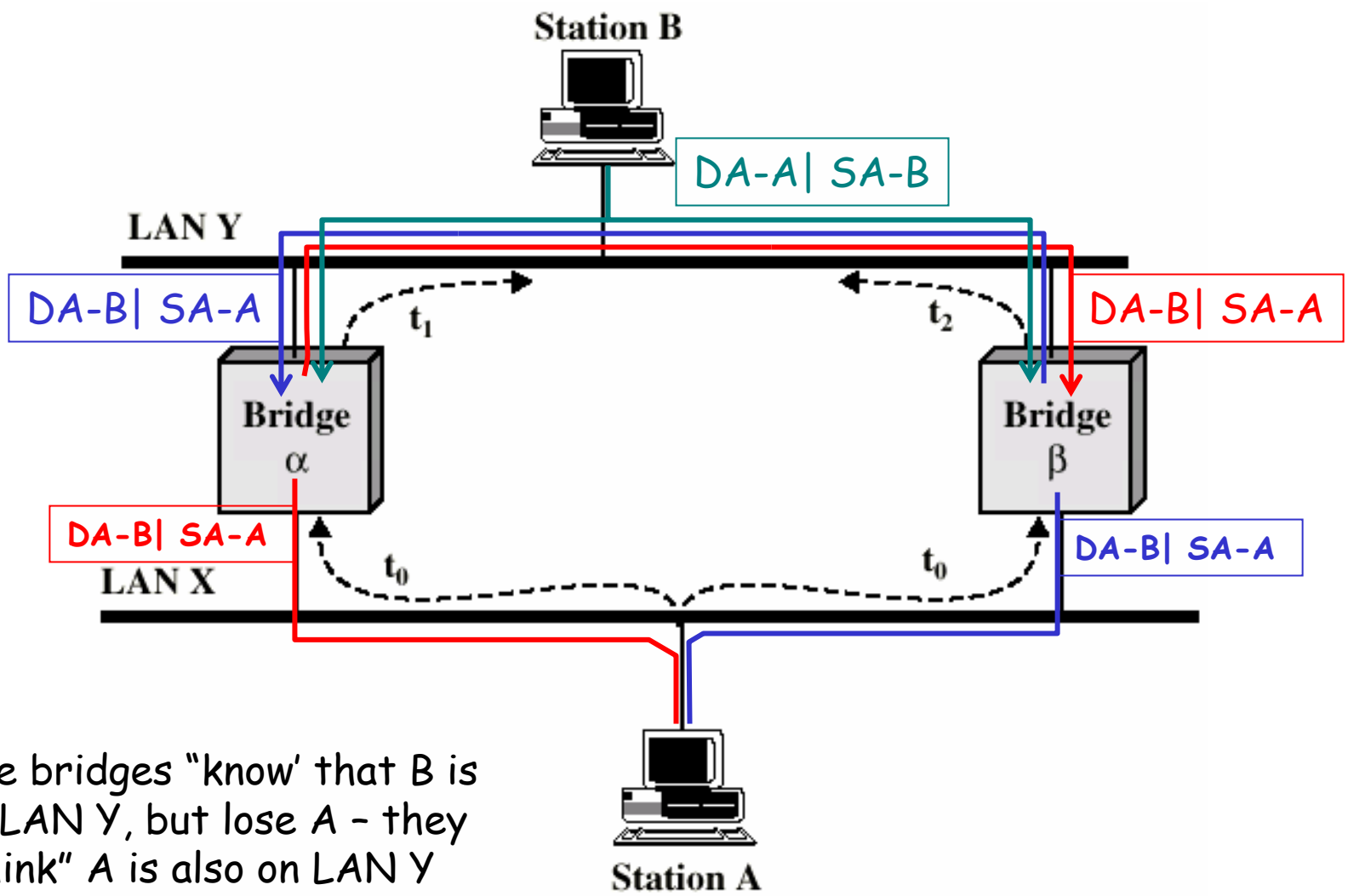
- Maintain forwarding database for each port
  - List station addresses reached through each port
- For a frame arriving on port X:
  - Search forwarding database to see if MAC address is listed for any port except X
  - If address not found, forward to all ports except X
  - If address listed for port Y, check port Y for blocking or forwarding state
    - Blocking prevents port from receiving or transmitting
  - If not blocked, transmit frame through port Y

Address Learning

Automatic mechanism to update the database

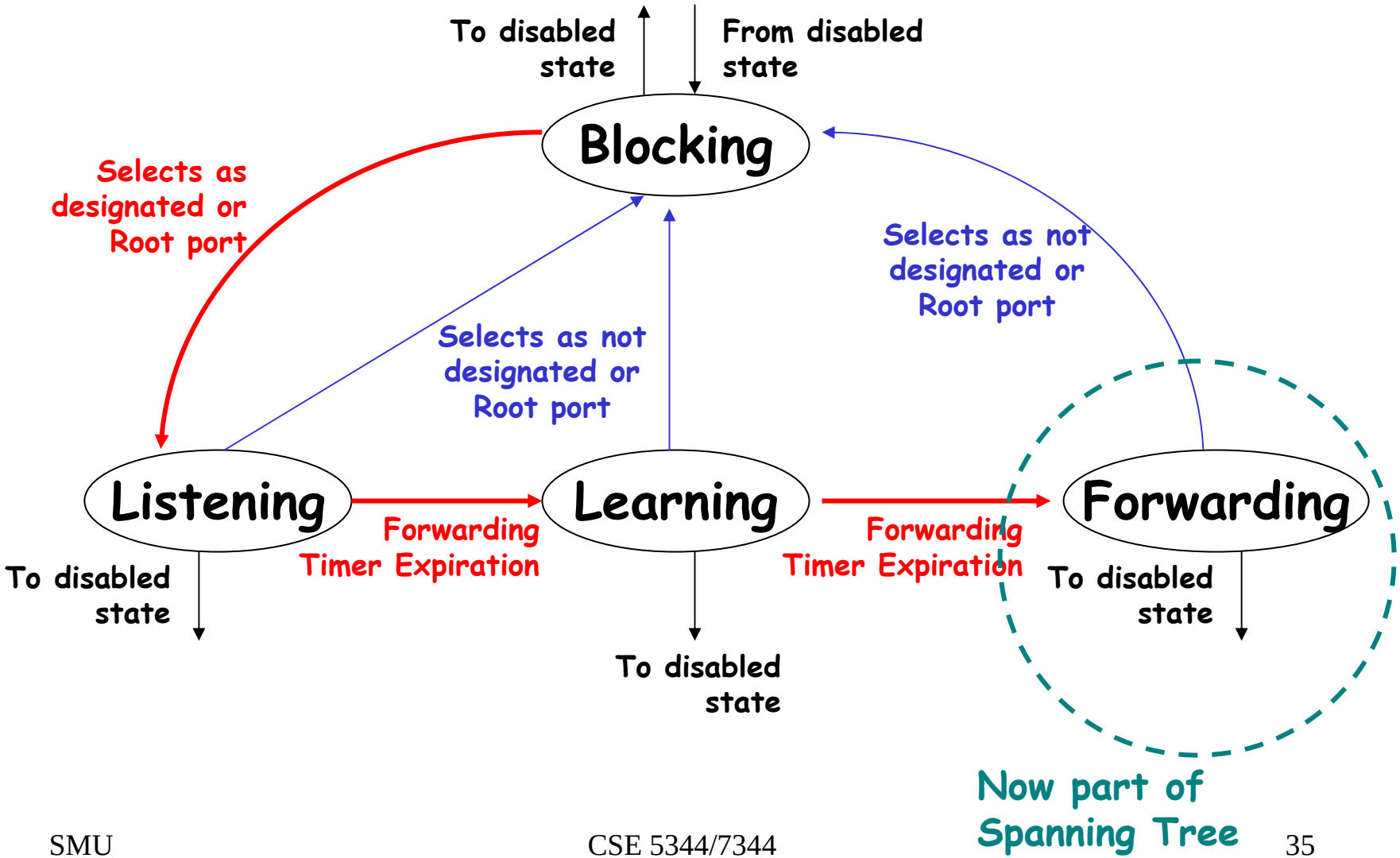
- Frame arrival provides ***direction*** information
  - When frame arrives at port X, it has come from the LAN attached to port X
  - Use the source address to update forwarding database for port X to include that address
- Each time frame arrives, source address checked against forwarding database
- Uses ***aging timer*** to allow for topology changes
- When the timer expires, the entry is purged

Bridges – Looping



The bridges "know' that B is on LAN Y, but lose A – they "think" A is also on LAN Y

Basic Operations



Bridges – Spanning Tree Algorithm

Designed to handle bridge loops – create a ***spanning tree*** that ***maintains connectivity*** but ***contains no closed loops***

- *Each bridge* is assigned *unique identifier*
- Exchange between bridges establishes the spanning tree
- Costs are associated with each port/link
- Special group MAC address

Basic Operations

- Map the contents of the incoming frame into an outbound frame
- Each bridge attachment to a LAN is called a port
- A bridge has *n* MAC addresses, one for each port

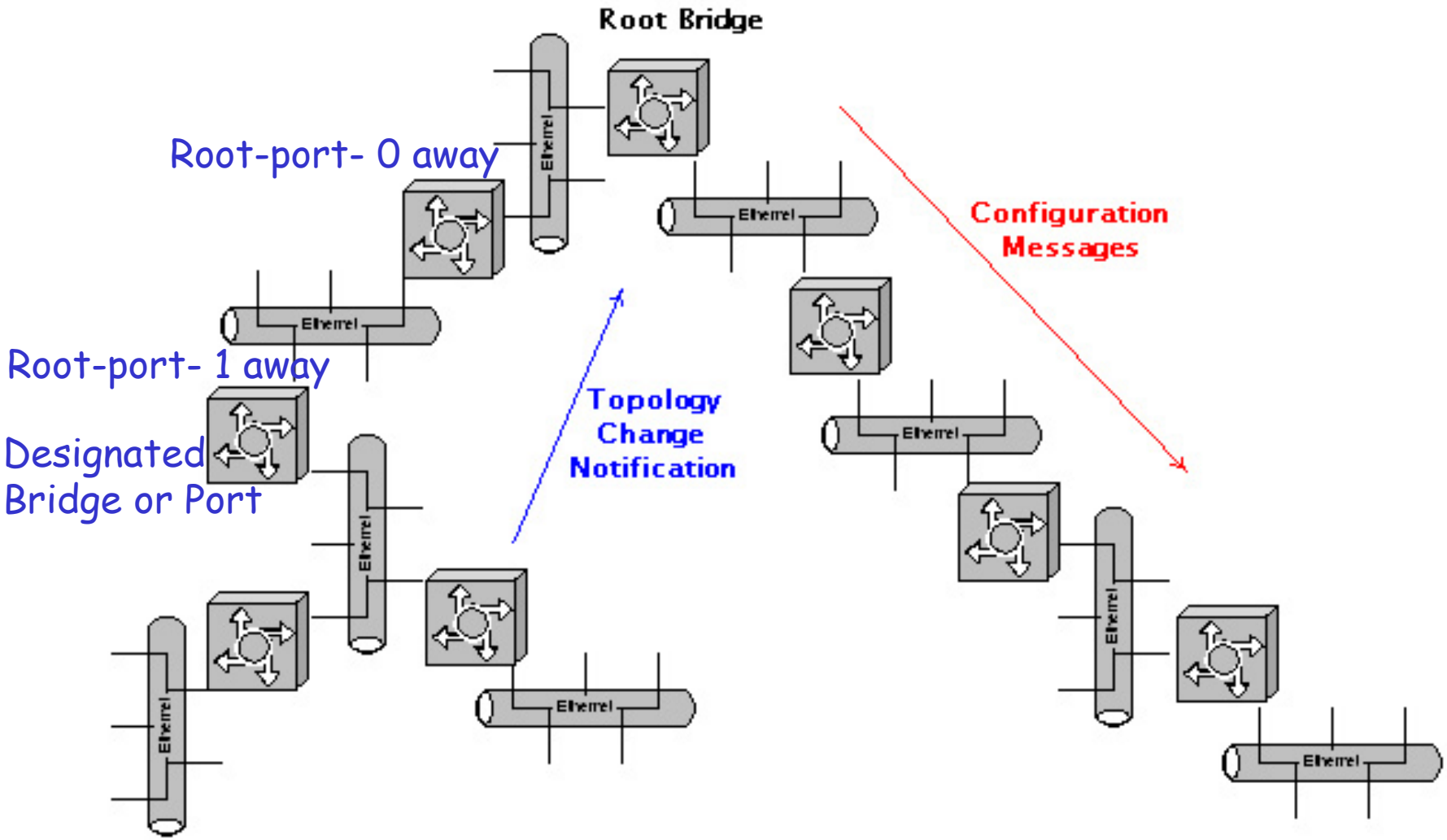
Concepts used in the Algorithm

- Root bridge
- Root path cost
- Root port
- Path cost
- Designated bridge/port

Construction of ST

- Determine the root bridge
- Determine the root port of all other bridges
- Determine the designated port of each LAN
- Bridge activities can be represented using a state diagram (as shown earlier)

ST Limitation



ST Limitations (cont'd)

- Vertical Idiosyncrasy
  - Once bridge detects failure it generates TCN (Topology Change Notification)
  - Has to wait for CM (Configuration Message) before it can change port states and discard database entries
  - Slow convergence (~60 s)
  - No authentication

Multiple Spanning Trees (MST)

- Provision multiple spanning trees including redundant trees
- No need to wait for roundtrip delay along the depth of the tree
- Faster convergence
- Potential for traffic engineering

Source Routing

- The *source node determines* the *path*
- MACs modified to include routing information
- Bridges need not maintain routing tables
- Bridge decides whether or not to forward a frame
  - Bases decision on info contained in frame
  - If not forwarded, frame is discarded
- Bridge needs to know
  - Its own unique identifier
  - Identifier of LAN to which it is attached

Source Routing

Addressing modes → Source indicates which type of routing to use:

Src Knows Dst Address
- Null
- Non-broadcast

Src Discovers Dst Address
- All-routes broadcast
- Single-route broadcast

Route Selection → by Src

- Manual
- Designated routing nodes
- Dynamic route discovery (802.5)

Option-1

- Src sends *all-routes broadcast* **request** frame to Dst
- Dst sends back a *non-broadcast* ***response*** on *each* of the *discovered* channels

Option-2

- Src transmits a *single-route broadcast* frame to Dst
- Dst sends back an *all-routes* response frame

Spanning Tree vs. Source Routing

- Philosophy
- Quality of routes
- Use of available bandwidth
- Complexity route discovery algorithm
- SRT (Source routing transparent) bridges

Spanning Tree vs. Source Routing

ST Approach

- More logic (routing burden) is put on each bridge
- Requires no changes to MAC format
- Full transparency
- Redundant bridges are always in stand-by mode
- No load-sharing across (redundant) bridges

SR Approach

- More logic (burden) is put on the station
- MACs are modified
- Not transparent
- All bridges can participate – load balancing is possible
- More bits to transmit
- More resource / messaging intensive

End of Class 5