COMPSCI 314 S1 C

IEEE 802.2

Logical Link Control

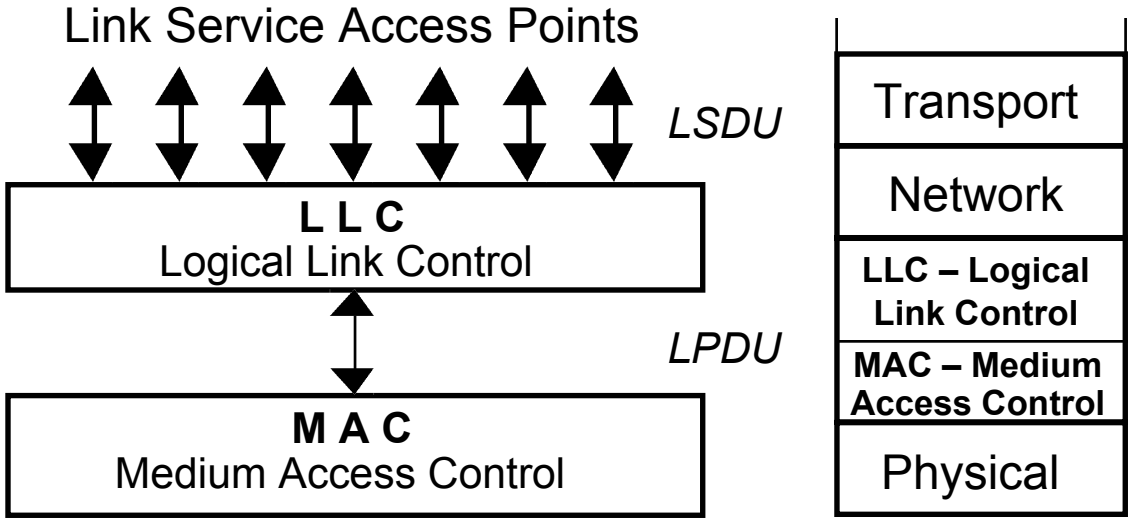
Interconnecting LANs

Spanning Tree Protocol

IEEE 802.2 – Logical Link Control

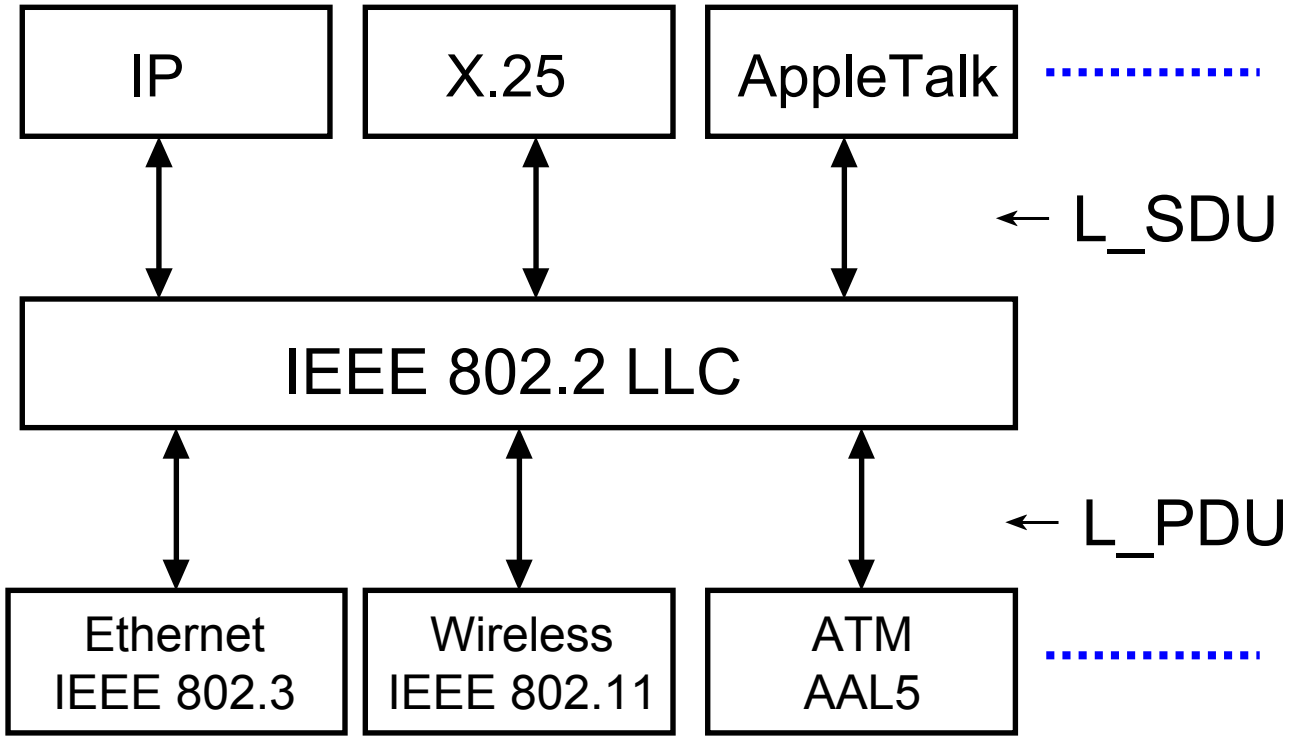
The upper part of the DataLink layer for the 802.*x* networks provides a common interface to the Network Layer from the different media.

- It accepts ***Link Service Data Units*** from the Network Layer and delivers ***Link Protocol Data Units*** to the MAC layer
- Several ***Link Service Access Points*** are multiplexed on to the basic service and effectively define sub-addresses for the node



Logical Link Control (2)

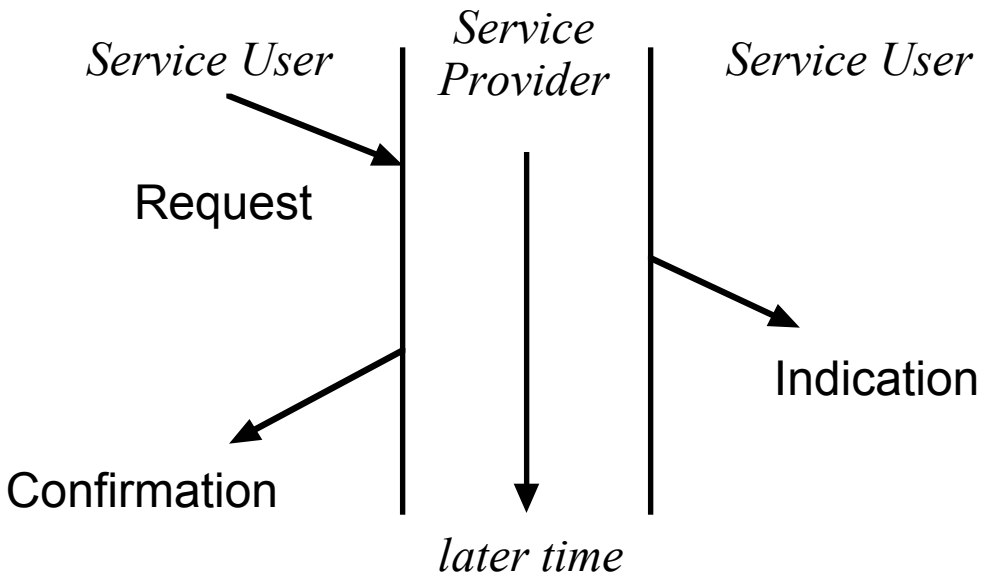
An actual LLC layer might have several systems above it and several physical protocols below it



Protocol diagrams

The 802.2 LLC protocols use three message types —

- A Request is passed down to request a service
- The Request appears at the 'peer' service user as an Indication
- A confirmation is returned to the requester. Other protocols may use more message types



Logical Link Control Services

The most important is the Unacknowledged Connectionless Service, with the two primitives —

L_DATA.request
L_DATA.indication }

with basic functions

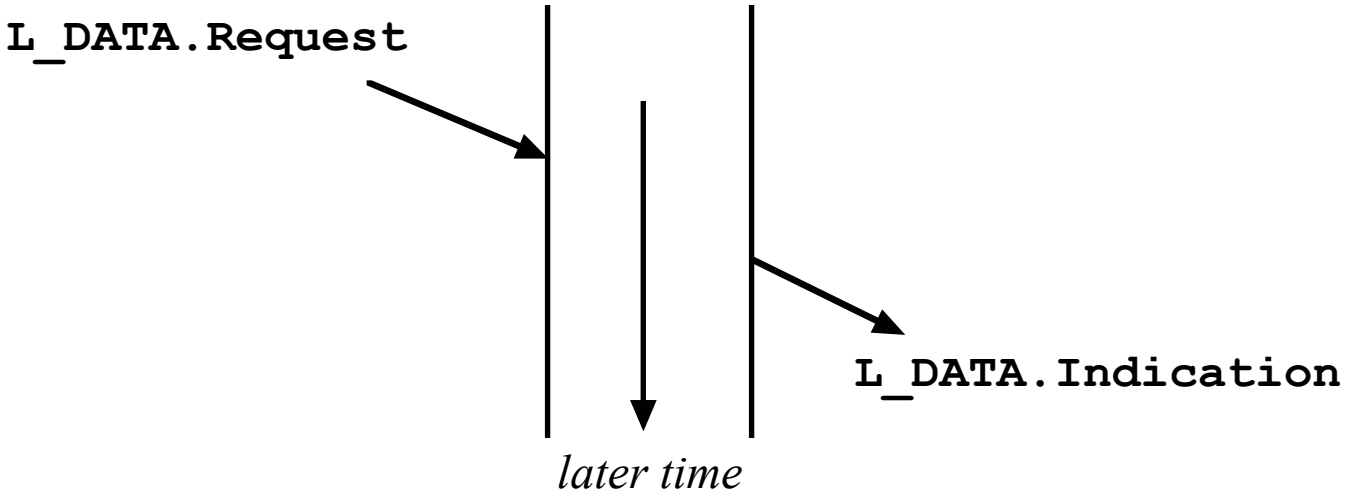
```
L_DATA.request(local_address,
   remote_address, l_sdu, service_class)
L_DATA.indication(local_address,
   remote_address, l_sdu, service_class)
```

The LLC layer is called with a request to pass ‘l_sdu’ (user data) from ‘local_address’ to ‘remote_address,’ or to receive l_sdu from a remote address.

There is no acknowledgement; the l_sdu is just sent and appears.

Logical Link Control Services (2)

The protocol diagram is trivial —



- The address parameters at least combine the MAC address field and the LLC address field (SSAP or DSAP – Source Service Access Point and Destination Service Access Point).
- The remote_address for the **L_DATA.request** may be a broadcast address.
- The **L_DATA.indication** returns the identical LSDU as was provided to the matching **L_DATA.request**.

MAC Service Data Units

The Logical Link Control Layer supplies m_sdu (MAC service data units) for transmission by the MAC layer.

The format of the m_sdu is given later, but uses the primitives –

```
MA_DATA.request(destination_address,
   m_sdu,
   requested_service_class)
```

and the corresponding –

```
MA_DATA.indication( destination_address,
   source_address,
   m_sdu,
   reception_status,
   requested_service_class)
```

Link Protocol Data Units

- The LLC layer receives information (the LLC SDU) through one of its 'Service Access Points' (SAPs), and delivers it to its MAC layer, or vice versa.
- The information transferred through the network must specify the 'Source Service Access Point' (SSAP) and 'Destination Service Point' (DSAP); held in the LLC header prefixed to the SDU.

For simple traffic the LLC header has the form –

DSAP address	SSAP address	Control	Information
8 bits	8 bits	8 bits	8*M bits

Link Protocol Data Units (2)

DSAP address	SSAP address	Control	Information
8 bits	8 bits	8 bits	8*M bits

- The Service Access Points are given “well known” addresses for the usual cases
- The Control byte is 0x 03 for connectionless data
- For routed ISO protocols the LLC Header value (in hexadecimal) is 0x FE-FE-03
- For data transfer the ‘Information’ field is the LLC SDU data

Link Protocol Data Units (3)

Protocols such as IP have a further level of encapsulation –

- The LLC Header, with address 0xAA, is followed by a 'SubNetwork Attachment Point' (SNAP) Header
- The 5-byte SNAP Header has
  - a 3 byte (24 bit) code giving an 'administering authority,' and
  - 16 bits for protocol (00-80 for IP)
- The full prefix is then

  0x AA-AA-03 LLC Header

  0x 00-00-00 08-00 SNAP Header
- When transferring AppleTalk the SNAP header value is

  0x 08-00-07 80-9B

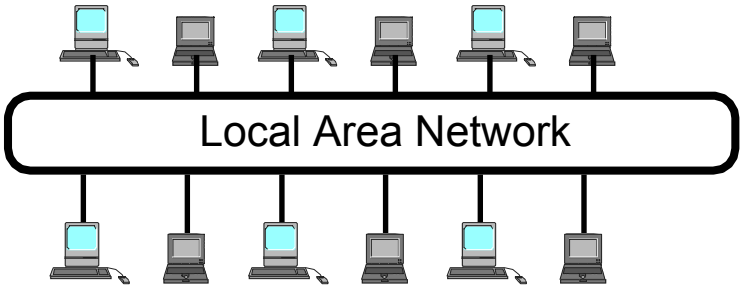
Link Protocol Data Units (4)

- Note that the SNAP protocol field can carry any Ethernet packet type (e.g. 0x 08-06 for ARP)
- In total, an 8-byte header is added to the user message (LLC_SDU) in forming the LLC_PDU submitted to the MAC layer
- The LSB (leftmost) address bit is 0 for unique addresses and 1 for group addresses. 0x00 is a null address and 0xFF is a broadcast address, usually intended for all destination SAPs

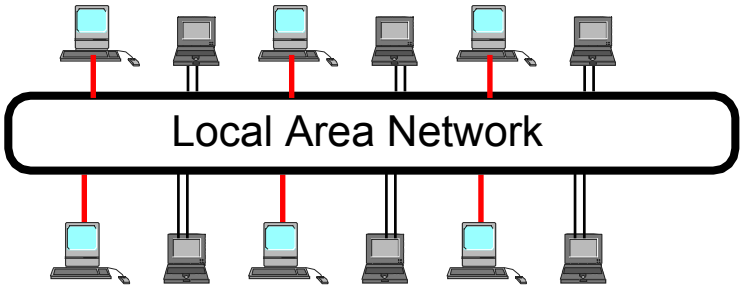
- *If our network only uses Ethernet links, we can use Ethernet convention rather than IEEE 802.3. The 802.3* length *field becomes a* protocol type *field, set to denote the traffic type*

Virtual LANs

- A traditional LAN consists of all the stations and bridges to one side of a router or gateway. All of the stations on the LAN can see all of the traffic. (Bridge routing may affect the detailed message distribution, but certainly all broadcast messages go to all stations.)
- A Virtual LAN uses intelligent switches or bridges to selectively forward traffic to some stations but not others, even if they are on the same physical LAN. Accessing is controlled to individual nodes or stations.



Standard LAN –
all nodes see each other



Virtual LAN –
Nodes divide into mutually invisible sets

Virtual LANs (2)

VLANs can give –

- Flexibility. Reconfiguration can be done by commands, rather then by recabling
- Security. Many VLANs encrypt their data and tunnel it through other protocols. Even if traffic is intercepted it may be unreadable
- Cost. A building may have 3 or 4 VLANs all sharing a single physical network. Thus only one set of cables, switches etc needs to be provided
- Terminals on the same *physical LAN* but different *virtual LANs* can communicate only through some mutually visible router

Virtual LANs (3)

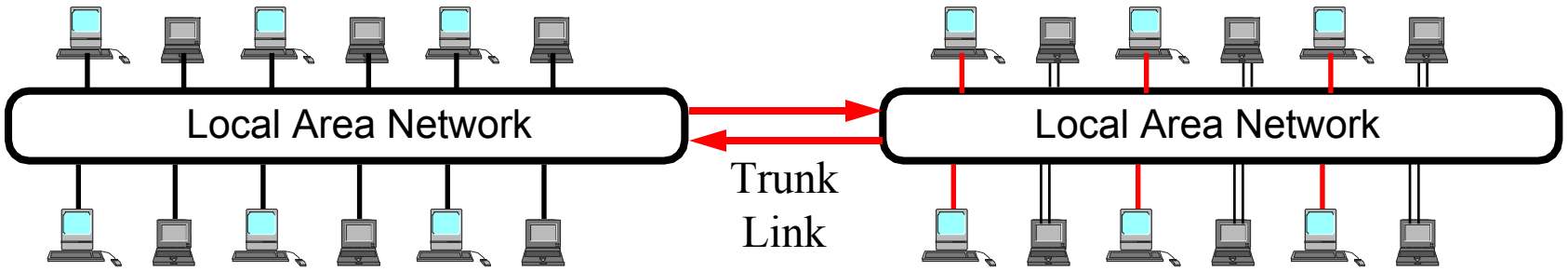
- The special feature of a *switch* is that it connects *workstations* (possibly through hubs) and may aggregate them into one or more *LANs*, whereas a *bridge* connects *LANs*. In simulating a LAN, a switch includes many of the features of a bridge, such as address learning and selective forwarding (and perhaps router functions as well, such as monitoring protocols)

DO NOT confuse Virtual LANs with Virtual Circuits

Note that a Virtual LAN depends on stations connecting individually into a switch and will not work at all with a traditional bus or loop topology.

Defining a VLAN

- At the simplest level a VLAN is simply the list of switch ports associated with the VLAN. For example ports 1, 2, 4, 5, 6 may be on one VLAN and ports 3, 7, 8 may be on another VLAN.
- But VLANs may be distributed across several switches (bridges), connected by "trunk links" with a trunk link shared by several VLANs. Messages between switches therefore include a *tag* sub-header (indicated by an Ethertype code), which includes a *VLAN identifier* and a priority. Equipment that is "VLAN aware" can examine the tag and direct the message appropriately.



PPP (Point-to-Point Protocol)

- PPP is another Data Link protocol, originally designed for Internet links over serial lines between routers
- Defined in RFCs 1661, 2153
- Provides framing for L_SDUs (i.e. packets to be sent)
- Provides SAPs for different higher-layer protocols (allowing the link to carry many protocols)
- Now used over many different link layers, e.g. ADSL uses PPP over Ethernet (carried over ATM links!)

Interconnection of LANs

- Repeaters work at the Physical Level and recognise the coding of bits on to the Physical medium:
  - All traffic on each side is repeated to the other side
  - Ethernet or IEEE 802.3 will recognise packet collisions and force jams
  - There is a limit to the number of repeaters in any path node-to-node; for Ethernet it is 5
- Bridges work at the DataLink level and recognise frame formats:
  - Some bridges selectively forward frames (e.g. spanning tree)
  - Some may work between formats (e.g. Ethernet and Token Ring)

Interconnection of LANs (2)

- Routers work at network level and recognise protocols and traffic types e.g. TCP/IP, Appletalk
  - They may selectively forward (or block) transmission depending on protocol (a primitive *firewall*)
- Gateways at Transport level can convert between protocols
- The more general term 'switch' is often used now for devices that combine functions from several levels

Problems of Bridging or Connecting LANs

The problems include —

- Different data rates:

  Ethernet 10, 100 or 1000 Mb/s,
  Token Ring 4 or 16 Mb/s, FDDI (100 Mb/s)

- Different Frame sizes:

  Ethernet 1500 octets, Token Ring (< 10 ms, say 5000 octet), Token Bus 8000 octets, FDDI 4096 octets

  Bridges cannot reliably fragment and reassemble frames

- Ethernet cannot handle ‘address recognised,’ priorities or ring maintenance

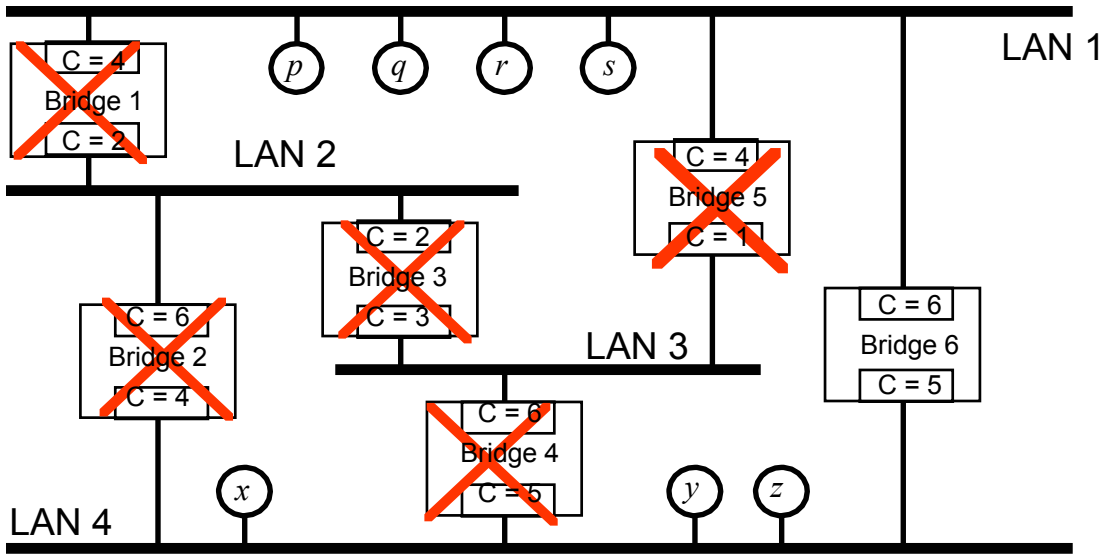
Bridges — forwarding and address learning

- A bridge builds a routing table of addresses which are visible from each port
- A message is not forwarded to a port which cannot see the destination (i.e. forward only broadcast messages, messages to that station or messages to a station which has not transmitted

BUT simple selective forwarding can give loops where LANs have multiple paths.
We solve this problem by building a spanning tree.

Also, we usually forget connections which have not seen traffic for say 20 minutes.

Example LAN Topology



source dest	*message is on LAN(s)*	*information learned by Bridge 6*
p → *r*	1, 4	P is on LAN 1
x → *q*	1, 4	X is on LAN 4
r → *s*	1, 4	R is on LAN 1
r → *p*	1	nothing
x → *p*	1, 4	nothing
z → *x*	4	Z is on LAN 4

Spanning Tree algorithm

- Each bridge has a unique identifier — MAC address & priority
- All bridges on the LAN share a group address ‘all stations on this LAN’
- Within each bridge, each port has a unique ‘port identifier’
- Each port has a cost of transmitting through that port to its LAN
- All bridges exchange Bridge Protocol Data Units (BPDUs) containing costs to reach other bridges and LANs

Spanning Tree algorithm (2)

1. Bridges exchange Bridge Protocol Data Units (BPDUs); if any sees one with *lower* bridge ID it drops out of contention for becoming the root bridge
2. The bridge which finds it has the lowest BridgeID becomes the *root bridge*
3. The root bridge sends BPDUs to all other stations (LAN broadcast) which accumulate *root path costs* (to the root) as they traverse the LAN

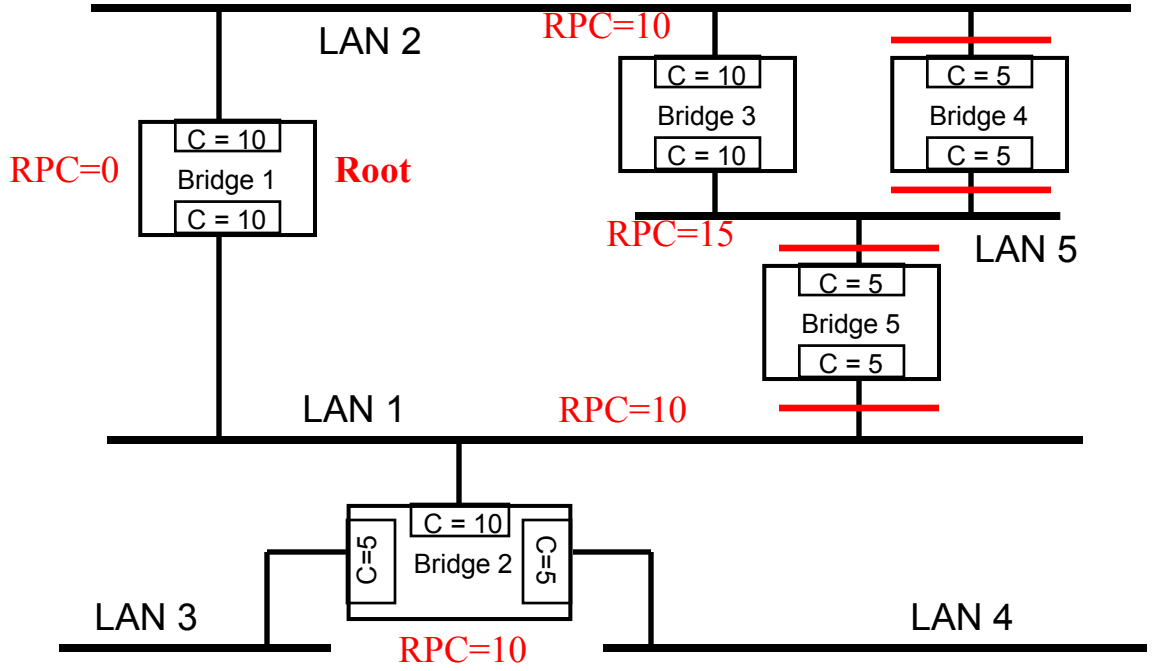
   Only the *receiving* port contributes to the Root Path Cost (RPC). Although BPDUs travel outwards from the root, they accumulate costs from the bridge to the root, *not* from root to bridge
4. For each LAN, the bridge port with minimum root path cost becomes the *designated port* for this LAN; its bridge becomes the *designated bridge* for the LAN

Spanning Tree algorithm (3)

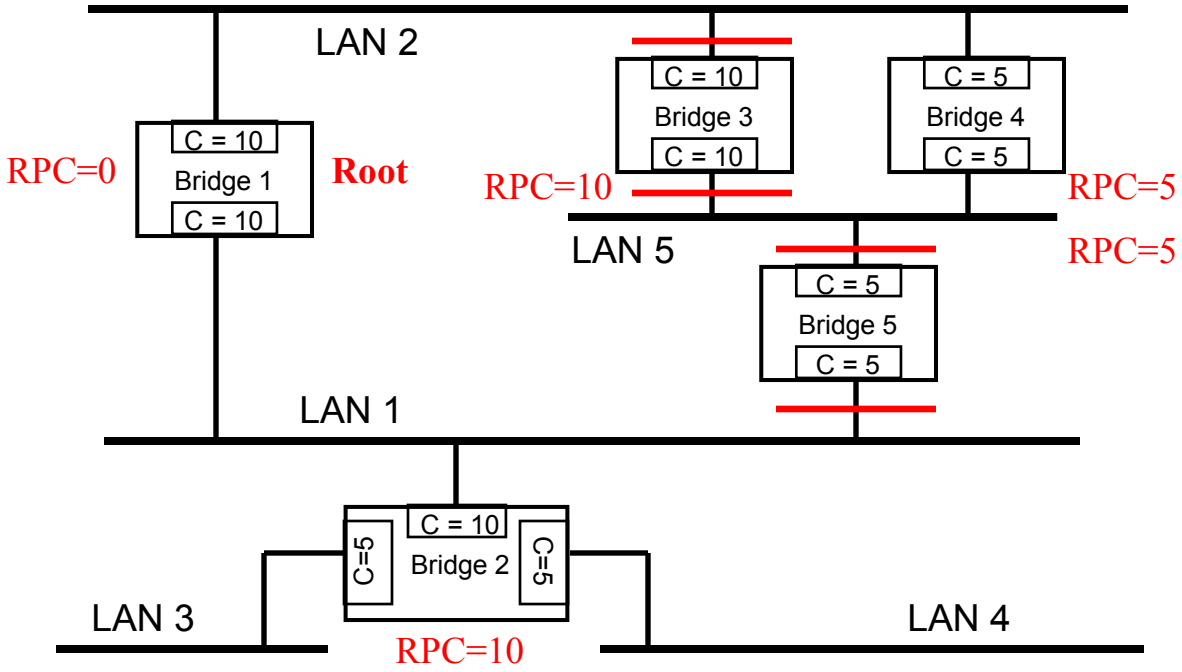
5. The designated bridge and port are the only ones to provide communication between the LAN and the root bridge. All other ports which directly see the root bridge are disabled. With multiple paths, prefer

   (i) the highest priority bridge, and then

   (ii) the lowest port identifier



Spanning Tree algorithm (4)



1. Bridge 1 has lowest Bridge ID and becomes the root bridge
2. LANs 1 & 2 are connected directly to the root bridge
3. From LAN5, Bridge 3 gives RPC = 10, while bridges 4 & 5 give RPC = 5. Bridge 3 drops out because of cost, and Bridge 5 drops out because of higher bridge ID

Spanning Tree Performance

- When a bridge or link fails, the Spanning Tree Protocol (STP) is run again to construct a new tree
- Recovery can take 30 to 60 seconds
- A newer protocol, Rapid Spanning Tree (RSTP), reduces this time to 1 or 2 seconds
- In situations needing even faster recovery, other protocols can also be used …

- Caution: Spanning Tree Protocol doesn't know about VLANs — they can be partitioned by disabled bridges