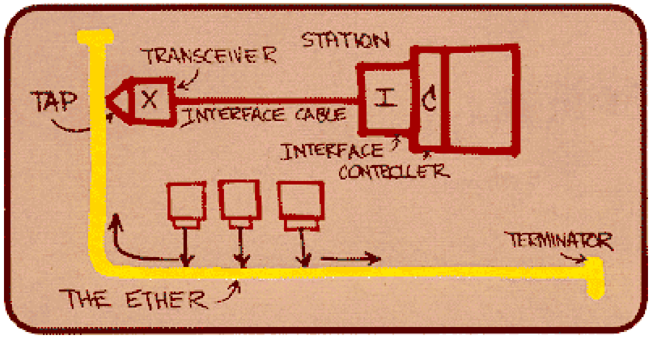
# Lecture 2

# What is Ethernet?

- Developed by Xerox in cooperation with DEC & Intel in 1976



- Ethernet specification is the basis for the IEEE 802.3 standard
  - Specifies the layer 1 (physical) and layer 2 (data-link) of the OSI reference model
  - Ethernet uses a bus or star topology
- Simple, very high speed, low cost
  - Scalable data rates: 10/100 Mbps, 1/10 Gbps, 40/100Gbps
  - Low-cost technology will always win
    - Token Ring, VGAnyLAN, FDDI, ATM, …

# Ethernet Today

- Ethernet has evolved far from its roots of half-duplex CSMA/CD LANs and is hard to pin down today
  - Estimated 250 million Ethernet interfaces
  - > 95% of Internet traffic is transported over Ethernet
- we may use the term today to describe
  - full duplex 10G point-to-point optical links (WAN)
  - metro Ethernet networks (MAN)
  - “Ethernet in the first mile” DSL access (Access)
  - passive optical “GEPON” networks (Access)
  - “wireless Ethernet” 10M hot spots (Home)
  - etc.

# IEEE 802, misc WGs, documents

802 LAN/MAN Standards Committee

- 802
- 802.1 LAN protocols WG
  - **802.1D Bridge, Spanning Tree**
  - **802.1Q VLAN**
  - 802.1ad Q-in-Q
  - 802.1ah Mac-in-Mac
- 802.2 LLC
- 802.3 Ethernet WG
  - **802.3**
  - 802.3z GbE
  - 802.3ad link aggregation
  - 802.3ah EFM
  - 802.3as 2000 byte frames
- 802.11 Wireless LAN WG (WiFi)
  - **802.11**
  - 802.11a/b/g/n
- 802.16 Broadband Wireless Access WG (WiMax)
- 802.17 RPR WG

# 802.3

actually, IEEE only calls 802.3 Ethernet

802.3 is a **large** standard, defining

- MAC frame format, including VLAN support
- medium specifications and attachment units (UTP, coax, fiber, PON)
- repeaters
- interfaces (e.g. MII, GMII)
- rate autonegotiation
- link aggregation

new projects continue to expand scope

- 802.3aq 10GBASE-LRM
- 802.3ar congestion management
- 802.3as frame expansion

# Ethernet Basics

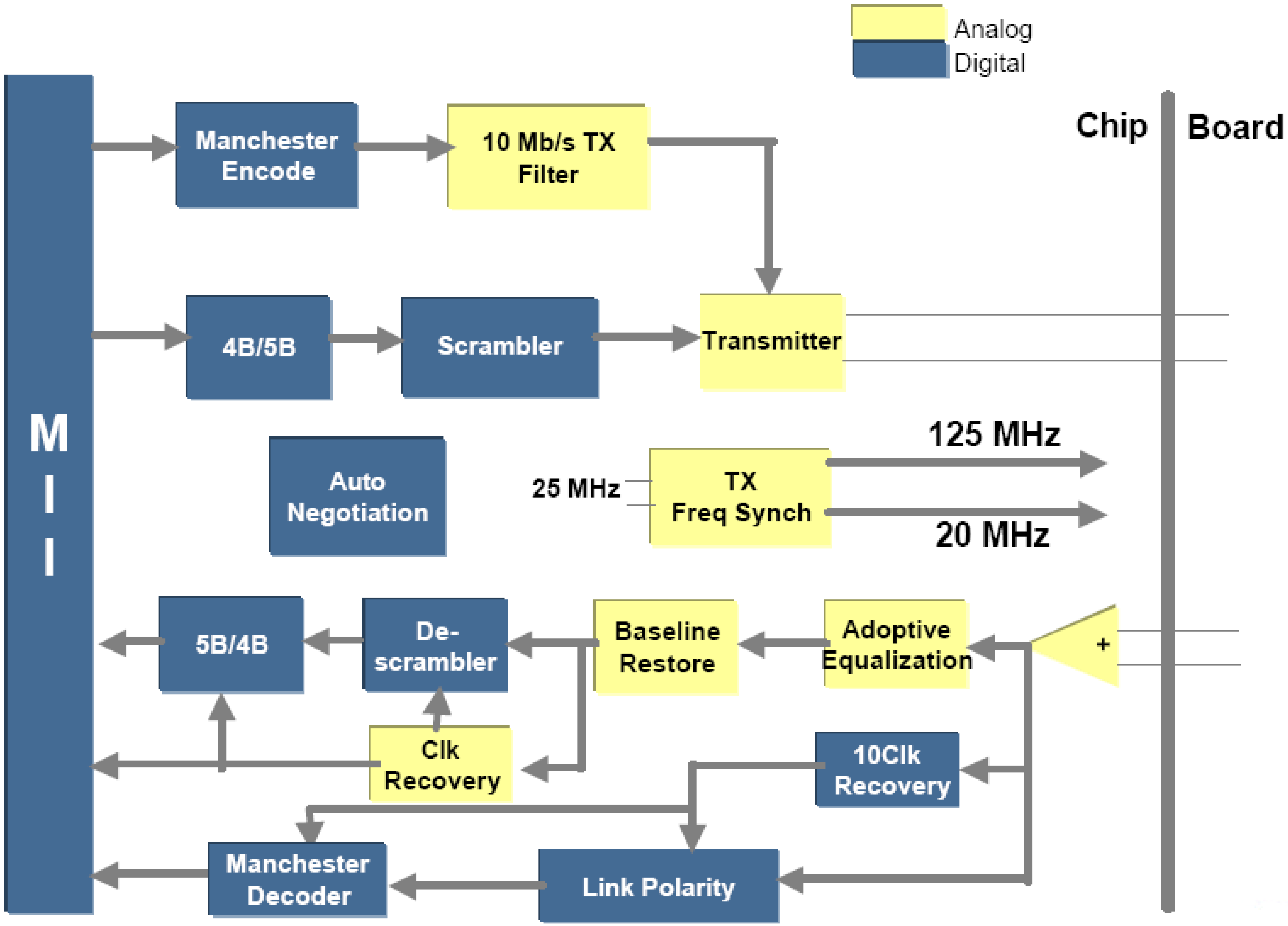
Defines layers 1 and 2 specifications

- Physical layer (layer 1)
  - Provides the electrical, mechanical & procedural specs for the transmission of bits through a communication link, medium or channel
- Data link layer (layer 2)
  - Ensures error control & synchronization between two entities
  - Includes Medium Access Control (MAC) & Logical Link Control (LLC) sub-layers

# PHY Layer Functions

- Circuit establishment and release
- Bit synchronization
- Service data unit
- Data transfer sequencing - serialization & latency
- Fault condition notification
- Network management
- Medium specific control functions
  - Electrical signals, light pulses, frequencies, …
  - Wire destinations, connector types, …
  - Signal strengths, timings, latency, distances, …

# 10/100 Ethernet PHY

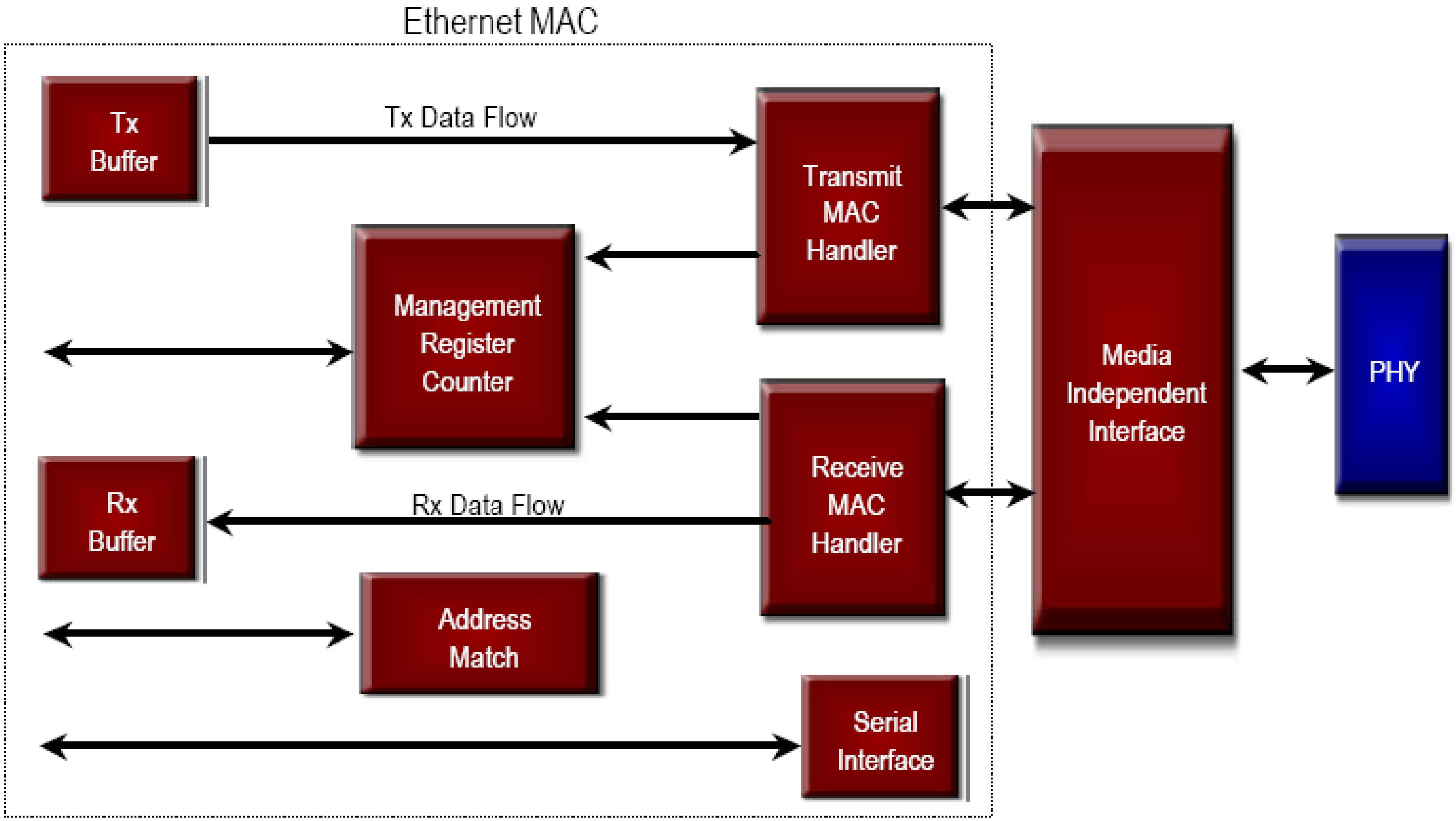


# Data Link Layer Functions

- Data link connection establishment and release
- Service data units
- Framing
- Data transfer
- Frame synchronization
- Frame sequencing
- Error detection
- Identification & parameter exchange
- Flow control
- Physical layer services
- Network management

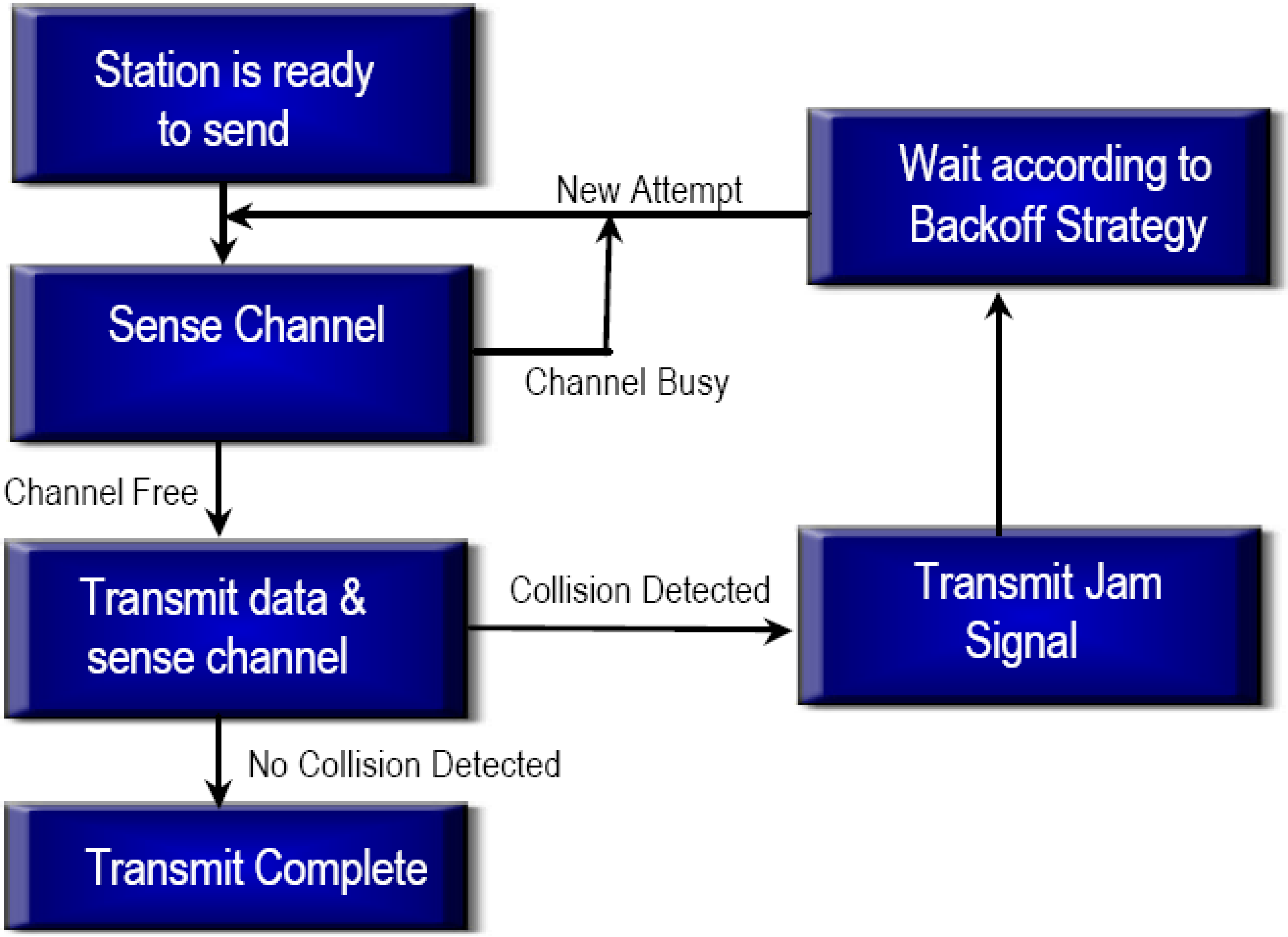
# Ethernet MAC

- Provides all functions necessary to attach an Ethernet physical layer to the host interface

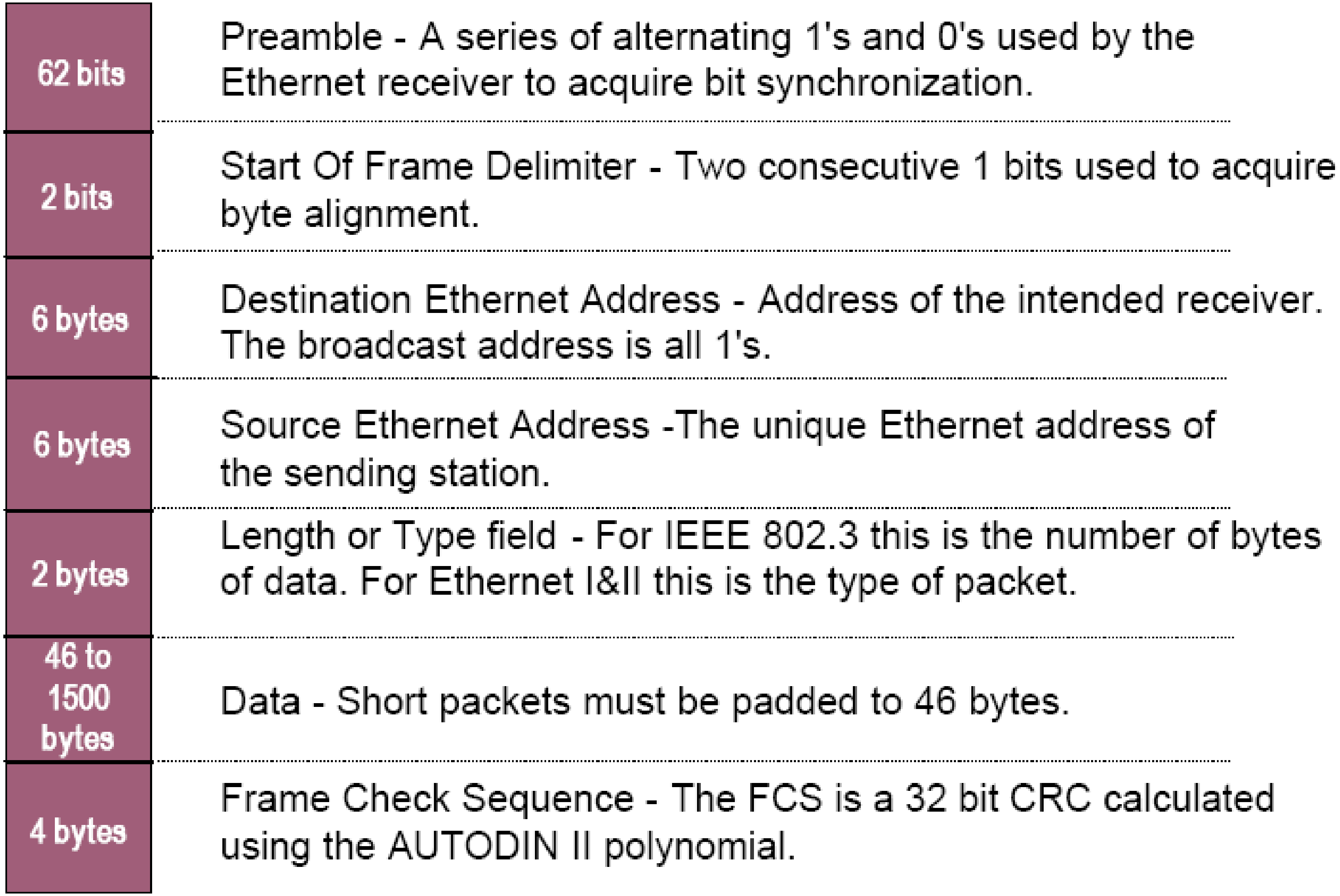


# CSMA/CD

- Carrier Sense Multiple Access w/ Collision Detect



# Ethernet (IEEE802.3) Frames



| | |
|---|---|
| 62 bits | Preamble - A series of alternating 1's and 0's used by the Ethernet receiver to acquire bit synchronization. |
| 2 bits | Start Of Frame Delimiter - Two consecutive 1 bits used to acquire byte alignment. |
| 6 bytes | Destination Ethernet Address - Address of the intended receiver. The broadcast address is all 1's. |
| 6 bytes | Source Ethernet Address -The unique Ethernet address of the sending station. |
| 2 bytes | Length or Type field - For IEEE 802.3 this is the number of bytes of data. For Ethernet I&II this is the type of packet. |
| 46 to 1500 bytes | Data - Short packets must be padded to 46 bytes. |
| 4 bytes | Frame Check Sequence - The FCS is a 32 bit CRC calculated using the AUTODIN II polynomial. |

# Ethernet Addressing

- the most important part of any protocol’s overhead are the address fields
- Ethernet has both source (SA) and destination (DA) fields
- the addresses need to be unique to the network
- the fields are 6-bytes in length in EUI-48 format

  (once called MAC-48, EUI = Extended Unique Identifier)

  $2^{48}$ = 281,474,976,710,656 possible addresses
- addresses can be “universally administered” (burned in) or “locally administered” (SW assigned)

# Ethernet Clients

the 2-byte *Ethertype* identifies the client type

assigned by IEEE Registration Authority

all Ethertypes are greater than 0600 (1536 decimal)

some useful Ethertypes :

- 0800 IPv4
- 0806 ARP
- 8100 VLAN tag
- 8138 Novell IPX
- 86DD IPv6
- 8809 slow protocols
- 8847 MPLS unicast
- 8848 MPLS multicast
- 88D8 CESoETH
- 88F5 MVRP
- 88F6 MMRP

see them all at http://standards.ieee.org/regauth/ethertype/eth.txt

# 802.1D

802.1 discusses MAC bridges

802.1D is also a large standard, defining

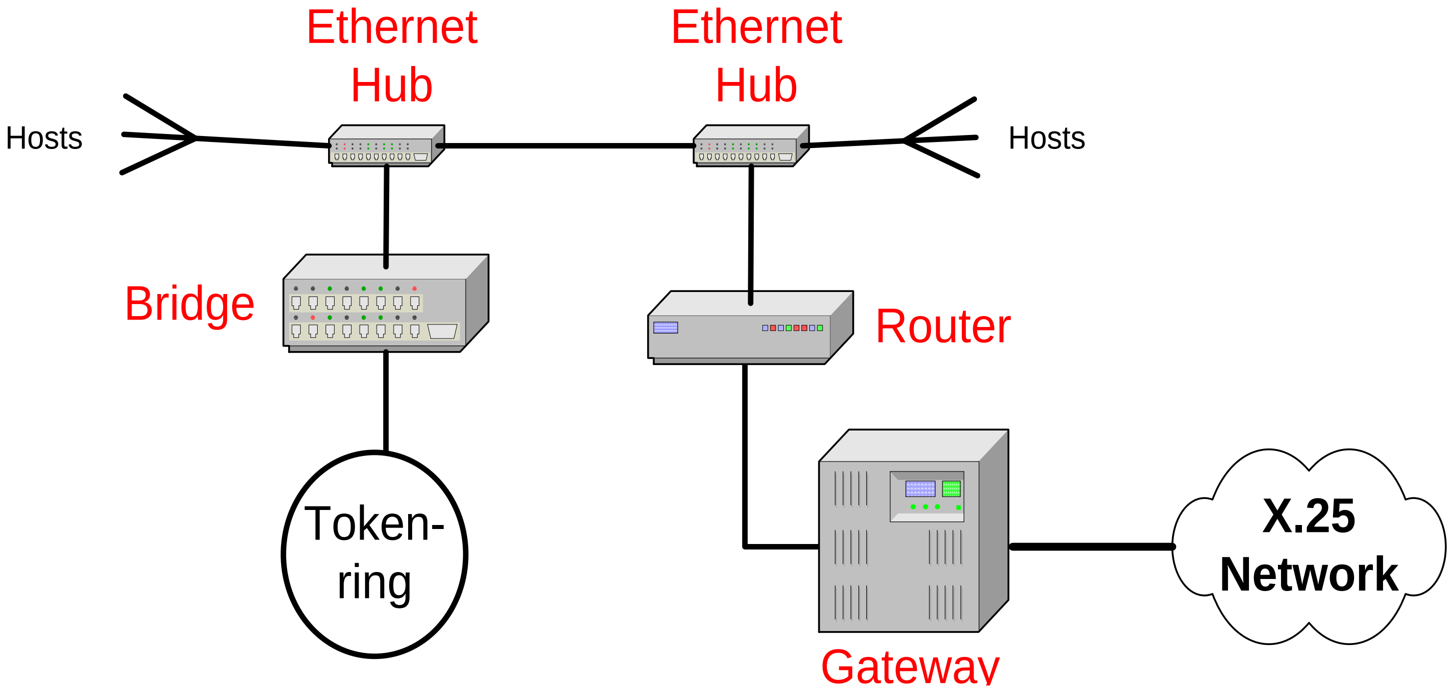
- bridge operation (learning, aging, STP, etc.)
- the architectural model of a bridge
- bridge Protocol and BPDUs
- GARP management protocols

802.1Q is a separate document on VLAN operation

new projects continue to expand scope

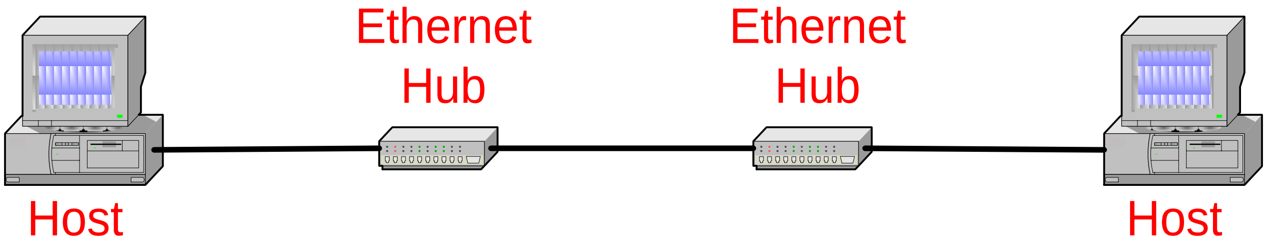
- 802.1ad – Q-in-Q
- 802.1af – MAC key security
- 802.1ag – OAM
- 802.1ah – MAC-in-MAC
- 802.1aj – 2-port MAC relay
- 802.1au – congestion notification

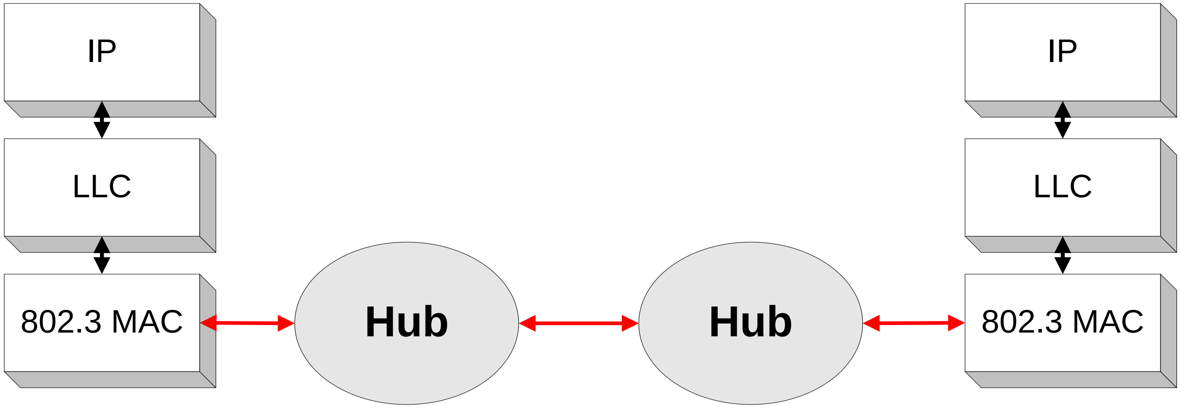
# Interconnection Devices



# Ethernet Hub

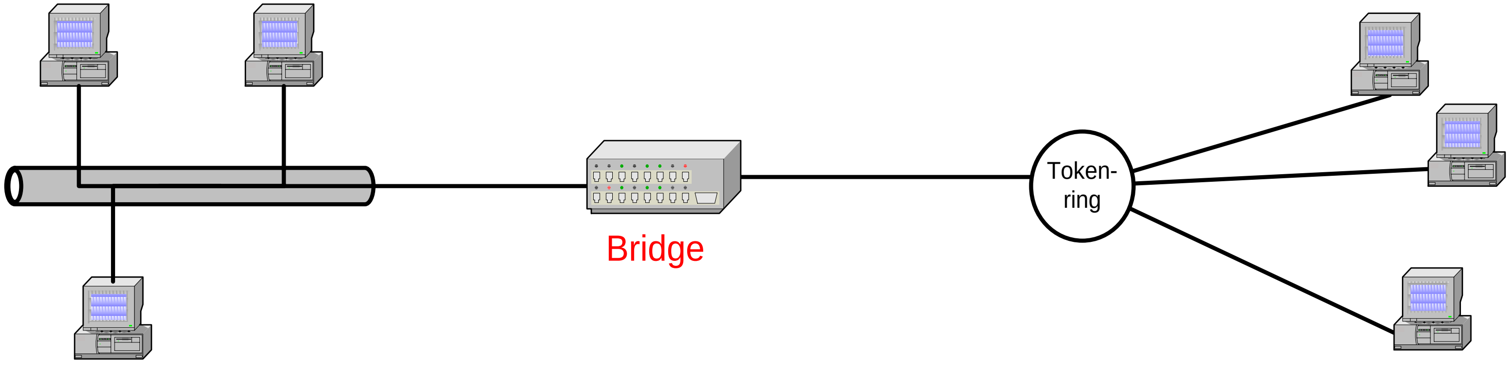
- Used to connect hosts to Ethernet LAN and to connect multiple Ethernet LANs
- Collisions are propagated,
- Transparent to nodes
- All interfaces must run at the same speed (no buffering)

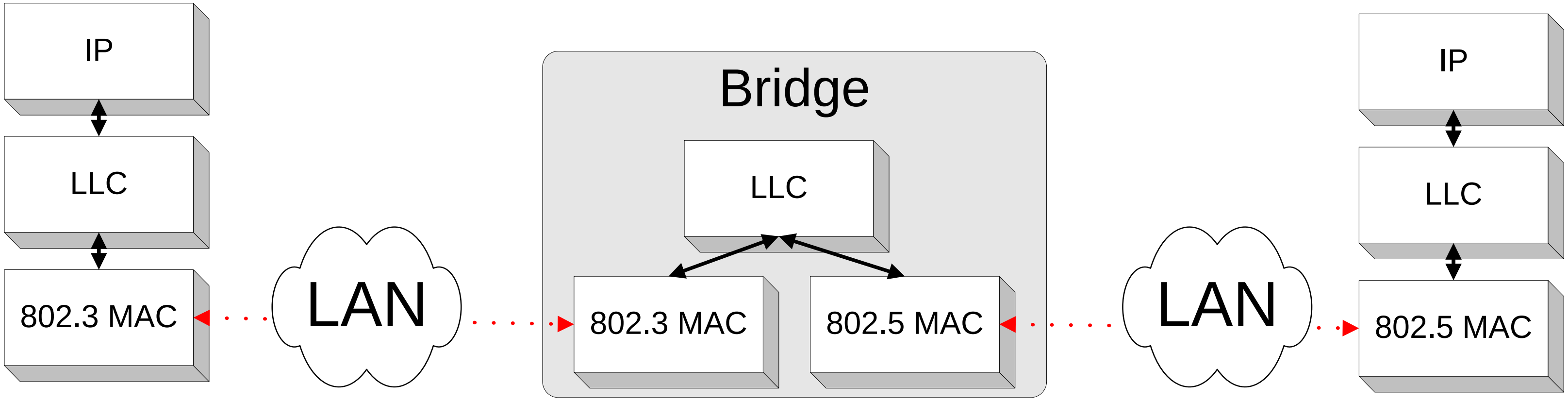




# Bridges/LAN switches

- A *bridge or LAN switch* is a device that interconnects two or more *Local Area Networks* (*LANs*) and forwards packets between these networks.
- Bridges/*LAN switches* operate at the Data Link Layer (Layer 2)




IP

LLC

802.3 MAC

LAN

Bridge

LLC

802.3 MAC

802.5 MAC

LAN

IP

LLC

802.5 MAC

# Terminology: Bridge, LAN switch, Ethernet switch

There are different terms to refer to a data-link layer interconnection device:

- The term bridge was coined in the early 1980s.
- Today, the terms LAN switch or (in the context of Ethernet) Ethernet switch are used.

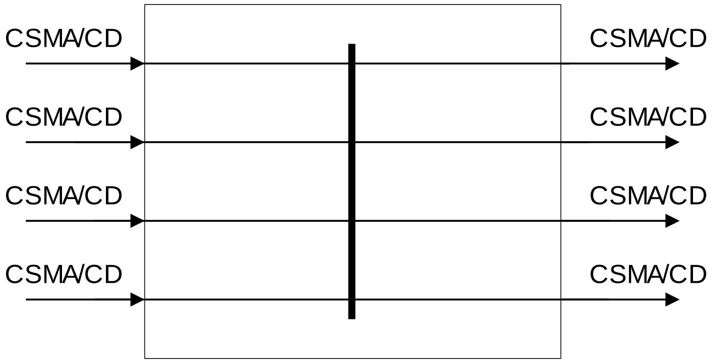
Convention:

- Since many of the concepts, configuration commands, and protocols for LAN switches were developed in the 1980s, and commonly use the old term `*bridge', we* will, with few exceptions, refer to LAN switches as bridges.

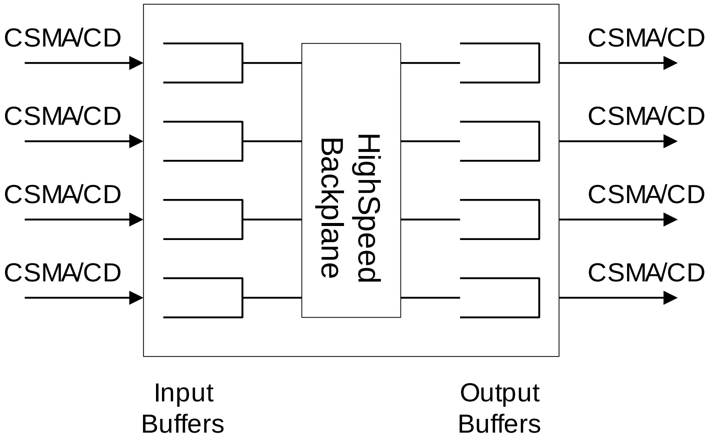
# Ethernet Hubs vs. Ethernet Switches

- An Ethernet switch is a packet switch for Ethernet frames
  - Buffering of frames prevents collisions.
  - Each port is isolated and builds its own collision domain
- An Ethernet Hub does not perform buffering:
  - Collisions occur if two frames arrive at the same time.

**Hub**

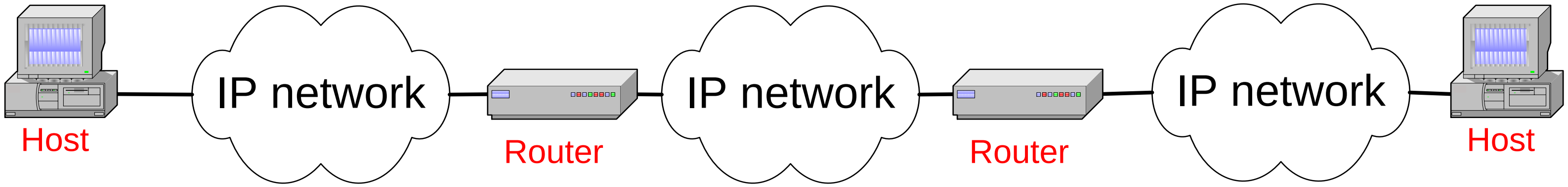


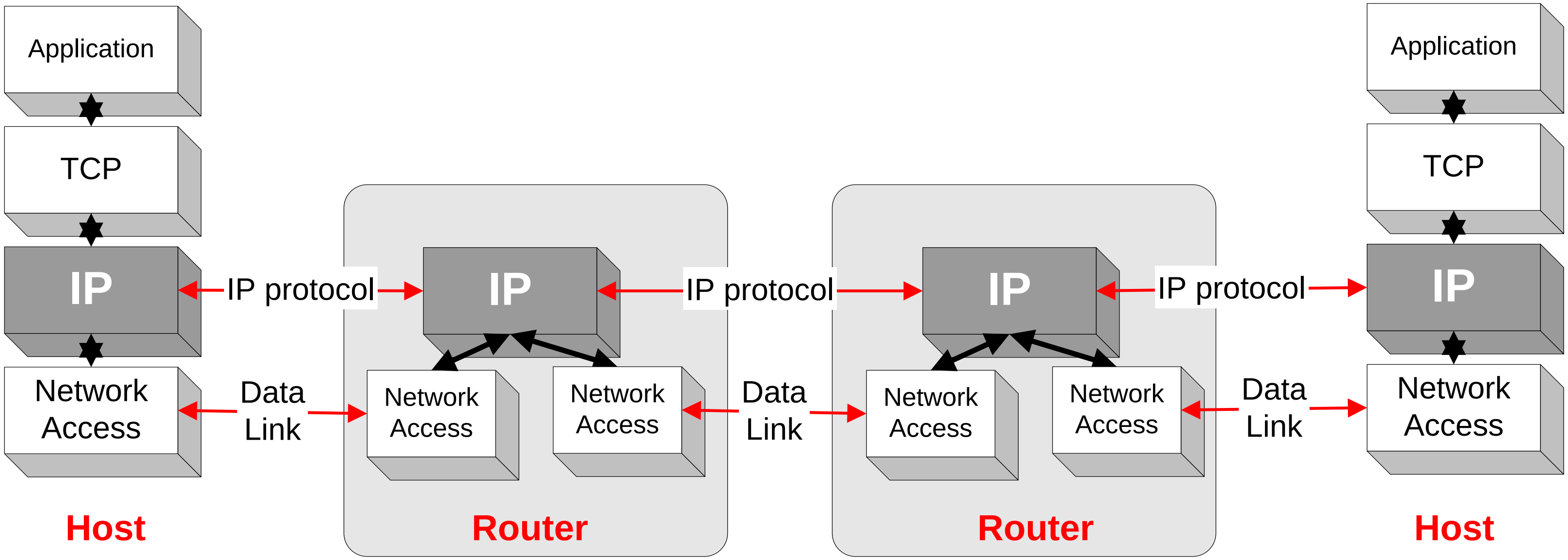
**Switch**



# Routers

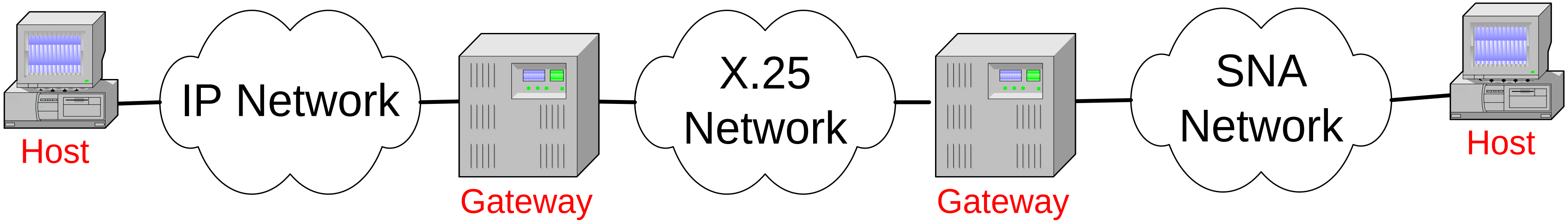
- Routers operate at the Network Layer (Layer 3)
- Interconnect IP networks





# Gateways

- The term “Gateway” is used with different meanings in different contexts
- “Gateway” is a generic term for routers (Level 3)
- “Gateway” is also used for a device that interconnects different Layer 3 networks and which performs translation of protocols (“Multi-protocol router”)



# Comparison

| | Repeating | Bridging | Switching | Routing |
|---|---|---|---|---|
| Works at Layer... | 1 | 2 | 2 | 3 |
| Transparent? | Yes | Yes | Yes | No |
| Performance | worst | ok | high | high delay |
| Complexity | low | medium | high | way-high |
| Topology | restricted, no loops | arbitrary | no loops | arbitrary |
| Packet Flooding | always | broadcast & unknown | broadcast (w/ default route) | never |
| Looping packet | catastrophic | catastrophic | catastrophic | TTL kills it |
| Unknown address | flood | flood or opt. discard | default | default or discard |
| Forwarding | instant | store & fwd | cut thru (typ) | store & fwd |
| Topology learning | none | STP | opt. STP | L3 protocol |

# Bridges

Overall design goal: **Complete transparency**

"Plug-and-play"

Self-configuring without hardware or software changes

Bridges should not impact operation of existing LANs

Three parts to understanding bridges:

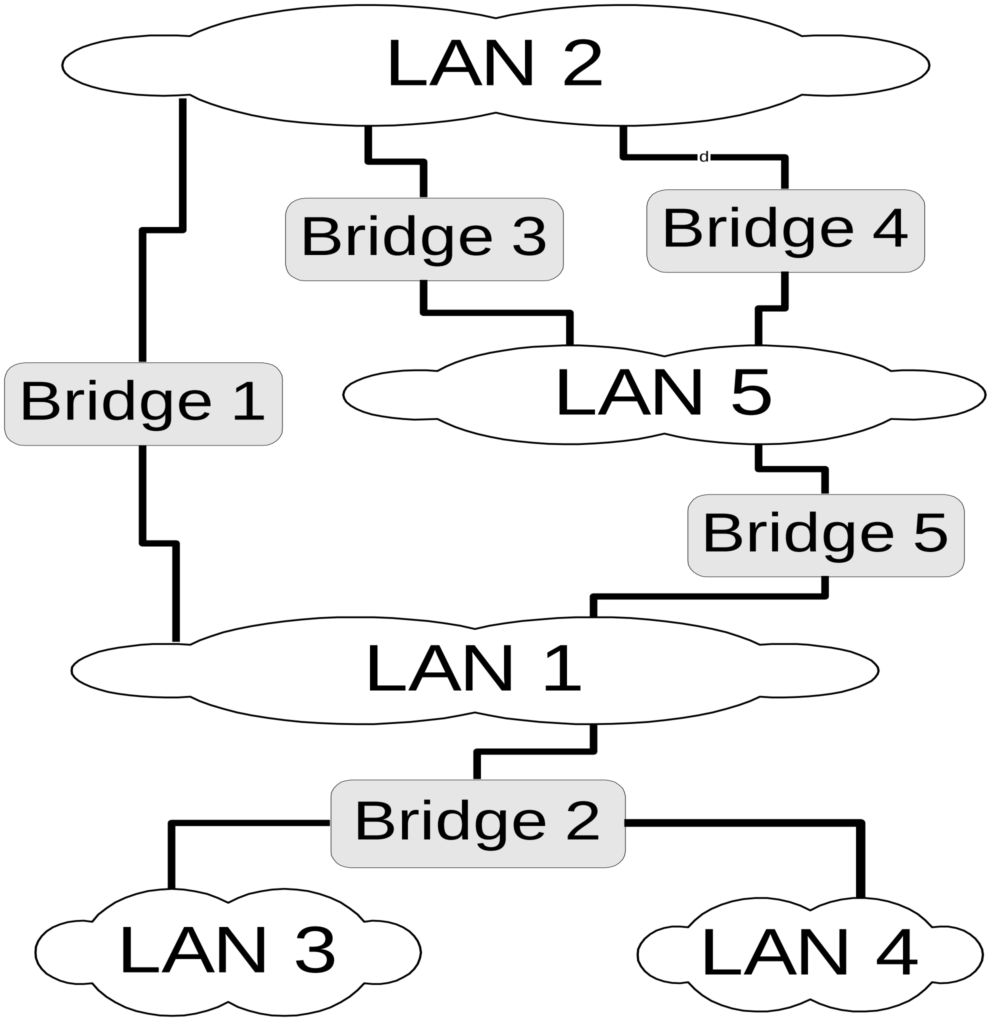
**(1) Forwarding of Frames**

**(2) Learning of Addresses**

**(3) Spanning Tree Algorithm**

# Need for a forwarding between networks

- What do bridges do if some LANs are reachable only in multiple hops ?
- What do bridges do if the path between two LANs is not unique ?



# (1) Frame Forwarding

- Each bridge maintains a **MAC forwarding table**
- Forwarding table plays the same role as the routing table of an IP router
- Entries have the form ( MAC address, port, age), where

| | |
|---|---|
| **MAC address:** | host name or group address |
| **port:** | port number of bridge |
| **age:** | aging time of entry (in seconds) |

with interpretation:

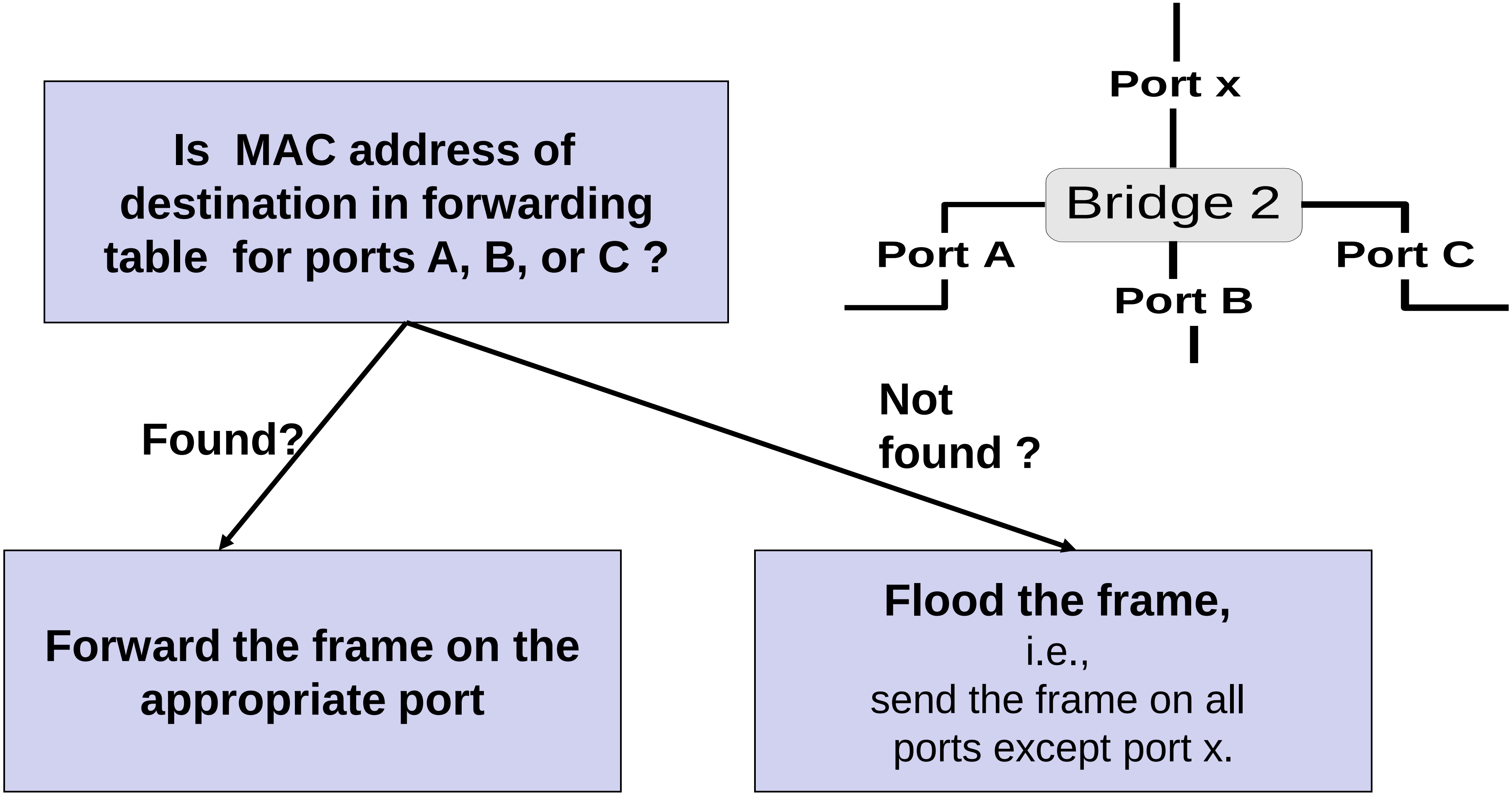
a machine with **MAC address** lies in direction of the **port** number from the bridge. The entry is **age** time units old.

MAC forwarding table

| **MAC address** | **port** | **age** |
|---|---|---|
| a0:e1:34:82:ca:34 | 1 | 10 |
| 45:6d:20:23:fe:2e | 2 | 20 |

# (1) Frame Forwarding

- Assume a MAC frame arrives on port x.



Is MAC address of destination in forwarding table for ports A, B, or C ?

Found?

Forward the frame on the appropriate port

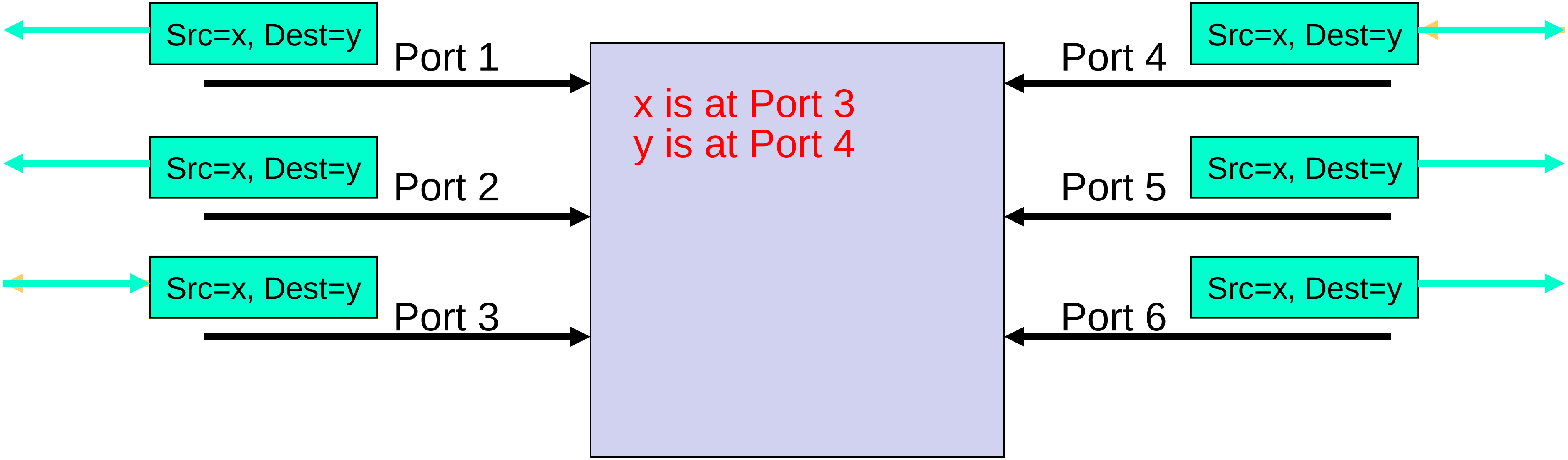
Not found ?

**Flood the frame,**
i.e.,
send the frame on all ports except port x.

# (2) Address Learning

- Routing tables entries are set automatically with a simple heuristic:

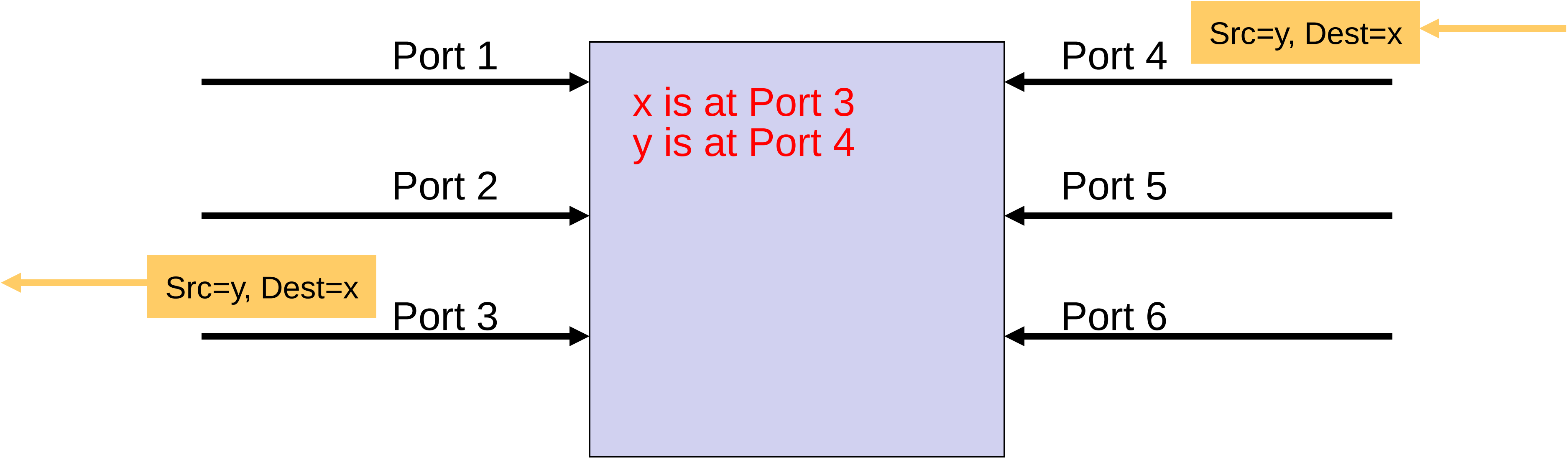
The source field of a frame that arrives on a port tells which hosts are reachable from this port.



# (2) Address Learning

Learning Algorithm:

- For each frame received, the source stores the source field in the forwarding database together with the port where the frame was received.
- All entries are deleted after some time (default is 15 seconds).



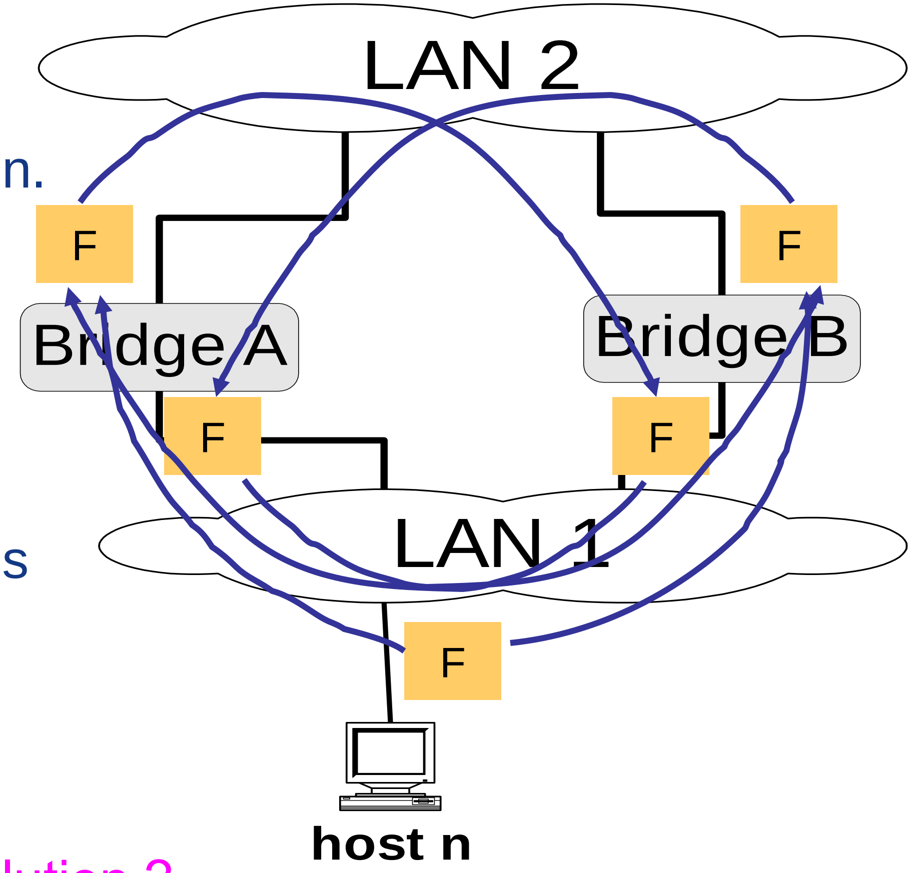
# Danger of Loops

- Consider the two LANs that are connected by two bridges.
- Assume *host n* is transmitting a frame F with unknown destination.

What is happening?

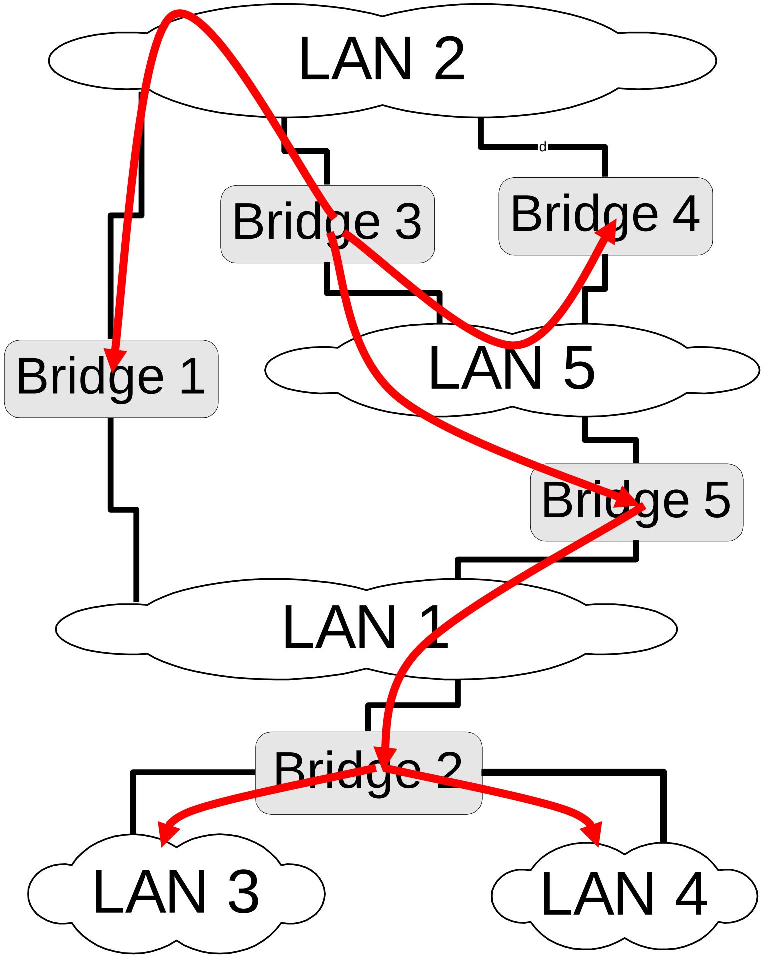
- Bridges A and B flood the frame to LAN 2.
- Bridge B sees F on LAN 2 (with unknown destination), and copies the frame back to LAN 1
- Bridge A does the same.
- The copying continues

Where’s the problem? What’s the solution ?

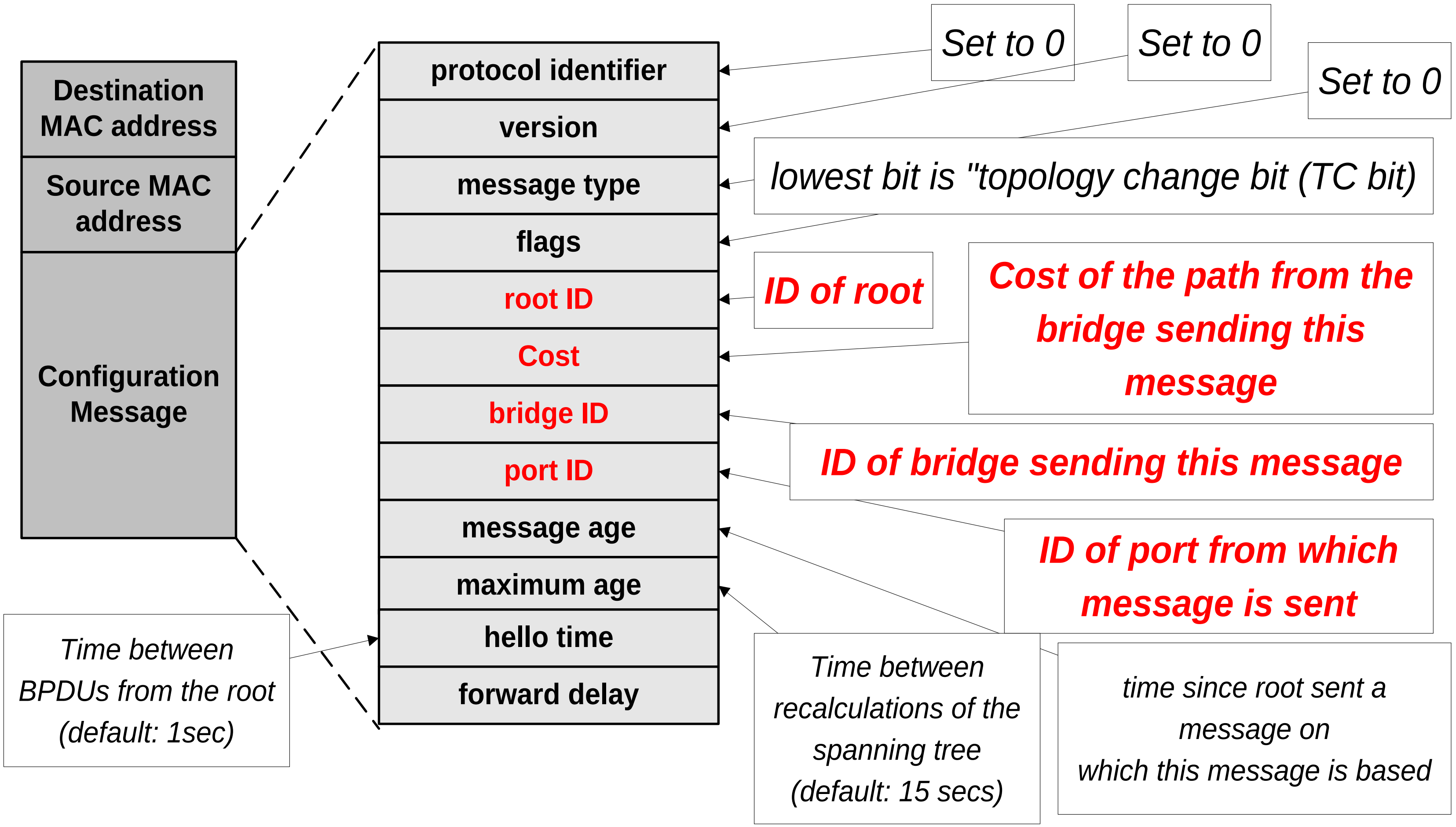


# Spanning Tree Protocol (IEEE 802.1d)

- The Spanning Tree Protocol (SPT) is a solution to prevent loops when forwarding frames between LANs
- The SPT is standardized as the IEEE 802.1d protocol
- The SPT organizes bridges and LANs as spanning tree in a dynamic environment
  - Frames are forwarded only along the branches of the spanning tree
  - Note: Trees don’t have loops
- Bridges that run the SPT are called transparent bridges
- Bridges exchange messages to configure the bridge (Configuration Bridge Protocol Data Unit or BPDUs) to build the tree.



# Configuration BPDUs



Destination MAC address

Source MAC address

Configuration Message

protocol identifier

version

message type

flags

root ID

Cost

bridge ID

port ID

message age

maximum age

hello time

forward delay

*Set to 0*

*Set to 0*

*Set to 0*

*lowest bit is "topology change bit (TC bit)*

***ID of root***

***Cost of the path from the bridge sending this message***

***ID of bridge sending this message***

***ID of port from which message is sent***

*Time between BPDUs from the root (default: 1sec)*

*Time between recalculations of the spanning tree (default: 15 secs)*

*time since root sent a message on which this message is based*

# What do the BPDUs do?

With the help of the BPDUs, bridges can:

- Elect a single bridge as the root bridge.
- Calculate the distance of the shortest path to the root bridge
- Each LAN can determine a designated bridge, which is the bridge closest to the root. The designated bridge will forward packets towards the root bridge.
- Each bridge can determine a root port, the port that gives the best path to the root.
- Select ports to be included in the spanning tree.

# Concepts

- Each bridge as a unique identifier: Bridge ID

  Bridge ID = Priority : 2 bytes

  Bridge MAC address: 6 bytes

  – Priority is configured
  – Bridge MAC address is lowest MAC addresses of all ports

- Each port of a bridge has a unique identifier (port ID).
- Root Bridge: The bridge with the lowest identifier is the root of the spanning tree.
- Root Port: Each bridge has a root port which identifies the next hop from a bridge to the root.

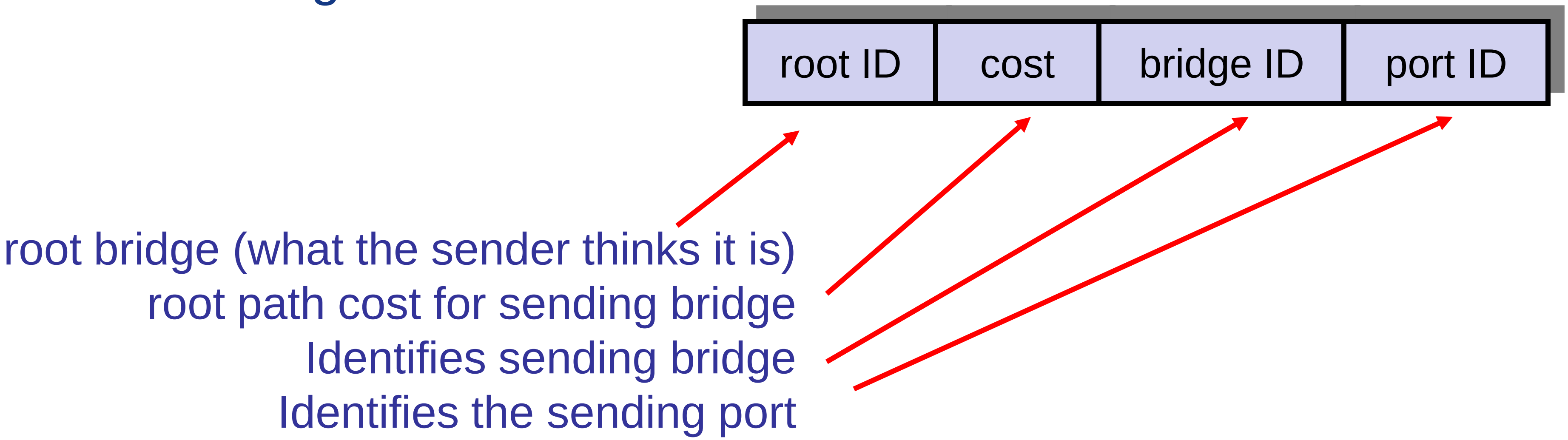
# Concepts

- Root Path Cost: For each bridge, the cost of the min-cost path to the root.
- Designated Bridge, Designated Port: Single bridge on a LAN that provides the minimal cost path to the root for this LAN:
  - if two bridges have the same cost, select the one with highest priority
  - if the min-cost bridge has two or more ports on the LAN, select the port with the lowest identifier

- Note: We assume that “cost” of a path is the number of “hops”.

# Steps of Spanning Tree Algorithm

- Each bridge is sending out BPDUs that contain the following information:



| root ID | cost | bridge ID | port ID |
|---|---|---|---|

root bridge (what the sender thinks it is)
root path cost for sending bridge
Identifies sending bridge
Identifies the sending port

- The transmission of BPDUs results in the distributed computation of a spanning tree
- The convergence of the algorithm is very quick

# Ordering of Messages

- We define an ordering of BPDU messages

| ID R1 | C1 | ID B1 | ID P1 |
|---|---|---|---|

**M1**

| ID R2 | C2 | ID B2 | ID P2 |
|---|---|---|---|

**M2**

We say M1 advertises a better path than M2 (“M1<<M2”) if

(R1 < R2),

Or (R1 == R2) and (C1 < C2),

Or (R1 == R2) and (C1 == C2) and (B1 < B2),

Or (R1 == R2) and (C1 == C2) and (B1 == B2) and (P1 < P2)

# Initializing the Spanning Tree Protocol

- Initially, all bridges assume they are the root bridge.
- Each bridge B sends BPDUs of this form on its LANs from each port P:

| | | | |
|---|---|---|---|
| B | 0 | B | P |

- Each bridge looks at the BPDUs received on all its ports and its own transmitted BPDUs.
- Root bridge is the smallest received root ID that has been received so far (Whenever a smaller ID arrives, the root is updated)

# Operations of Spanning Tree Protocol

- Each bridge B looks on all its ports for BPDUs that are better than its own BPDUs
- Suppose a bridge with  BPDU:

M1

| | | | |
|---|---|---|---|
| R1 | C1 | B1 | P1 |

receives a “better” BPDU:

M2

| | | | |
|---|---|---|---|
| R2 | C2 | B2 | P2 |

Then it will update the BPDU to:

| | | | |
|---|---|---|---|
| R2 | C2+1 | B1 | P1 |

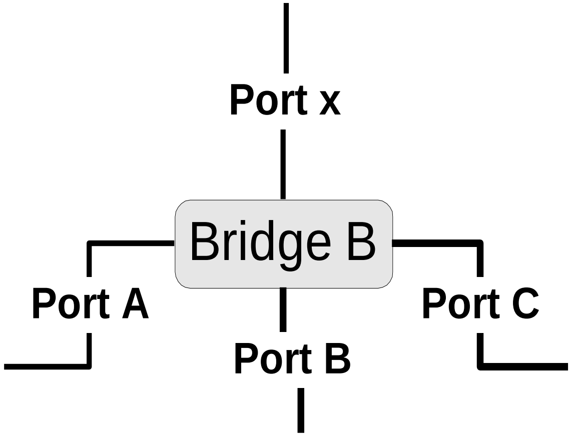
- However, the new BPDU is not necessarily sent out
- On each bridge, the port where the “best BPDU” (via relation “<<“) was received is the root port of the bridge.

# When to send a BPDU

- Say, B has generated a BPDU  for each port x

| | | | |
|---|---|---|---|
| R | Cost | B | x |

- B will send this BPDU on port x only if its BPDU is better (via relation “<<“) than any BPDU that B received from port x.



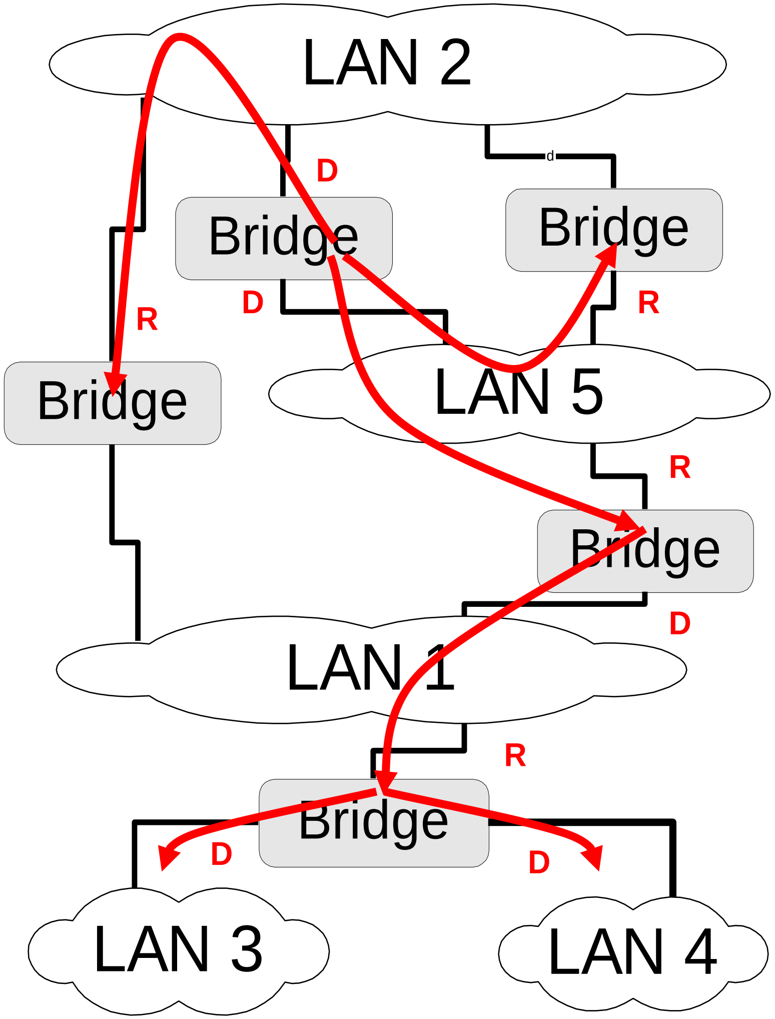
- In this case, B also assumes that it is the designated bridge for the LAN to which the port connects
- And port x is the designated port of that LAN

# Selecting the Ports for the Spanning Tree

- Each bridges makes a local decision which of its ports are part of the spanning tree
- Now B can decide which ports are in the spanning tree:
  - B’s root port is part of the spanning tree
  - All designated ports are part of the spanning tree
  - All other ports are not part of the spanning tree
- B’s ports that are in the spanning tree will forward packets (=forwarding state)
- B’s ports that are not in the spanning tree will not forward packets (=blocking state)

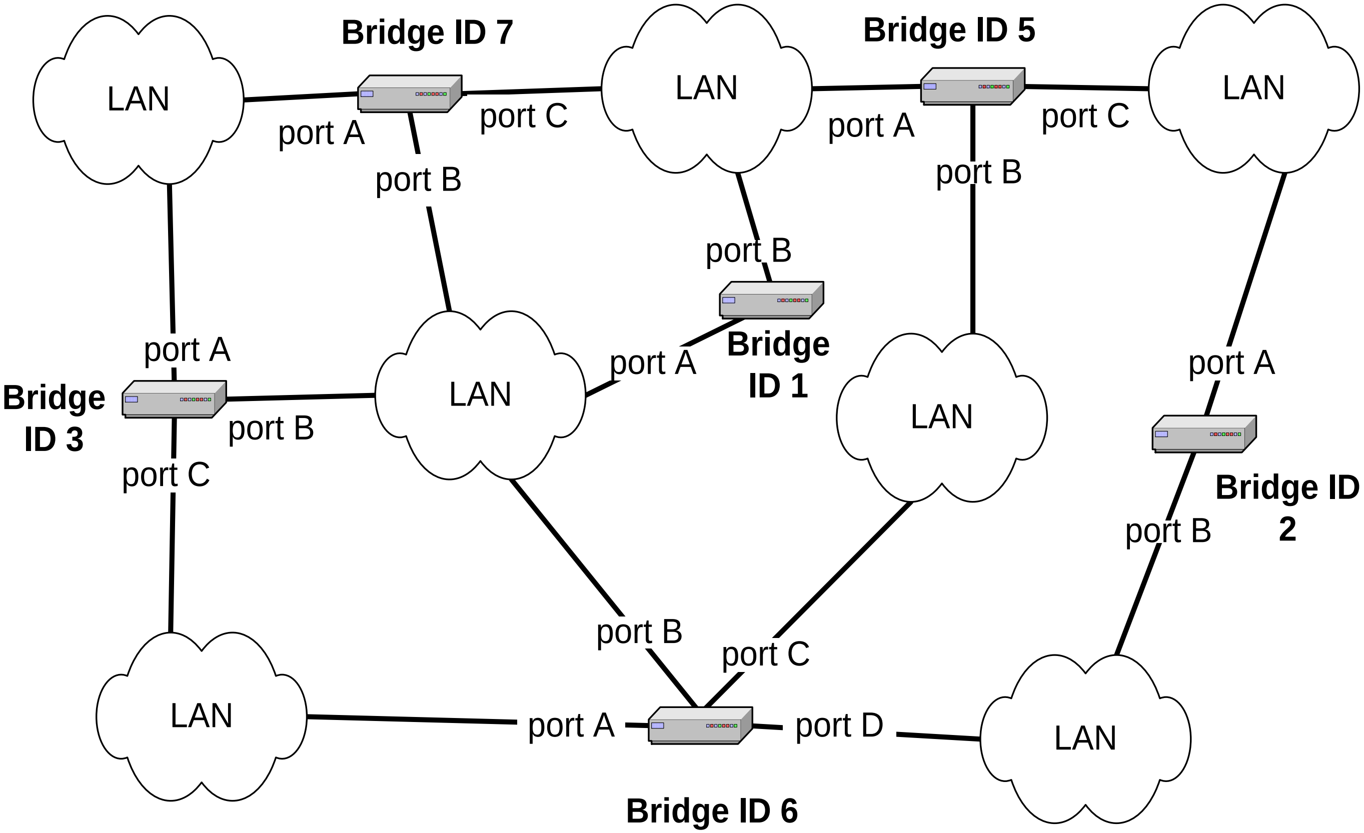
# Building the Spanning Tree

- Consider the network on the right.
- Assume that the bridges have calculated the designated ports (D) and the root ports (R) as indicated.
- What is the spanning tree?
  - On each LAN, connect R ports to the D ports on this LAN



# Example

- Assume that all bridges send out their BPDU's once per second, and assume that all bridges send their BPDUs at the same time
- Assume that all bridges are turned on simultaneously at time T=0 sec.



# Example: BPDU's sent by the bridges

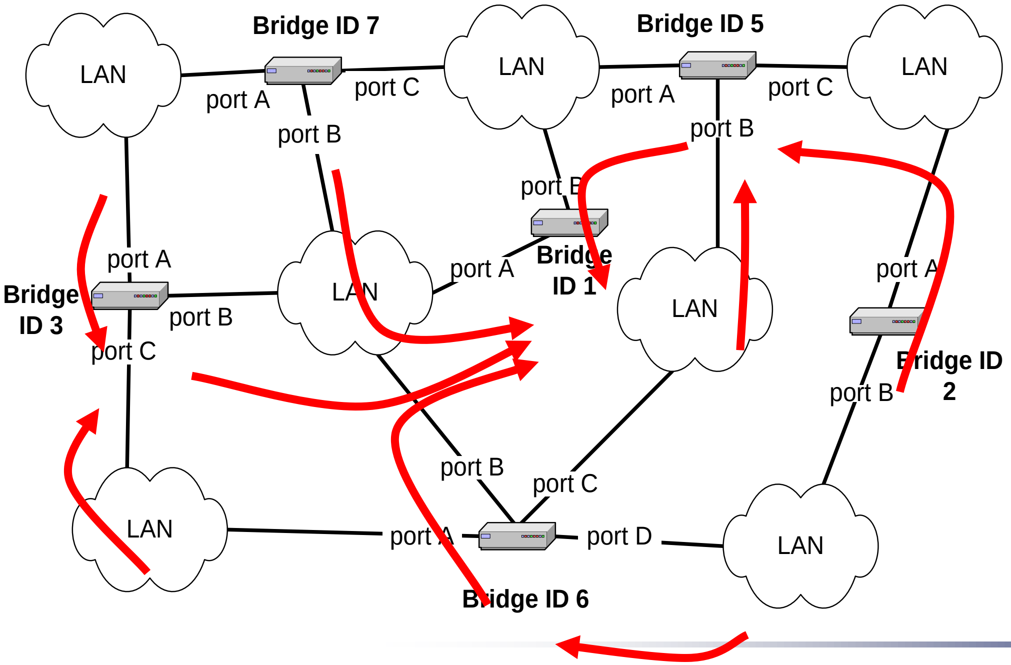
| | Bridge 1 | Bridge 2 | Bridge 3 | Bridge 5 | Bridge 6 | Bridge 7 |
|---|---|---|---|---|---|---|
| **T=0sec** | (1,0,1,port)<br>sent on ports: A,B | (2,0,2,port)<br>ports A,B | (3,0,3,port)<br>ports A,B,C | (5,0,5,port)<br>ports A,B,C | (6,0,6,port)<br>ports A,B,C,D | (7,0,7,port)<br>ports A,B,C |
| **T=1sec** | (1,0,1,port)<br>A,B | (2,0,2,port)<br>A,B | (1,1,3,port)<br>A,C | (1,1,5,port)<br>B,C | (1,1,6,port)<br>A,C,D | (1,1,7,port)<br>A |
| **T=2sec** | (1,0,1,port)<br>A,B | (1,2,2,port)<br>none | (1,1,3,port)<br>A,C | (1,1,5,port)<br>B,C | (1,1,6,port)<br>D | (1,1,7,port)<br>none |

In the table (1,0,1,port) means that the BPDU is (1,0,1,A) if the BPDU is sent on port A and (1,0,1,B) if it is sent on port B.
At T=1, Bridge 7 receives two BPDUs from Bridge 1: (1,0,1,A) and (1,0,1,B). We assume that A is numerically smaller than B. If this is not true, then the root port of Bridge 7 changes.

# Example: Settings after convergence

| | Bridge 1 | Bridge 2 | Bridge 3 | Bridge 5 | Bridge 6 | Bridge 7 |
|---|---|---|---|---|---|---|
| Root Port | - | A | B | A | B | B |
| Designated Ports | A,B | - | A,C | B,C | D | - |
| Blocked ports | - | B | - | - | A,C | A,C |

Resulting tree:



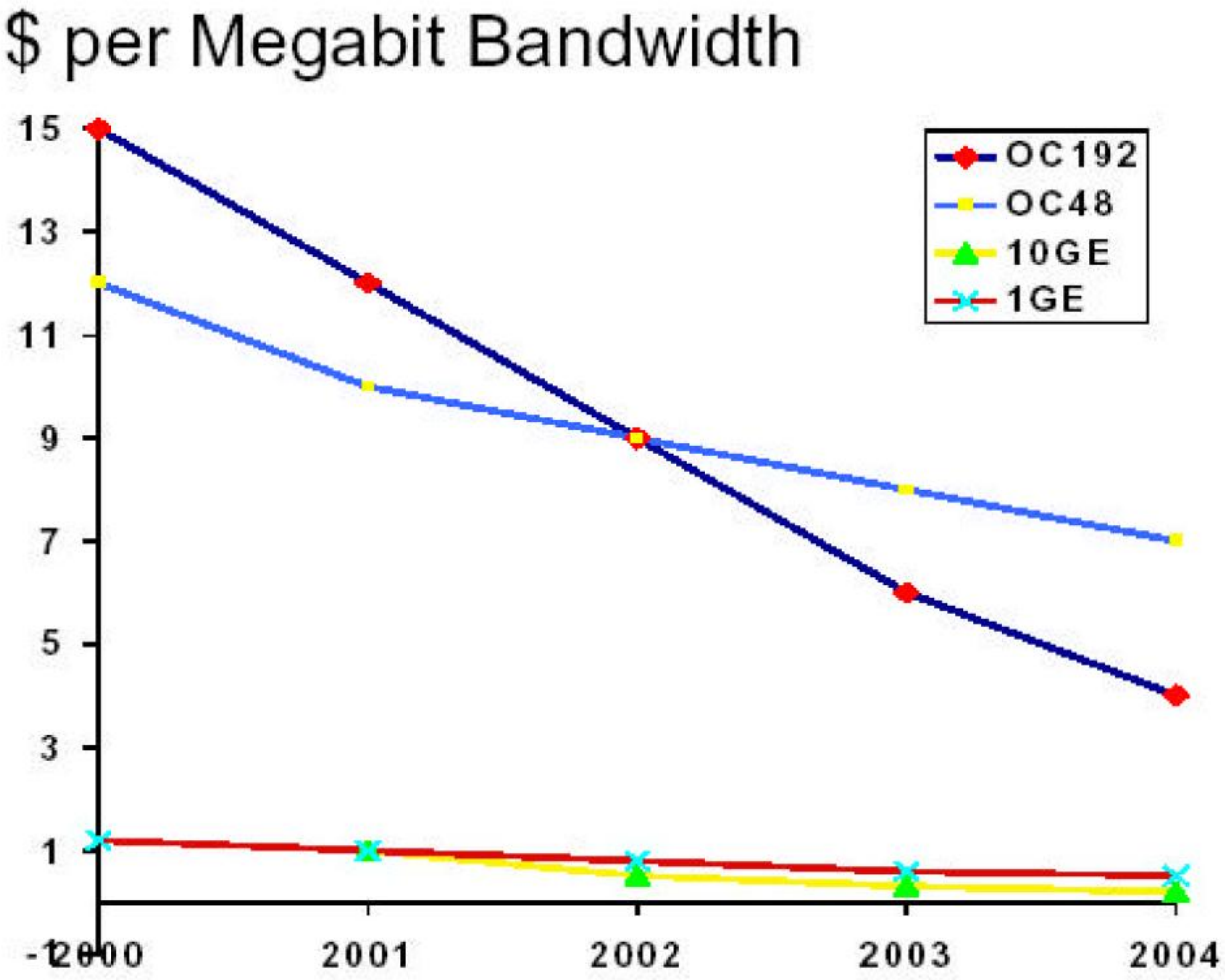
# The Mother of SPT: Radia Perlman

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree

# Need for Speed

- IEEE 802.3, CSMA/CD (1983)
- IEEE 802.3i, 10Base-T (1990)
- IEEE 802.3u, 100Base-T (1995)
- IEEE 802.3z, 1000Base-X, GbE (1999)
- IEEE 802.3 ae, 10GbE (2002)
- IEEE 802.3 ba, 40/100GbE (2007)

# Ethernet Economics



# GbE & 10 GbE - Still Ethernet

- Faster, Cheaper, Simpler
- An IEEE 802.3 standards project
  - Evolution from a 10 Mbps shared coaxial connection to a dedicated 10 Gbps optical connection
- Ethernet ubiquity leads to rapid development & adoption
  - Leverages features of its Ethernet predecessors
    - No change to Ethernet frames
    - Support Ethernet enhancements (e.g., link aggregation)
    - Specify implementation interfaces to enable interoperability
  - Minimizes user learning curve and support costs
    - Same management architecture
    - Compatibility with familiar tools

# Gigabit Ethernet - Summary

- IEEE 802.3z
- 1000 Mbps
- 802.3 Ethernet frame format
  - Preserves minimum and maximum frame size of 802.3
- Full & half-duplex operation
- Meets all 802 requirements except Hamming distance
- Support star-wired topologies
- Support fiber & copper
  - At least 2 km over SMF
  - At least 500 m over MMF
  - At least 25 m over copper
- Collision domain diameter of 200m
- Accommodate 802.3x flow control

# Customer Challenges

- Enterprise
  - Multiple locations in one metro area
  - Need to support high-bandwidth applications: imaging, CAD/CAM, storage
- Service Providers
  - Need high-speed connections between POPs
  - Need flexible bandwidth, just-in-time provisioning
- Metro/MANs
  - Very low cost
  - Emergence of metro Ethernet services

# 1/10 GbE Vision

***Integrated Services Video – Voice – Data***

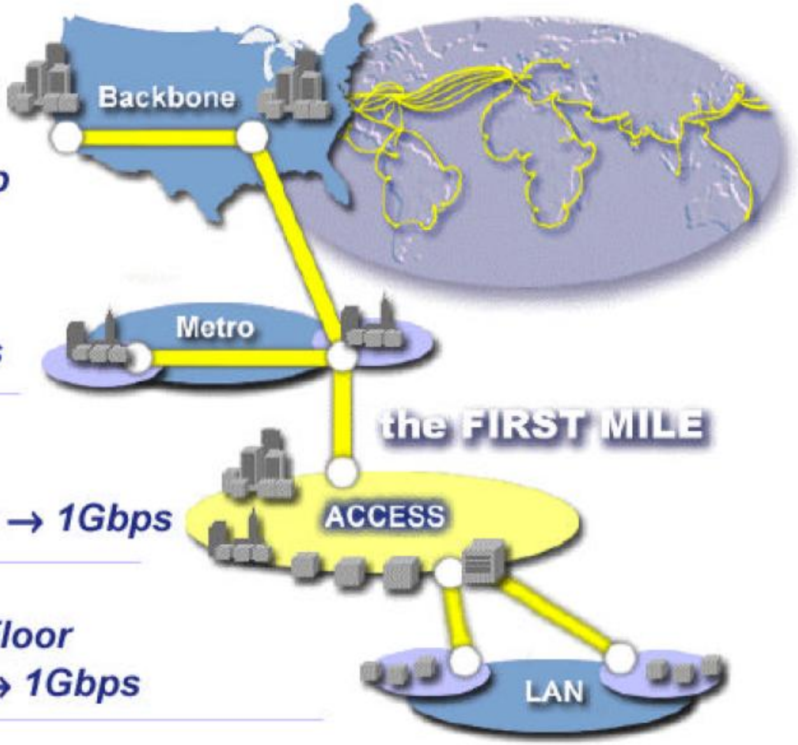
***New World Order***

**Backbone** *Continent to Continent*
*Coast to Coast*
*all over Fiber at 10 Gbps & up*

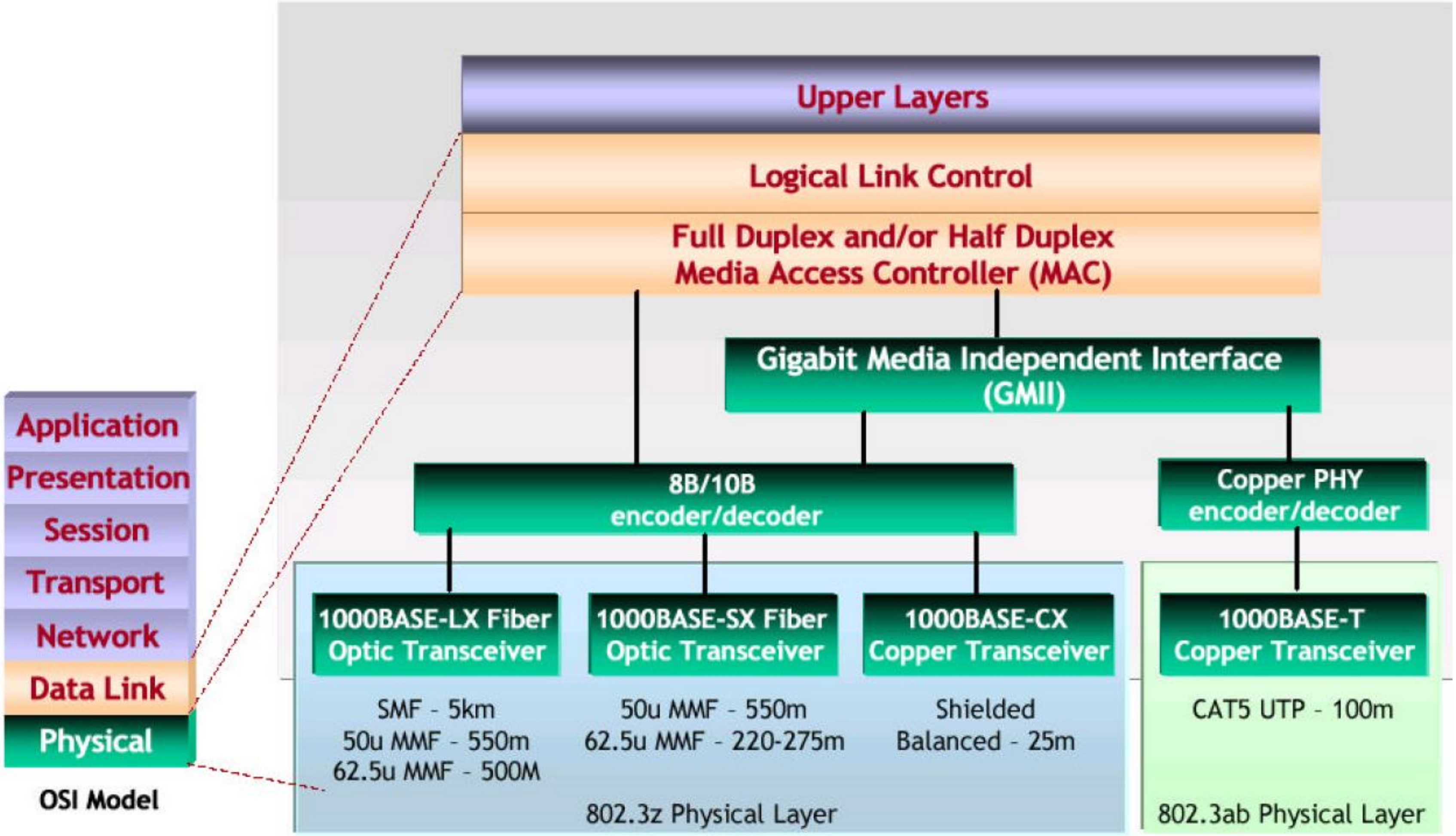
**Metro** *City to City – Town to Town*
*all over Fiber at 1Gbps → 10 Gbps*

**Access** *to the Optically Fibered World*
*"First Mile / Last Mile" 56 kbps → 1Gbps*

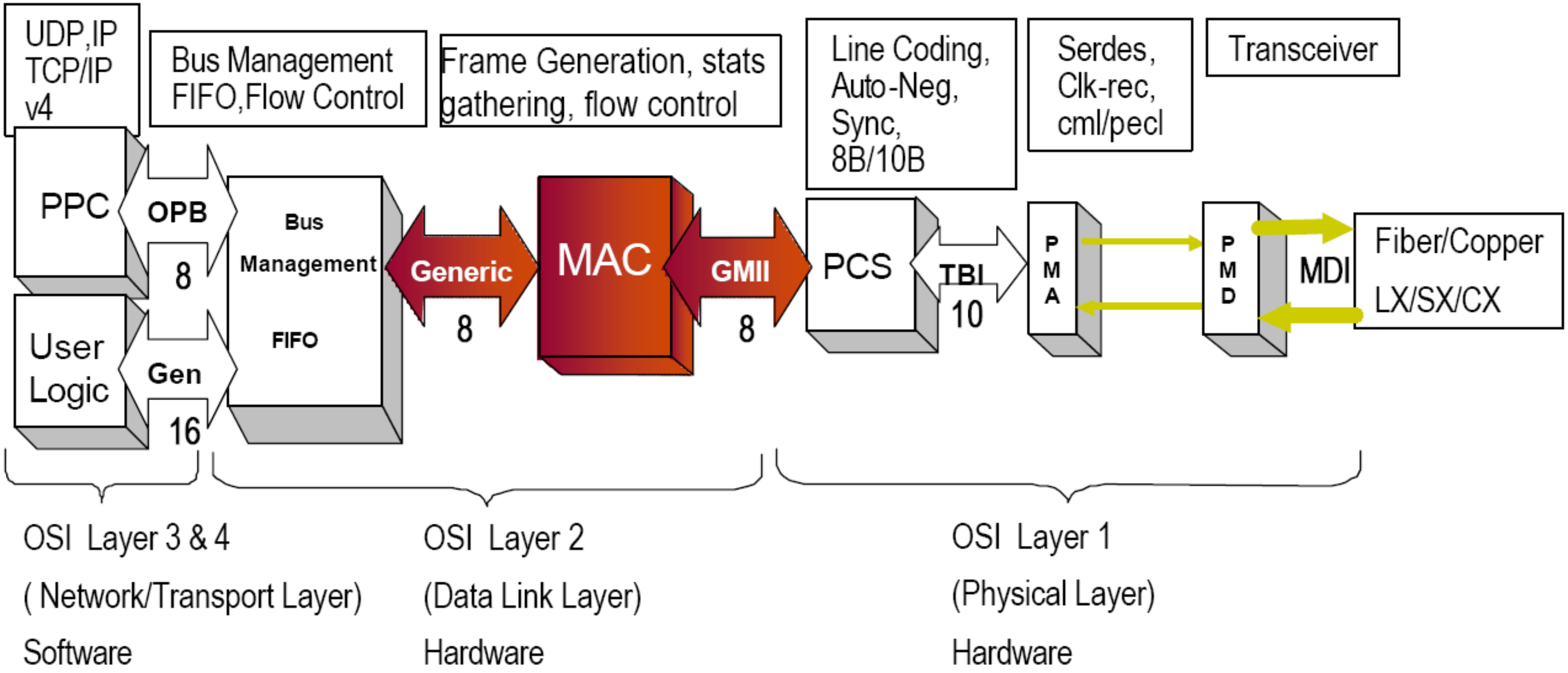
**LAN** *Desktop to Desktop – Floor to Floor*
*10 Mbps → 1Gbps*



# Gigabit Ethernet OSI model



# Building a 1000Base-X Ethernet Svstem



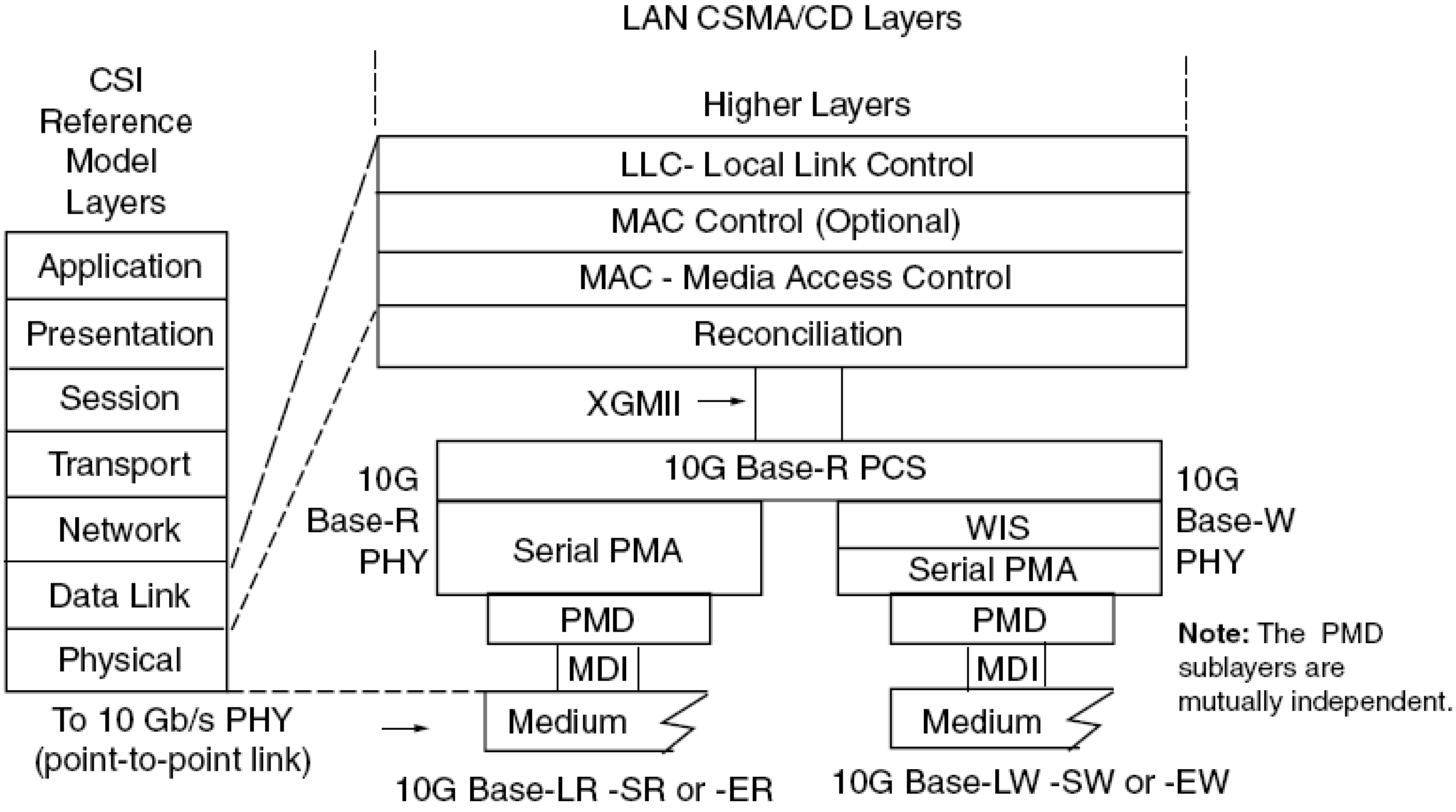
MDI - Medium Dependent Interface
PCS - Physical Coding Sublayer
PMA - Physical Medium Attachment
PMD - Physical Medium Dependent
GMII - Gigabit Media Independent Interface

# GbE vs. 10 GbE

- Gigabit Ethernet (802.3z)
  - Copper & Optical Fiber media only
  - Half (CSMA/CD) & Full Duplex only
  - Carrier extension & frame burst
  - 8b/10b coding scheme
  - Leverage Fibre Channel PMDs
  - Up to 5 km distance

- 10 Gigabit Ethernet (802.3ae)
  - Optical Fiber media only
  - Full Duplex only
  - Throttle MAC speed
  - New coding scheme (64b/66b)
  - New optical PMDs
  - Up to 40 km distance
  - Direct attachment to SONET/SDH gear

# 10GbE Layer Model



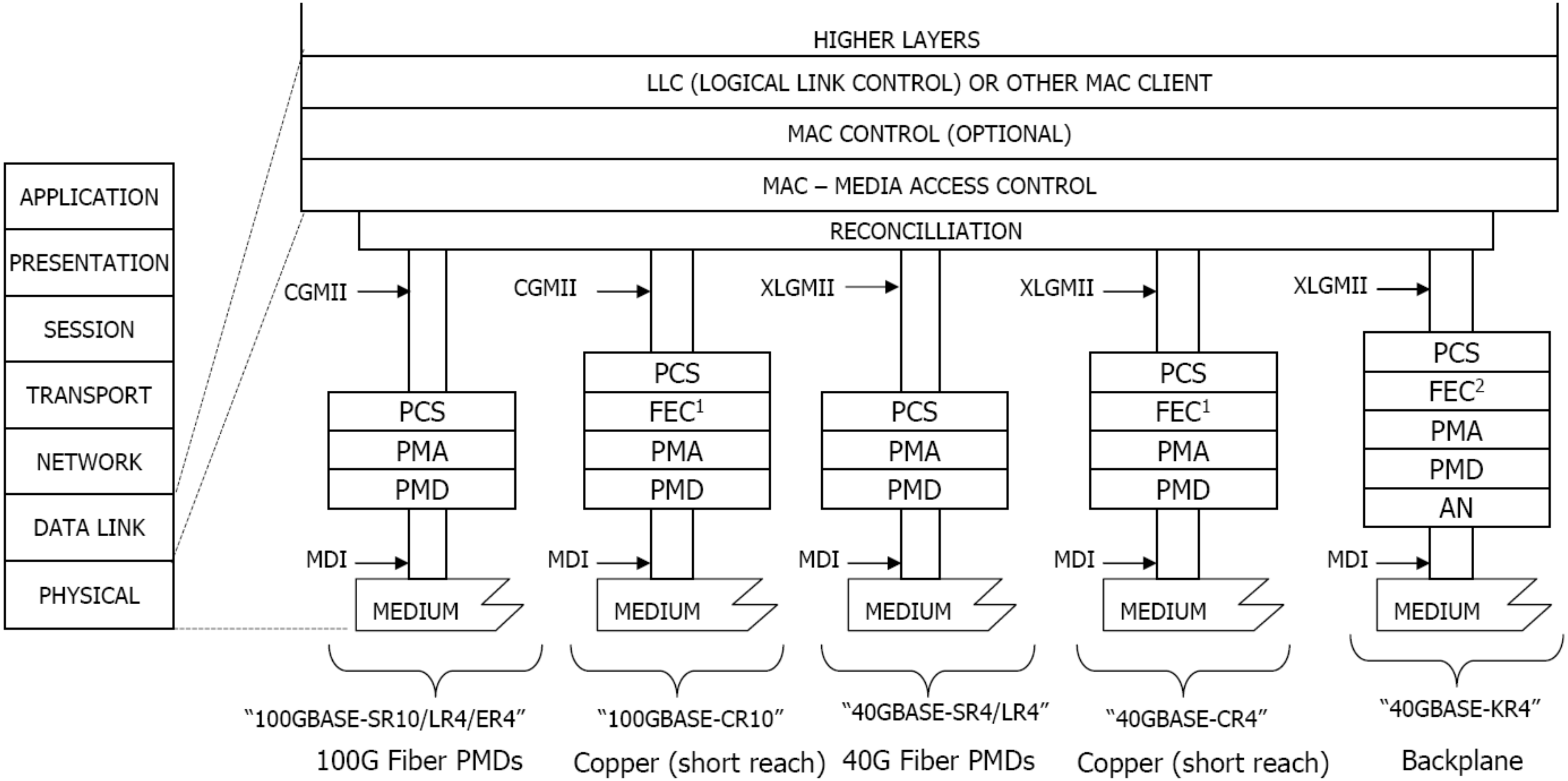
Medium:
E - PMD for Fiber = 1550 nm Wavelength
L - PMD for Fiber = 1310 nm Wavelength
S - PMD for Fiber = 850 nm Wavelength

Encoding:
R - 64B/66B Encoding without WIS
W - 64B/66B Encoding with WIS

# 10 GbE Components

- PHY standards
  - A LAN PHY operating at a data rate of 10.3125 Gbps
  - A WAN PHY operating at a data rate of 9.95328 Gbps
    - Compatible with OC-192c/SDH VC-4-64c payload rates
- Optical transceivers - PMD interfaces
  - 850nm Serial, MMF, 65M
  - 1310nm WWDM, MMF (300M), SMF (10km)
  - 1310nm Serial, SMF 10km
  - 1550nm Serial, SMF 40km

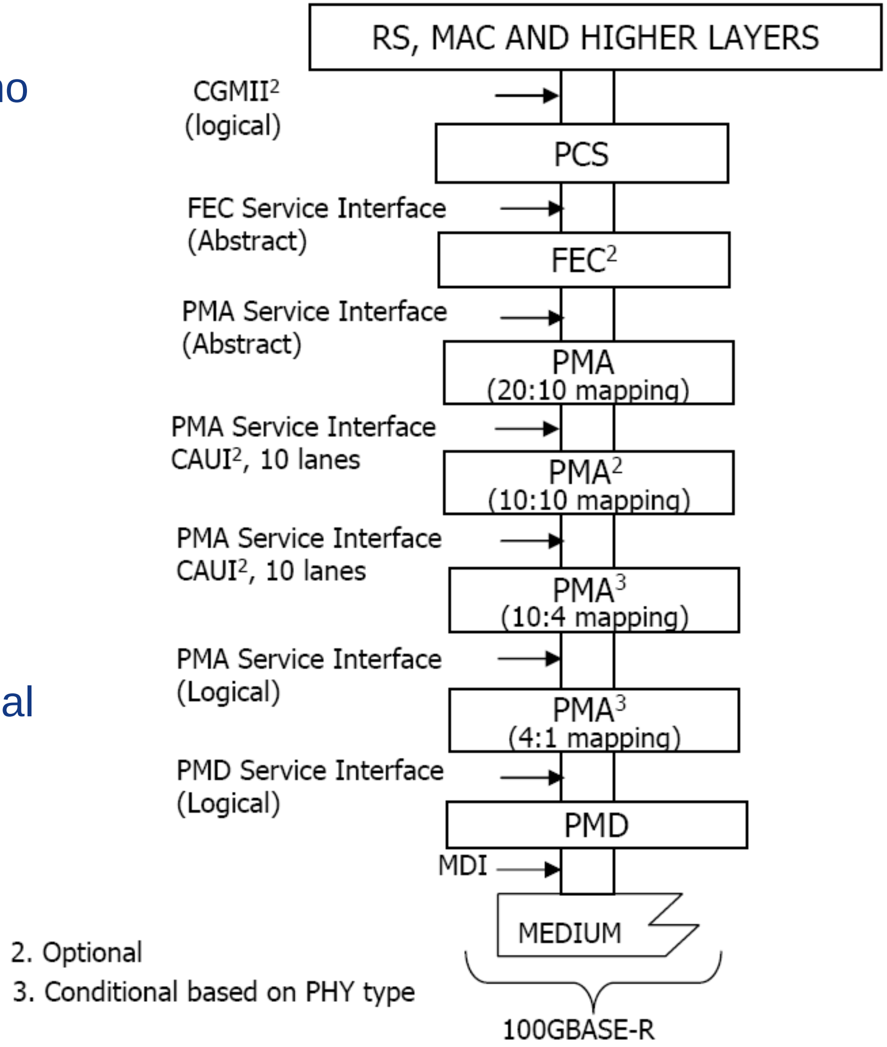
# 40/100GbE Layer Model



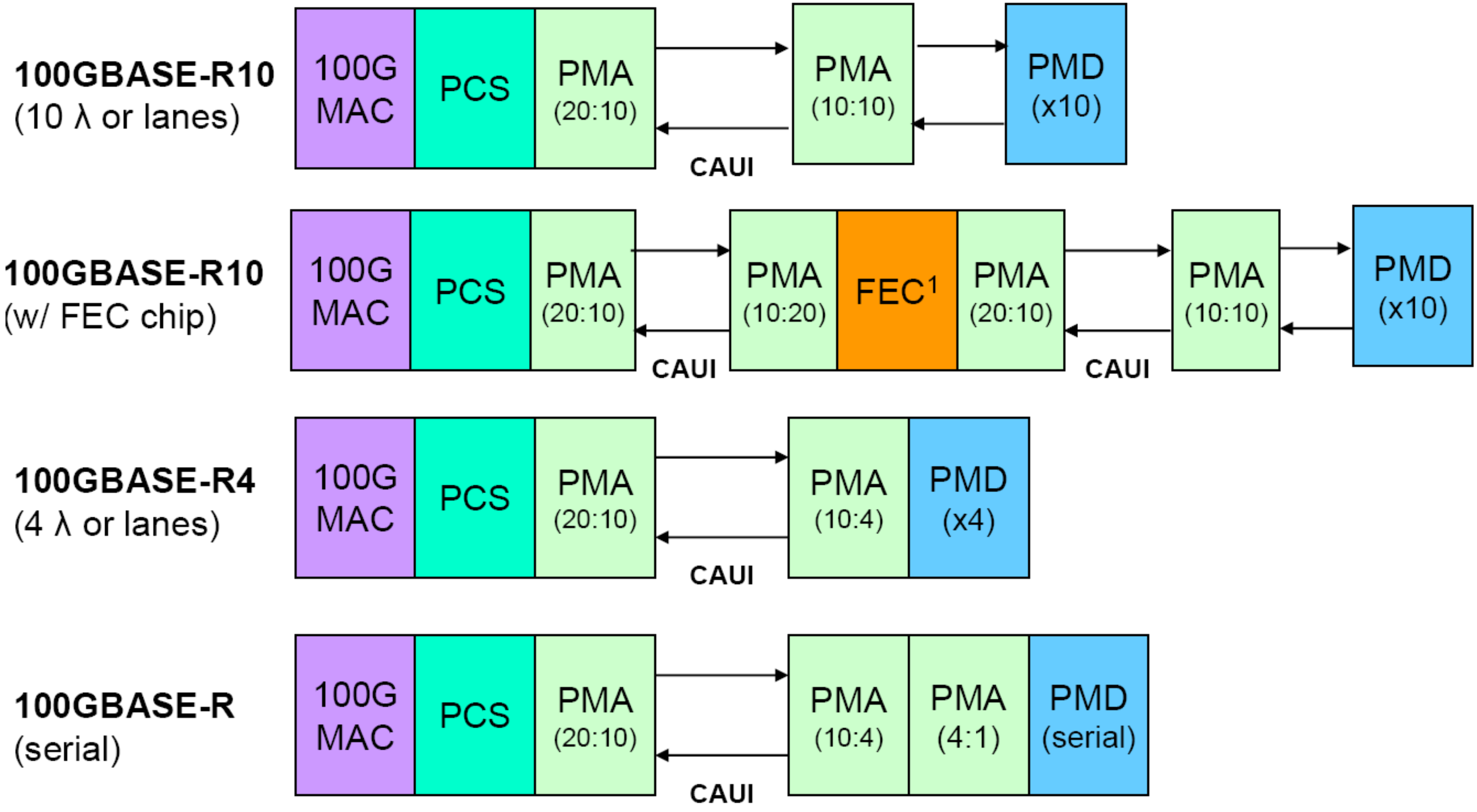
Note: 1. CR4 & CR10 may use optional FEC
2. Optional

# Proposed 100GbE architecture

- XLGMII (intra-chip)
  - Logical, define data/control, clock, no electrical specification
- PCS
  - 64B/66B encoding
  - Lane distribution and alignment
- XLAUI (chip-to-chip)
  - 10.3125 GBaud electrical interface
  - 4 lanes, short reach
- FEC service interface
  - Abstract, can map to XLAUI electrical interface
- PMA Service interface
  - Logical n lanes, can map to XLAUI electrical interface
- PMD Service interface
  - Logical


RS, MAC AND HIGHER LAYERS
CGMII2 (logical)
PCS
FEC Service Interface (Abstract)
FEC2
PMA Service Interface (Abstract)
PMA (20:10 mapping)
PMA Service Interface CAUI2, 10 lanes
PMA2 (10:10 mapping)
PMA Service Interface CAUI2, 10 lanes
PMA3 (10:4 mapping)
PMA Service Interface (Logical)
PMA3 (4:1 mapping)
PMD Service Interface (Logical)
PMD
MDI
MEDIUM
2. Optional
3. Conditional based on PHY type
100GBASE-R

# Possible 100GbE implementations



Note: 1. CR10 may use optional FEC