



南京大學

NANJING UNIVERSITY



Computer Networks

Wenzhong Li, Chen Tian

Nanjing University

Material with thanks to James F. Kurose, Mosharaf Chowdhury, and other colleagues.



Chapter 2. Direct Link Networks

- Link Layer Service
 - Framing
 - Link access
 - Reliable delivery
 - Error detection and correction
- Local Area Network (LAN)
 - Token Ring
 - Ethernet
- Medium access control
- Bridges and Layer-2 switch
- Wireless Networks



Bridges and Layer-2 switch

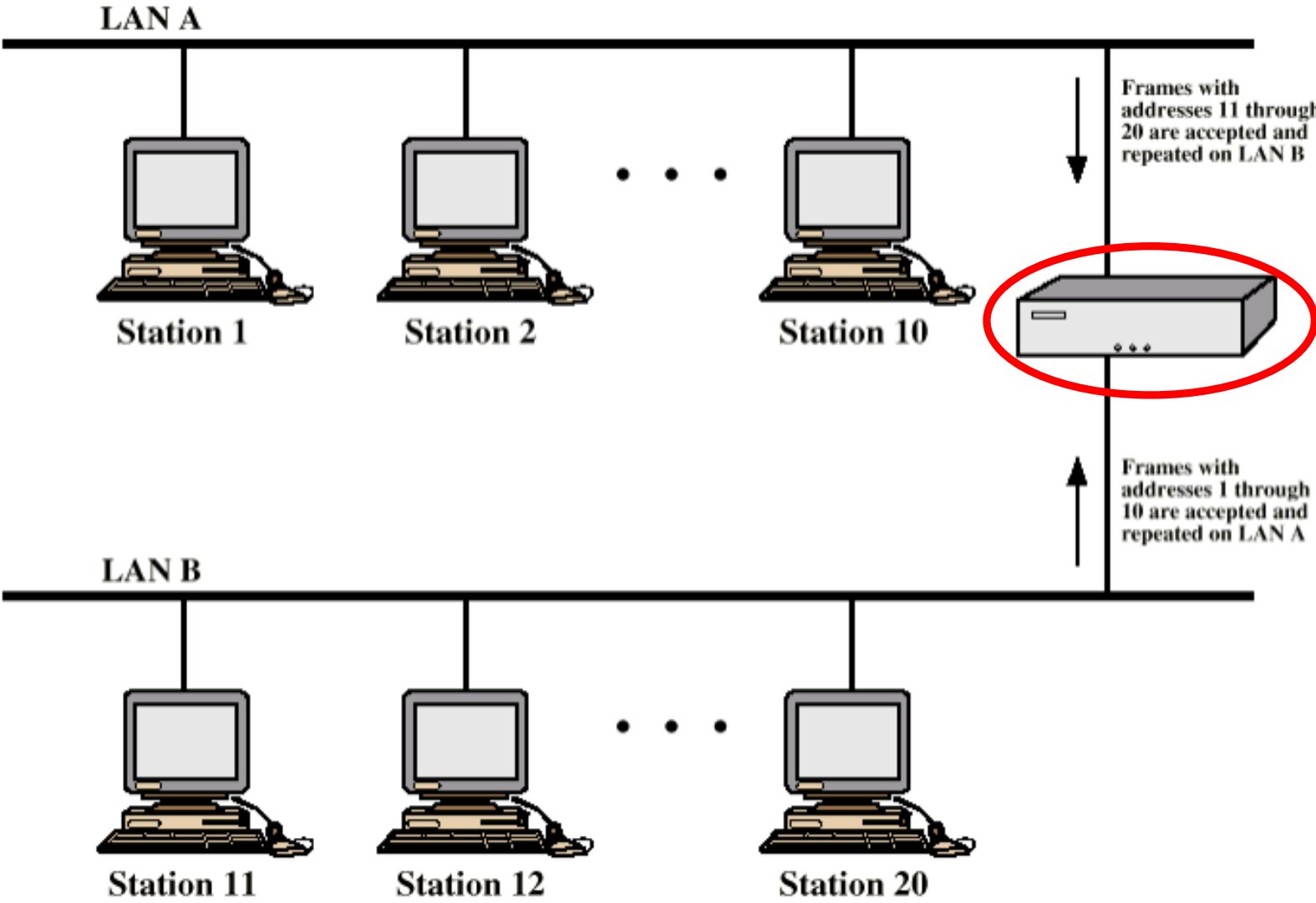


Interconnection of LANs

- Ability to **expand beyond single LAN**
- Provide interconnection to other LANs/WANs
- **Bridge** is used (later **Layer-2 switch**)
 - Connects LANs, usually **more than two LANs**
 - Identical protocols for physical and MAC layers
 - **Store, forward** LAN frames
 - Exact bitwise copy of frame
 - **Switch (route) functions** needed



Bridge Operation





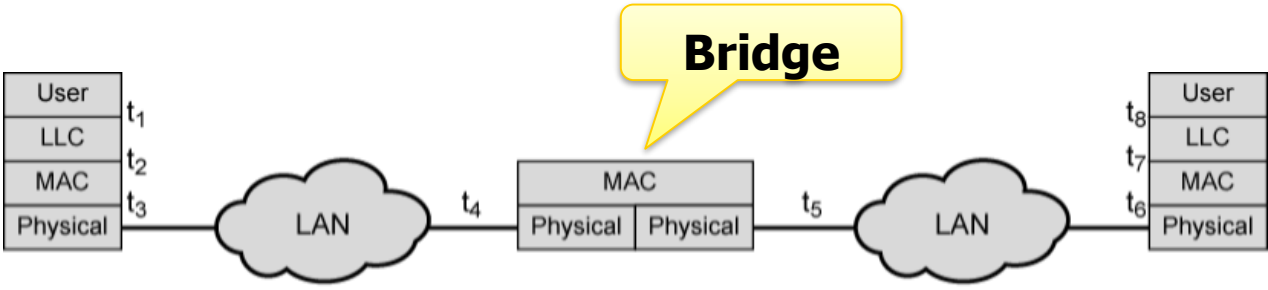
Requirements of a Bridge

- **Store and Forward**
 - Read frames transmitted on one LAN, Examine frames' MAC address, **selectively store** those address to other LANs
 - Using MAC protocol of second LAN, **retransmit** each frame
- **Transparent**
 - Stations are unaware of presence of bridges
- **Plug-and-play, self-learning**
 - Bridges do not need to be configured



Bridge Protocol Architecture

- IEEE 802.1D, MAC level
 - Use **MAC address** for switching
- Relaying MAC frames



(a) Architecture

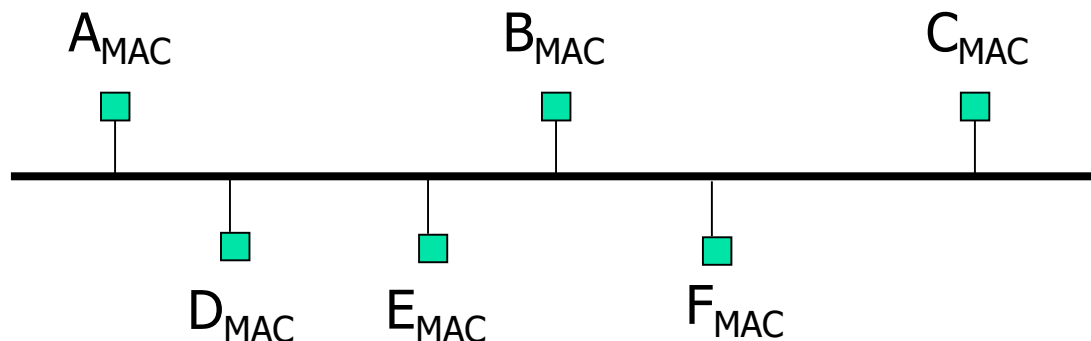


Mechanisms of a Bridge

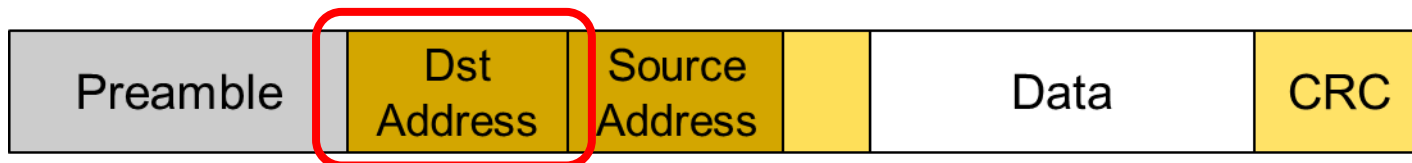
- Mechanisms
 - Transparent bridge: 802.1D
 - Frame broadcasting
 - Loop resolution
 - Address learning



Broadcast Ethernet

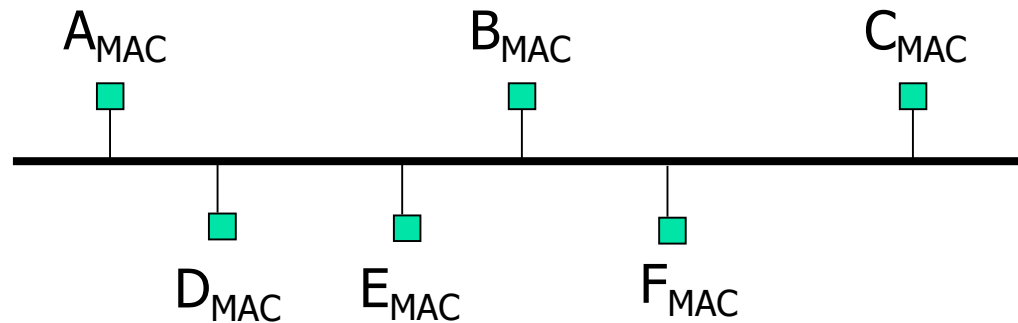


- Sender transmits frame onto broadcast link
- Each receiver's link layer passes the frame to the network layer:
 - If destination address matches the receiver's MAC address OR if the destination address is the broadcast MAC address (ff:ff:ff:ff:ff:ff)





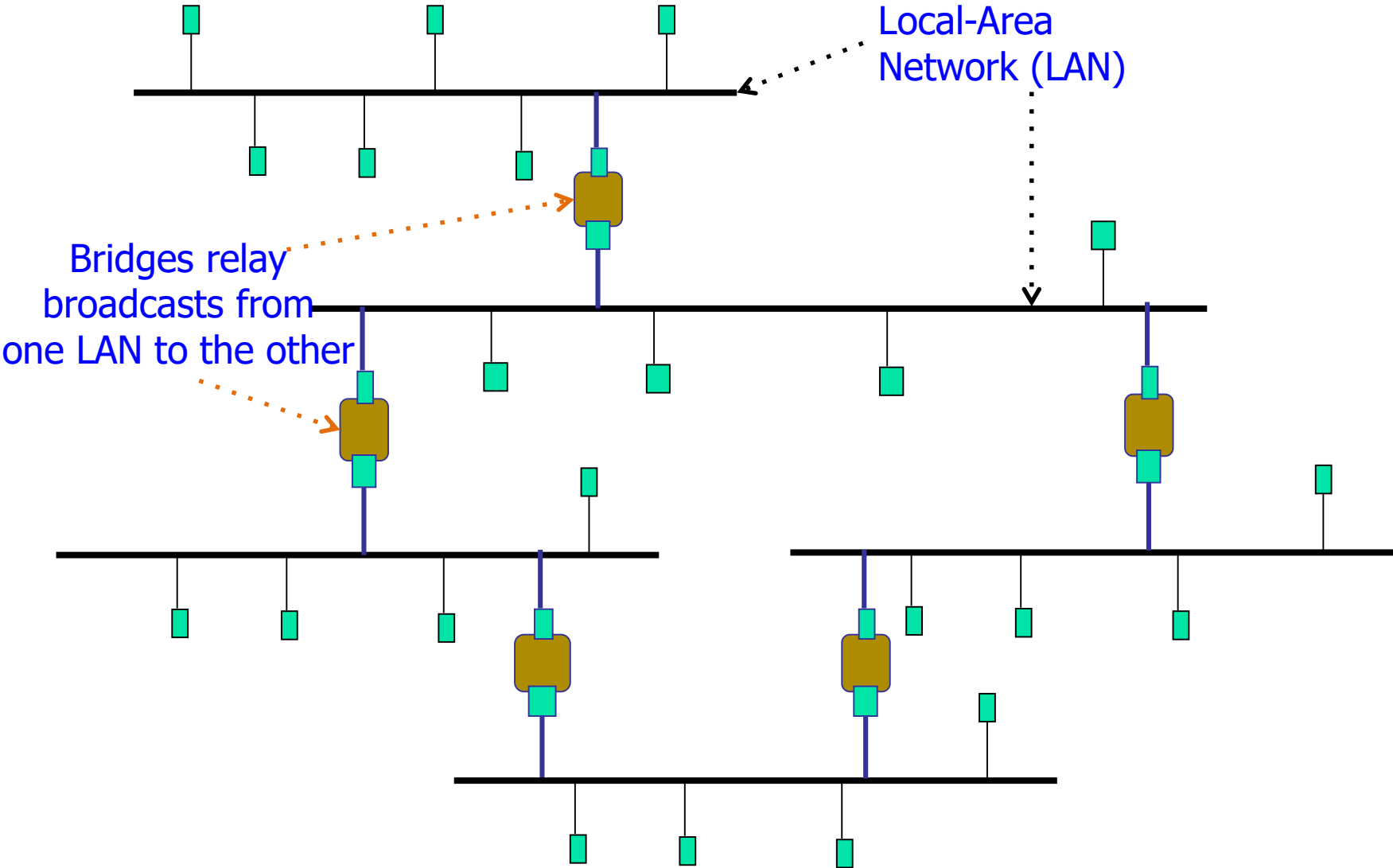
Broadcast Ethernet



- Ethernet is “plug-n-play”
- A new host plugs into the Ethernet and is good to go
 - No configuration by users or network operators
 - Broadcast as a means of bootstrapping comm.



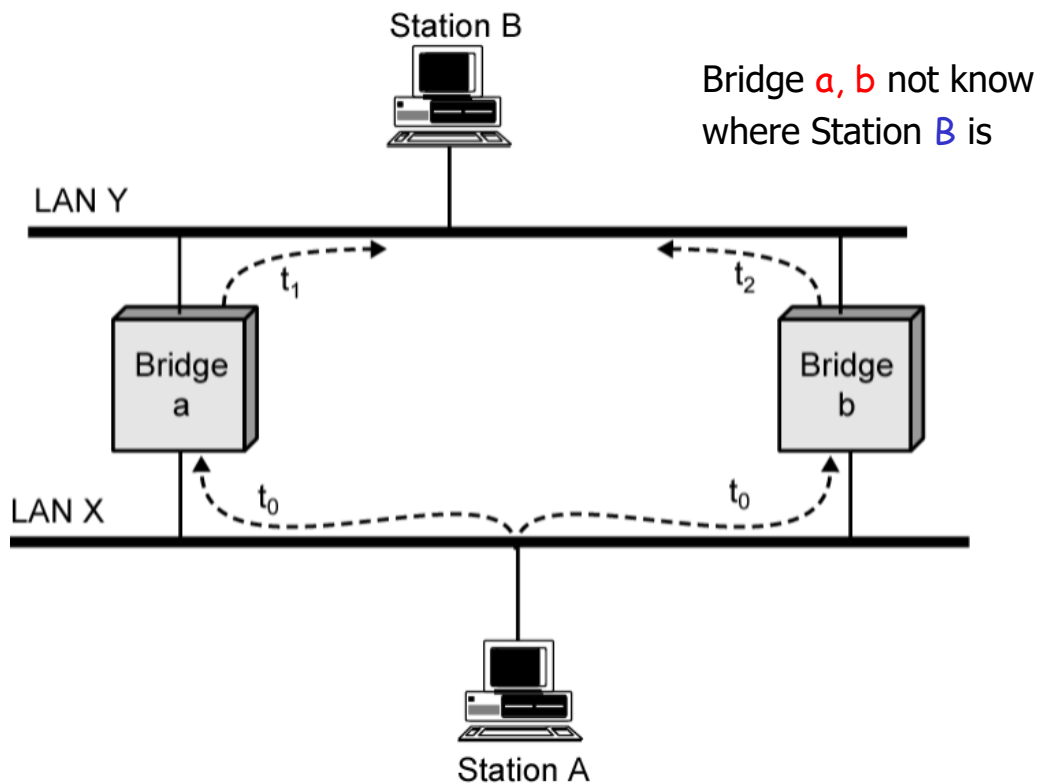
Broadcasting in extended LANs





What causes “broadcast storm”?

Loops!



Radia Perlman

Perlman's idea: eliminate loops in the topology

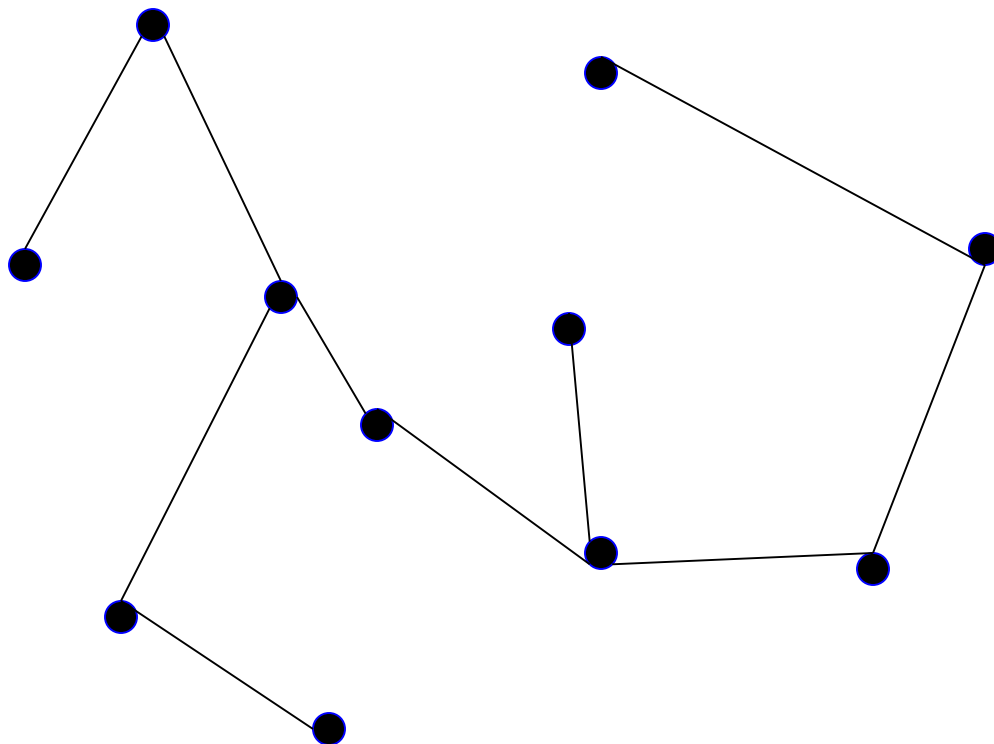


Easiest way to avoid loops

- Use a topology where loops are impossible!
- Take arbitrary topology and build a **spanning tree**
 - Sub-graph that includes all vertices but contains no cycles
 - Links not in the spanning tree are not used to forward frames

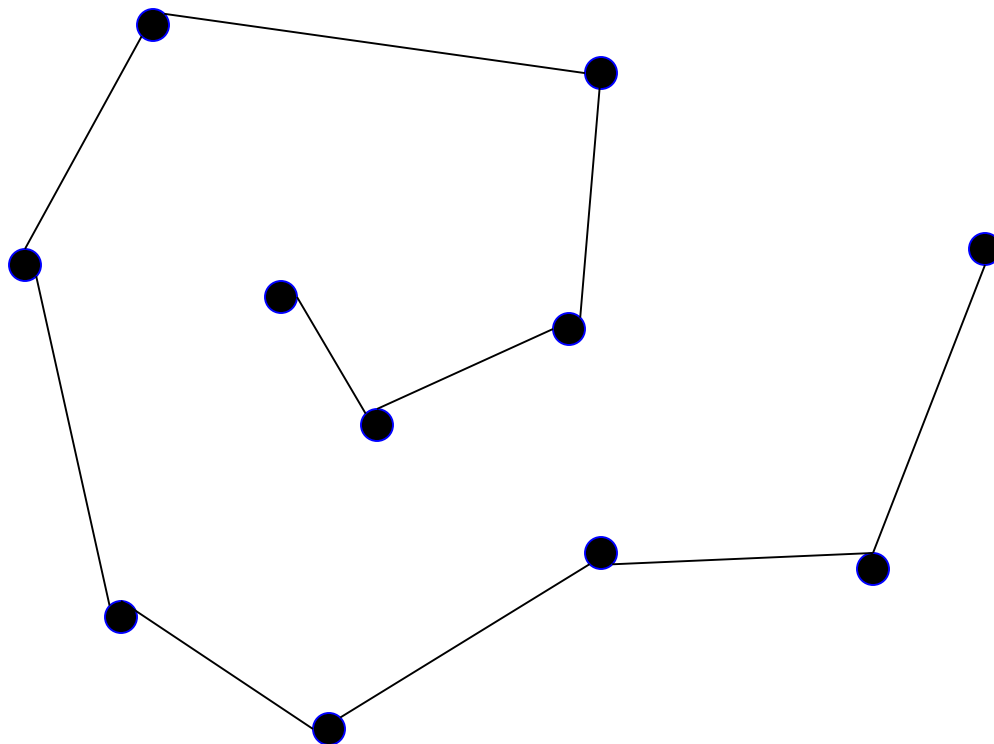


A spanning tree



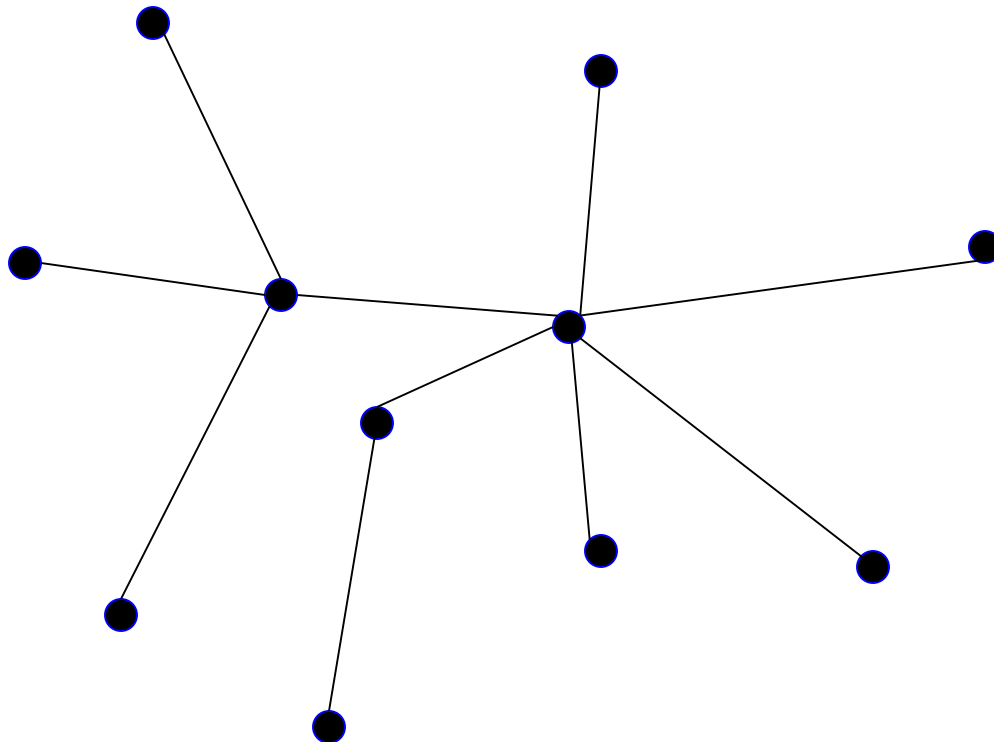


Another spanning tree





Yet another spanning tree





Spanning tree protocol (Perlman'85)

- Protocol by which bridges construct a spanning tree
- Nice properties
 - Zero configuration (by operators or users)
 - Self healing
- Still used today



Spanning tree algorithm

- Take arbitrary network topology as input
- Pick subset of links that form a spanning tree



Algorithm has two aspects

- Pick a root
 - Destination to which shortest paths go
 - Pick the one with the smallest identifier (MAC addr.)
- Compute shortest paths to the root
 - No shortest path can have a cycle
 - Only keep the links on shortest-paths
 - Break ties in some way (so we only keep one shortest path from each node)
- Ethernet's spanning tree construction does both with a single algorithm



Breaking ties

- When there are multiple shortest paths to the root, choose the path that uses the neighbor with the lower ID
 - One could use any tiebreaking system, but this is an easy one to remember and implement



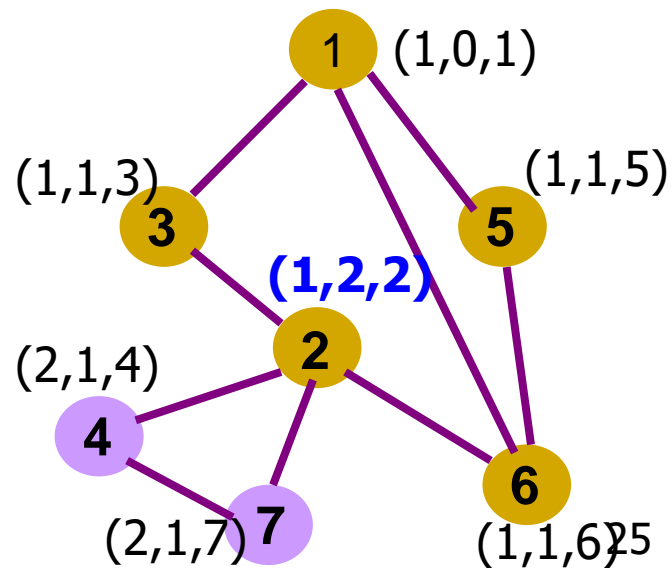
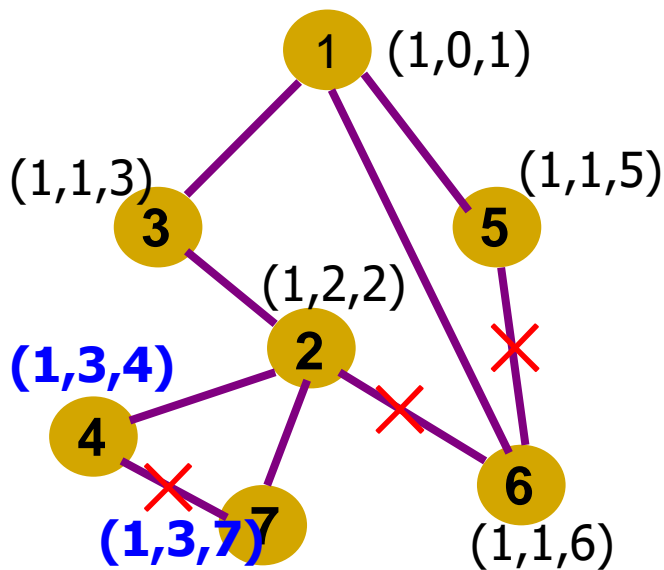
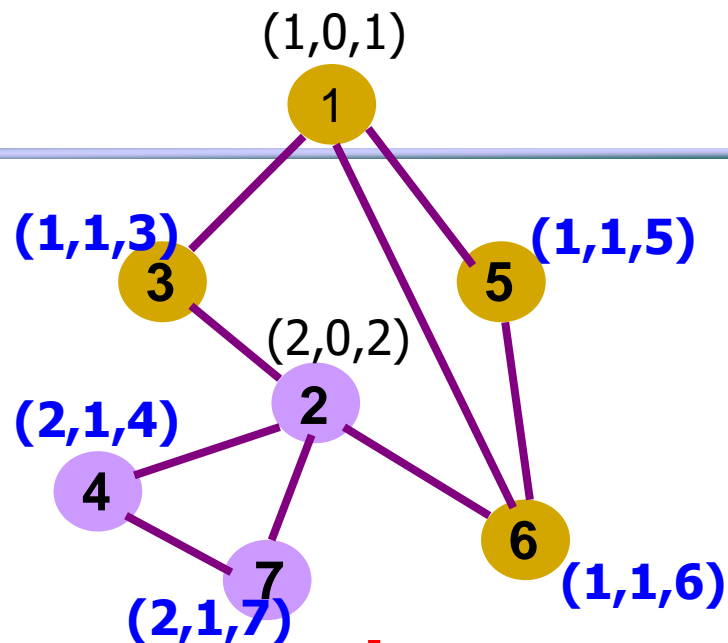
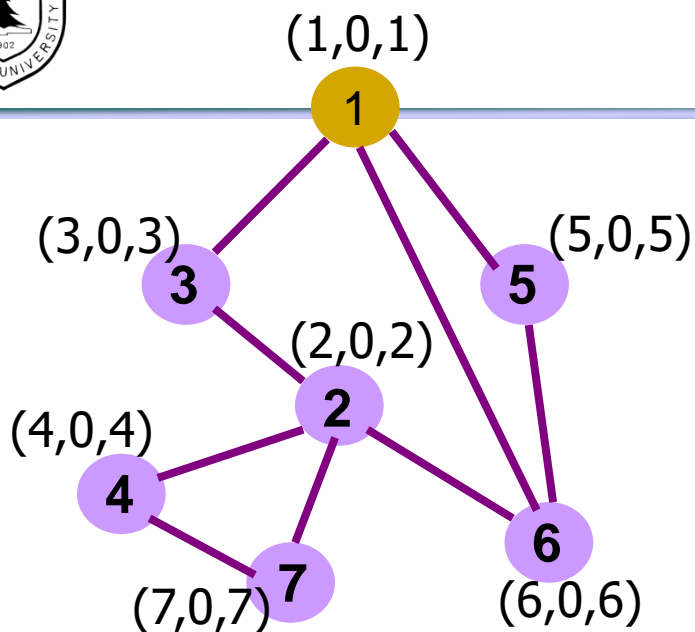
Constructing a spanning tree

- Messages (Y, d, X)
 - From node X
 - Proposing Y as the root
 - And advertising a distance d to Y
- Switches elect the node with smallest identifier (MAC address) as root
- Each node determines if a link is on its shortest path to the root; excludes it from the tree if not



Steps in the spanning tree algorithm

- Initially, each node proposes itself as the root
 - Switch X announces $(X, 0, X)$ to its neighbors
- Nodes update their view of the root
 - Upon receiving (Y, d, Z) from Z , check Y 's id
 - If Y 's id $<$ current root: set root = Y
- Nodes compute their distance from the root
 - Add 1 to the shortest distance received from a neighbor
- If root or shortest distance to it **changed**, send neighbors updated message $(Y, d+1, X)$





Robust spanning tree algorithm

- Algorithm must react to failures
 - Failure of the root node
 - Failure of other bridges and links
- Root sends periodic root announcement messages
 - Other bridges continue forwarding messages
- Detecting failures through timeout
 - If no word from root, time out and claim to be the root!

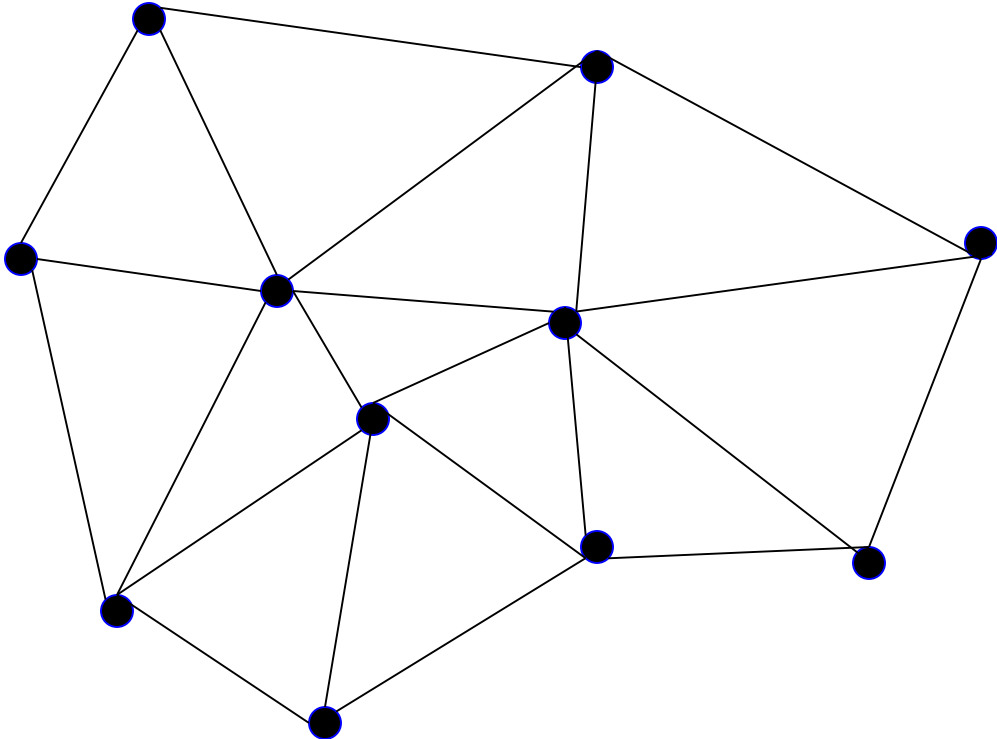


Flooding on a spanning tree

- Bridges flood using the following rule:
 - (Ignore all ports not on spanning tree!)
 - Originating bridge sends packet out all ports
 - When a packet arrives on one incoming port, send it out all ports other than the incoming port

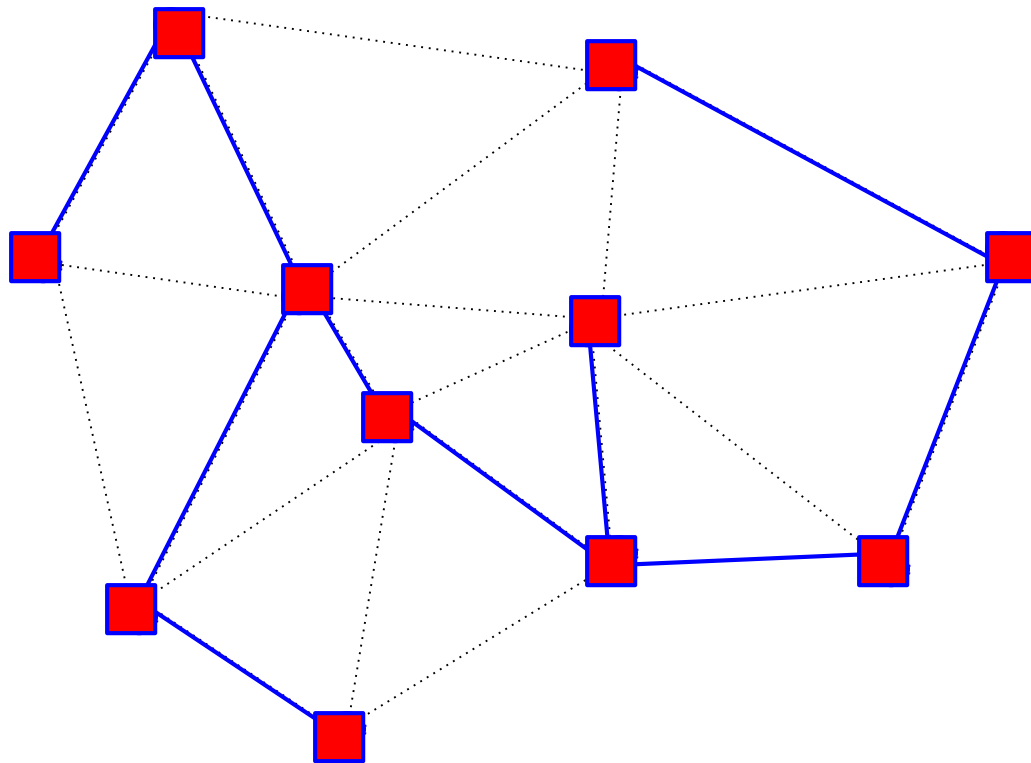


Flooding on spanning tree





Flooding on spanning tree





But isn't flooding wasteful?

- Yes, but we can use it to bootstrap more efficient forwarding
- **Idea**: watch the packets going by, and learn from them
 - If node A sees a packet from node B come in on a particular port, it knows what port to use to reach B!
 - **Works because there is only one path to B**



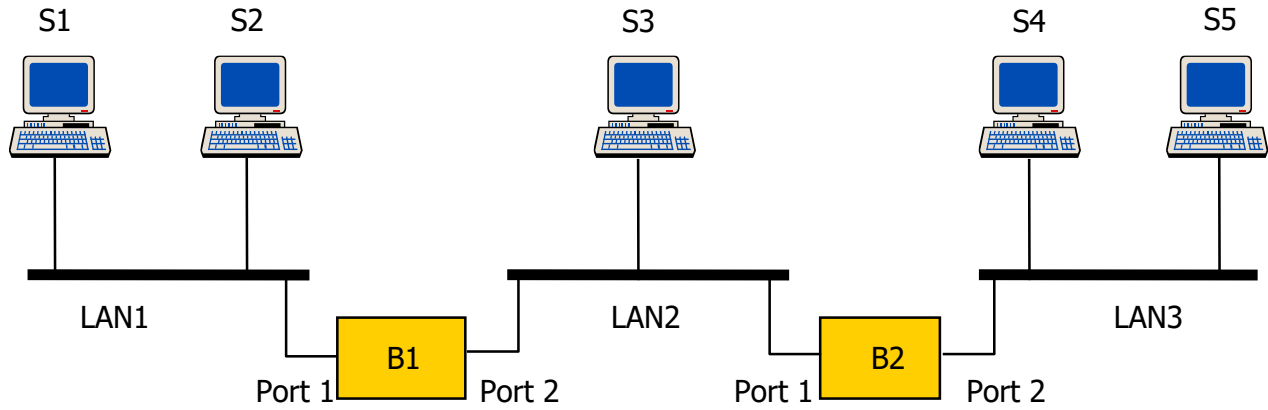
Nodes can “learn” routes

- Switch learns how to reach nodes by remembering where flooding packets came from
 - If flood packet from Node A entered bridegroom on port 4, then bridge uses port 4 to send to Node A



Address Learning

- Each bridge maintains a **forwarding database**
- **Forwarding database** can be **learned**
- 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
- **Timer** on each entry in database, Entry deleted when timer is off
- Each time frame arrives, **source address checked** against forwarding database

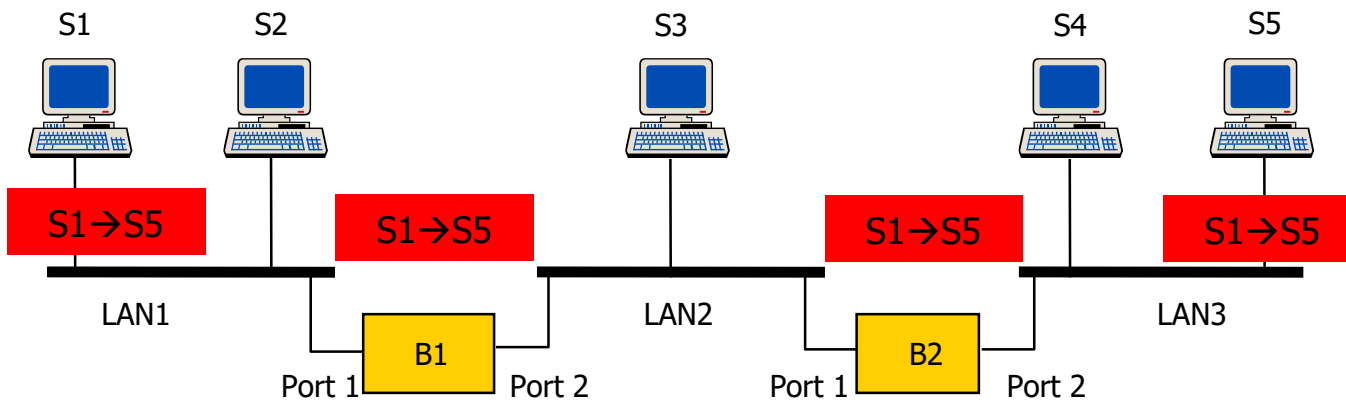


Address	Port

Address	Port



S1→S5

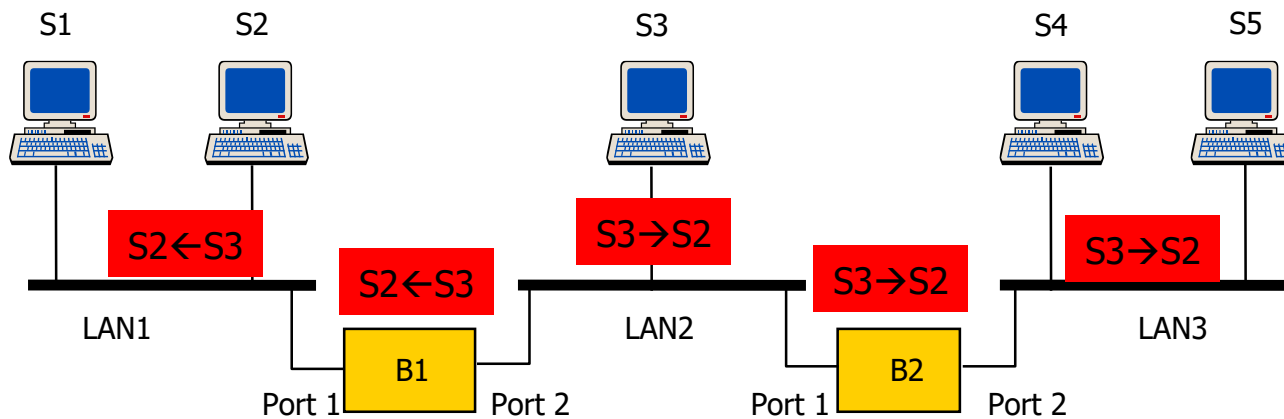


Address	Port
S1	1

Address	Port
S1	1



S3 → S2

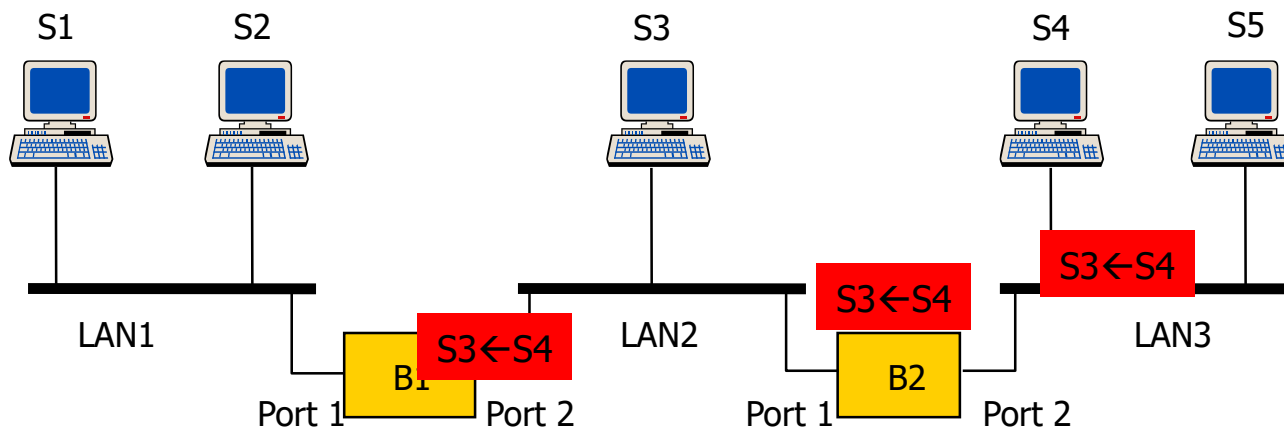


Address	Port
S1	1
S3	2

Address	Port
S1	1
S3	1



S4→S3

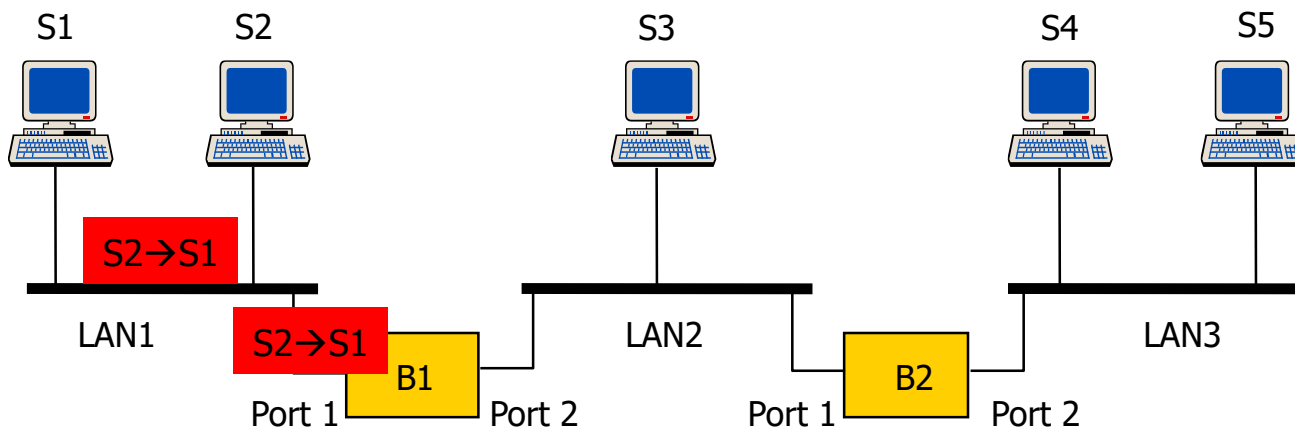


Address	Port
S1	1
S3	2
S4	2

Address	Port
S1	1
S3	1
S4	2



S2→S1



Address	Port
S1	1
S3	2
S4	2
S2	1

Address	Port
S1	1
S3	1
S4	2



Summary of bridge frame forwarding

- Maintain **forwarding database** for each port
 - <Mac Address, Port, Timestamp>: station addresses reached through each port
- For a frame arriving on **port X**
 - Search forwarding database to see if dest MAC address is listed for any port:
 - If address not found, **forward to all ports** (flooding) except X
 - If address listed for port X, drop it
 - If address listed for port Y, check port Y for **Blocking or Forwarding** state, transmit if **Forwarding state**
- **Blocking** used to form a tree



Types of devices for interconnecting LANs

- **Hubs:** physical repeaters
- **Bridges:** connecting LANs (forwarding + address learning)
- **Layer 2 switches:** connecting Hosts or LANs (bridge functions + collision free)
- **Layer 3 switches:** involving router functions

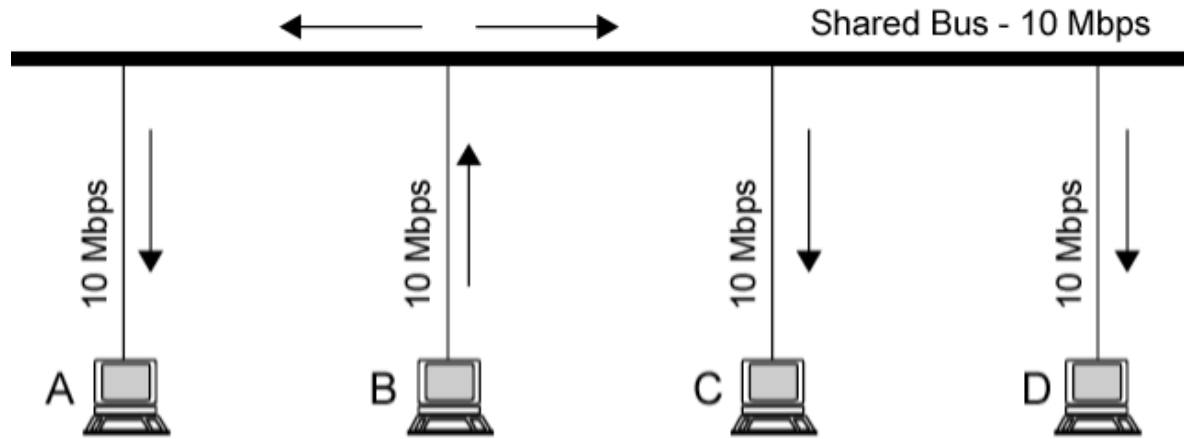


Hubs

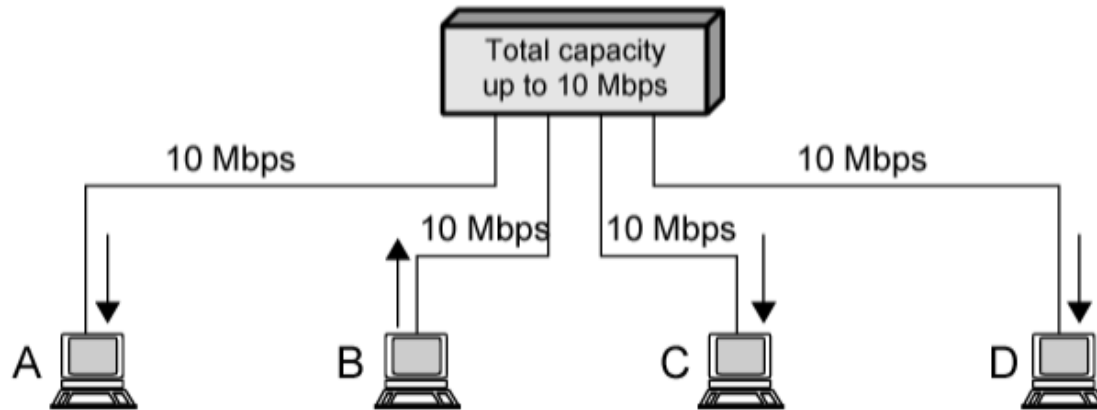
- Active central element of **star layout**
- Each station connected to hub by two lines
 - Transmit and receive
- Hub **acts as a repeater**
 - When a station transmits, hub repeats signal on outgoing line to all other stations
- **Physically star, logically bus**
 - Transmission from any station received by all other stations
 - If two stations transmit at the same time, collision



Shared Medium Bus and Hub



(a) Shared medium bus



(b) Shared medium hub



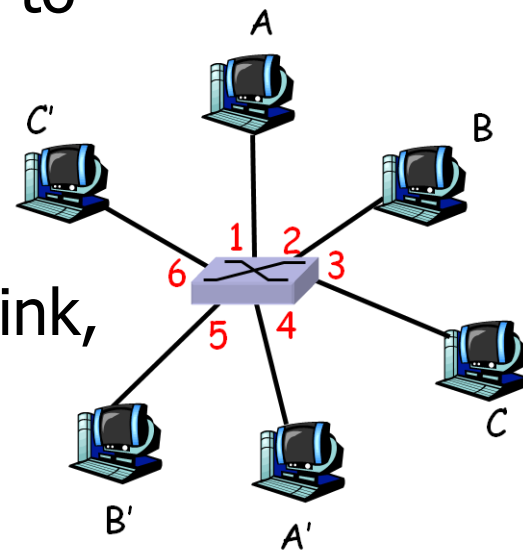
Layer 2 Switch: Requirement

- Link-layer device: takes an active role
 - Store, forward Ethernet frames
 - Examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
 - Hosts are unaware of presence of switches
- Plug-and-play, self-learning
 - Switches do not need to be configured
- Switch vs. Bridge
 - Bridge: connect LANs (normally 2-4 ports)
 - Switch: connect multiple hosts/subnets (a lot of ports, can achieve collision-free transmission)



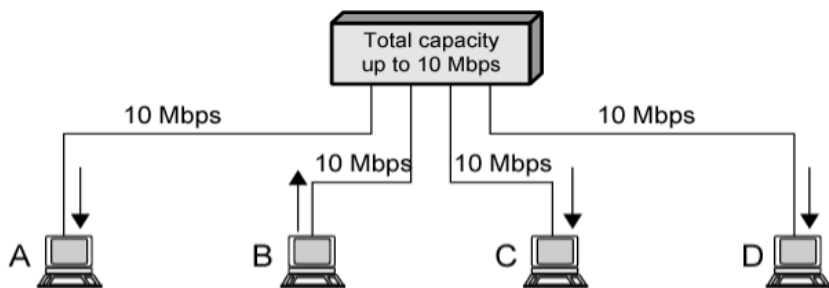
Why switched Ethernet?

- Stations have dedicated, direct connection to switch
- Switches buffer and forward frames
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
 - each link is its own collision domain
- **Switching:** A-to-A' and B-to-B' can transmit simultaneously, without collisions
- Multiplying capacity of LANs
 - Each port/link forms a LAN segment (**no collisions**)
 - **More than one station** transmitting at a time

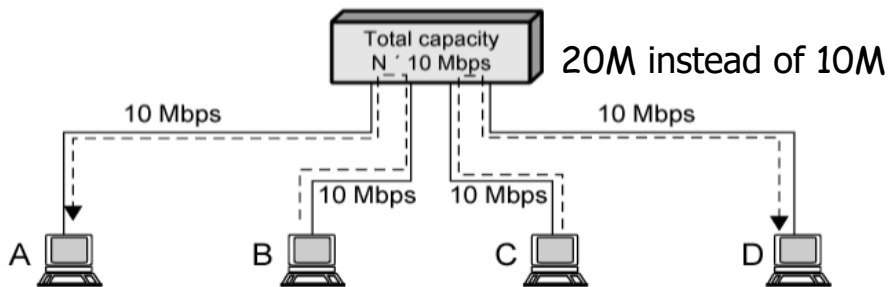




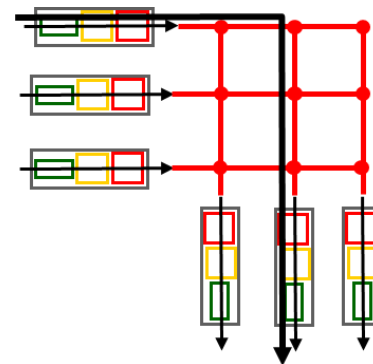
Shared Medium Hub and Layer 2 Switch



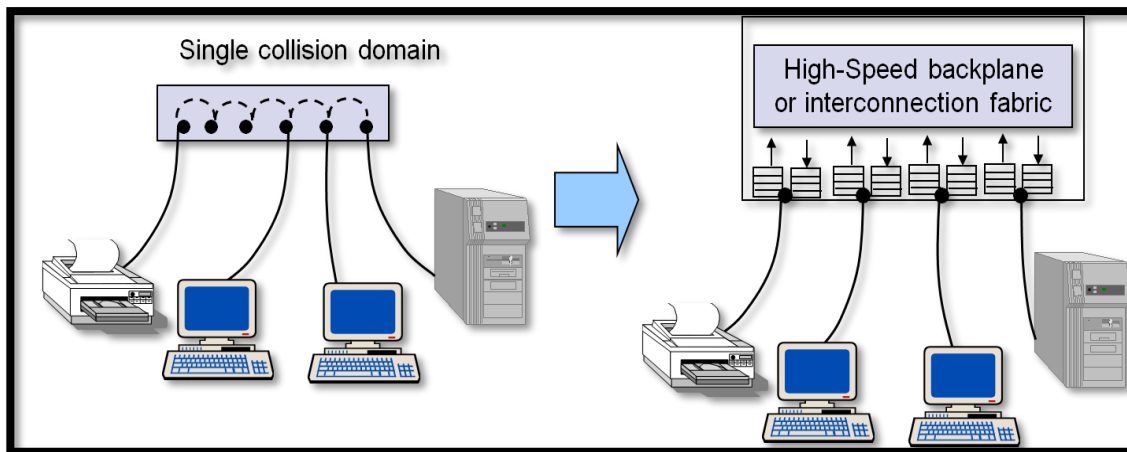
(b) Shared medium hub



(c) Layer 2 switch



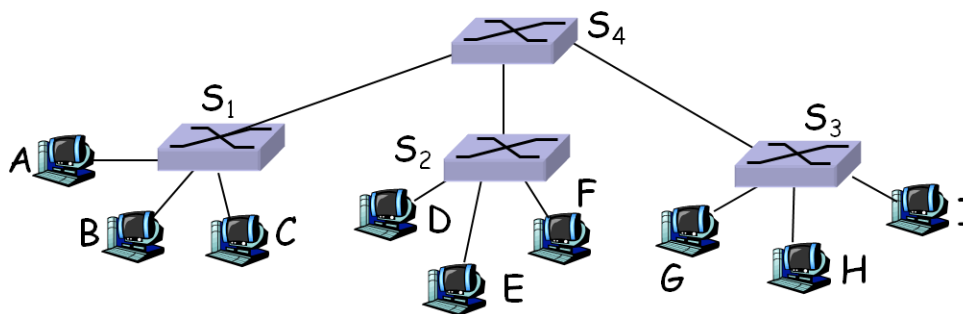
switching fabrics





Layer 2 Switch Benefits

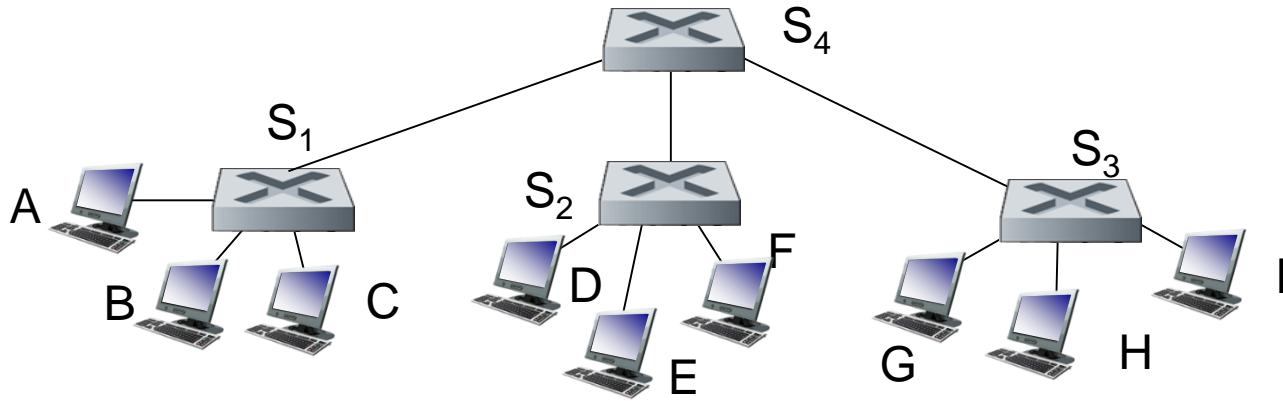
- No change to attached stations to convert bus/hub LAN to switched LAN
 - For Ethernet LAN, each station uses Ethernet MAC protocol
- Station has **dedicated capacity** equal to original LAN
 - Assuming switch has sufficient capacity to keep up with all stations
- Layer 2 switch **scales easily**
 - New layer 2 switch added to accommodate additional stations





Interconnecting switches

- ❖ Switches can be connected together

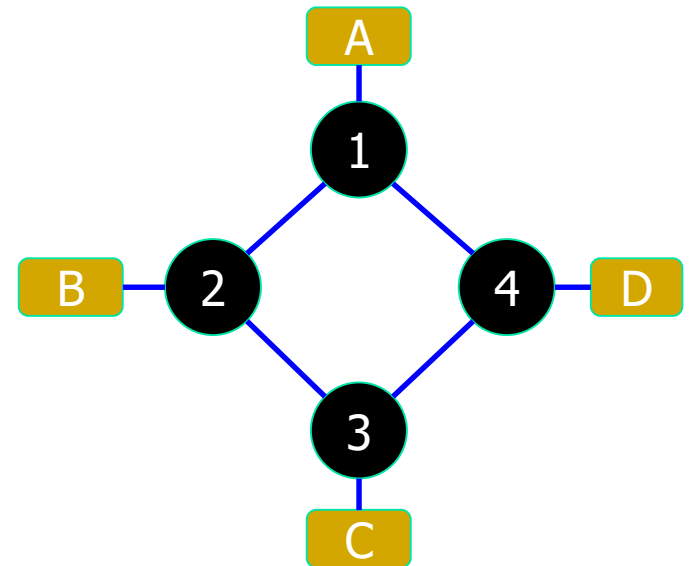




Loop resolution

- Example: A wants to broadcast a message

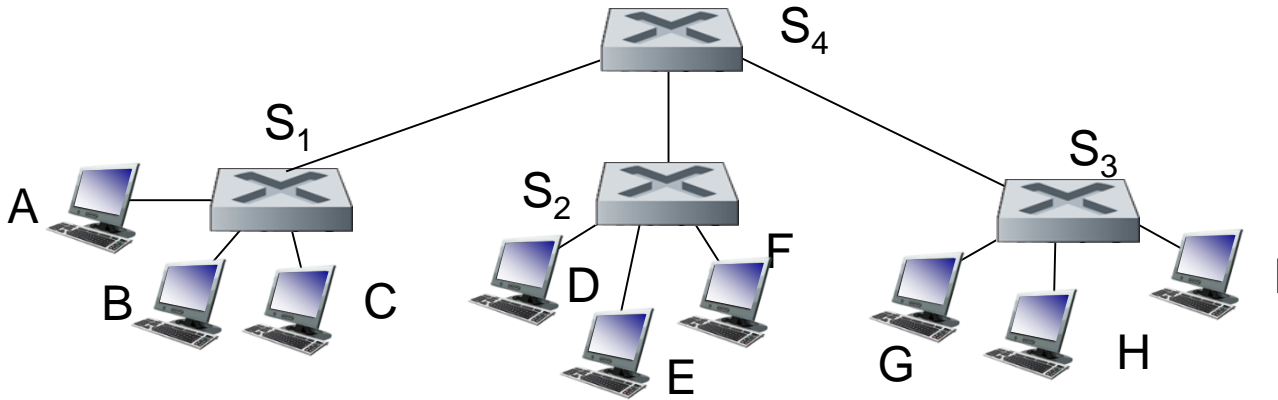
- A sends packet to 1
- 1 Floods to 2 and 4
- 2 Floods to B and 3
- 4 Floods to D and 3
- 3 Floods packet from 2 to C and 4
- 3 Floods packet from 4 to C and 2
- 4 Floods packet from 3 to D and 1
- 2 Floods packet from 3 to B and 1
- 1 Floods packet from 2 to A and 4
- 1 Floods packet from 4 to B and 2
-



- Broadcast storm still happens in a switched network if it contains a cycle of switches
- **Loop resolution: spanning tree algorithm**



Ethernet switches are “self learning”



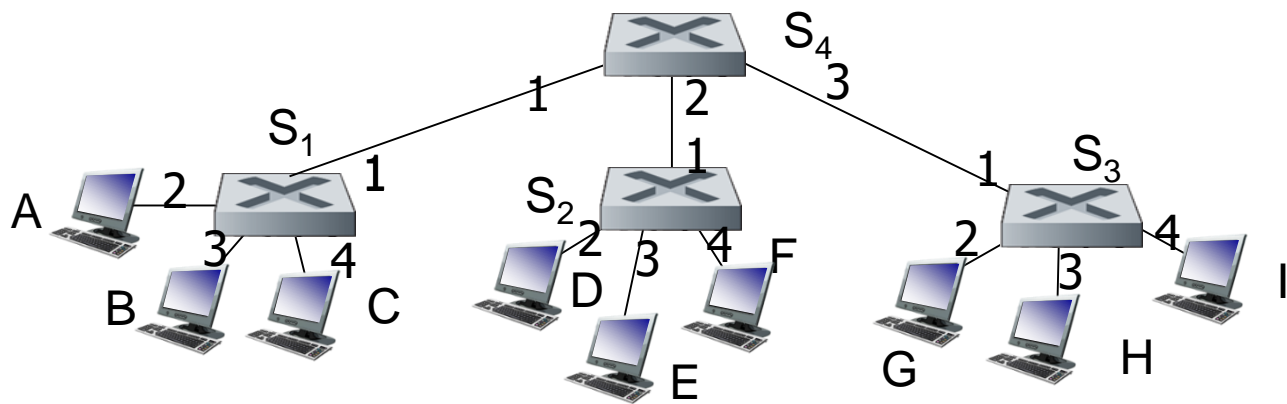
Q: sending from A to G - how does S₁ know to forward frame destined to G via S₄ and S₃?

❖ A: self learning! (works exactly the same as in single-switch case!)



Self-learning multi-switch example

Suppose C sends frame to I, I responds to C



- ❖ Q: show switch tables and packet forwarding in S_1, S_2, S_3, S_4 (assuming they are empty in the beginning)



Summary of switch frame forwarding

when frame received at switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
 then {
 if destination on segment from which frame arrived
 then drop frame
 else forward frame on interface indicated by entry
 }
 else flood /* forward on all interfaces except arriving
 interface */



Layer 3 Switches

- As number of stations grows, layer 2 switches reveal inadequacies
 - Broadcast overload
 - Lack of multiple paths
- Layer 3 switch
 - Implement packet-forwarding (IP) logic of a router in hardware
 - Interconnecting similar LANs, as in layer 2 switches



Broadcast Overload

- Set of stations and LANs connected by layer 2 switches builds singular physical net
- All nodes share **common MAC broadcast address**
- **Broadcast frame** delivered to all stations attached to LANs connected by layer 2 switches
- Under broadcast, layer 2 switches become hubs
- IP causes many broadcasts under daily work, e.g. ARP, DHCP, IGMP



Lack of Multiple Paths

- Layer 2 switches uses spanning tree for routing (switching)
- **Dictate no closed loops:** only one path between any two stations
- Limits both performance and reliability

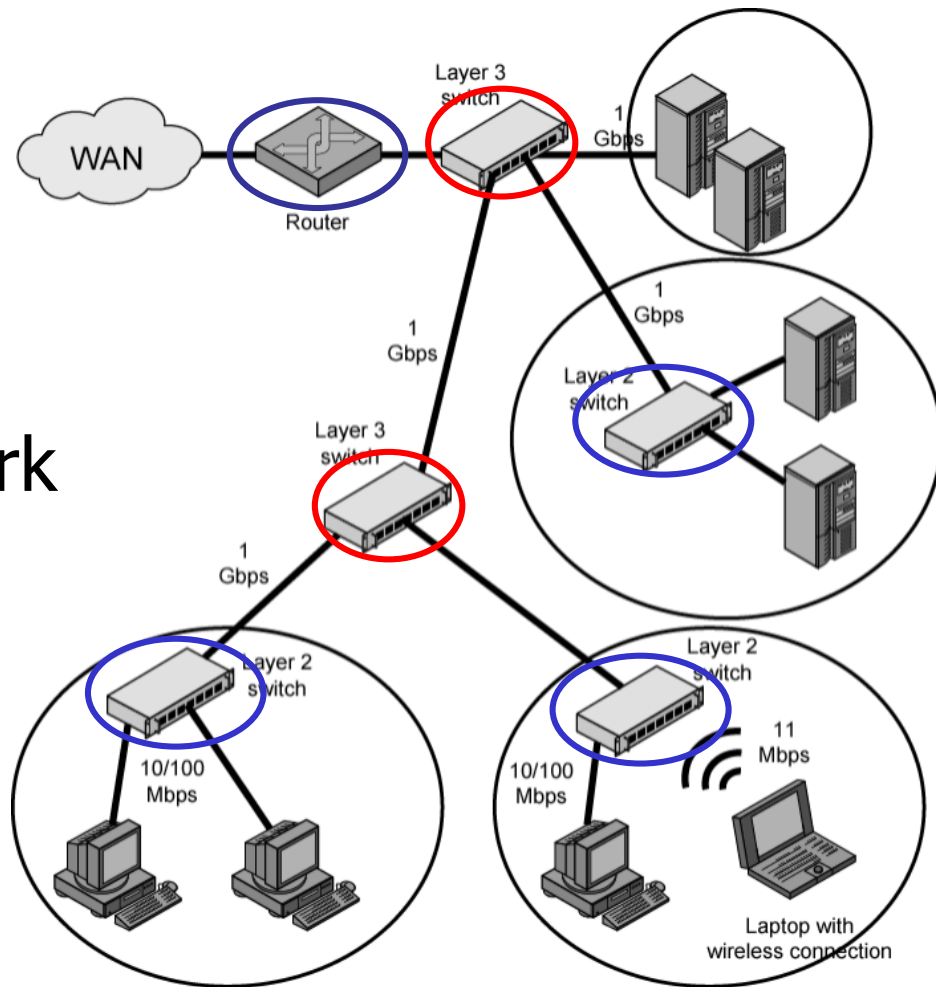
Solution:

- Break up LANs into **sub-networks** connected by **switches using router functions (layer 3 switch)**
- MAC broadcast frames limited to single sub-network
- Allow use of multiple paths between sub-networks



Typical Large LANs in an Organization

- **Circles** in diagram identify separate LAN subnetworks
- **MAC broadcast frame** limited to its subnetwork





Summary

- Bridge and Layer 2 Switch
 - 概念
 - 广播风暴问题
 - 生成树算法
 - 地址学习机制
- 比较: Bridge, hub, Layer 2 Switch, Layer 3 Switch (Router)