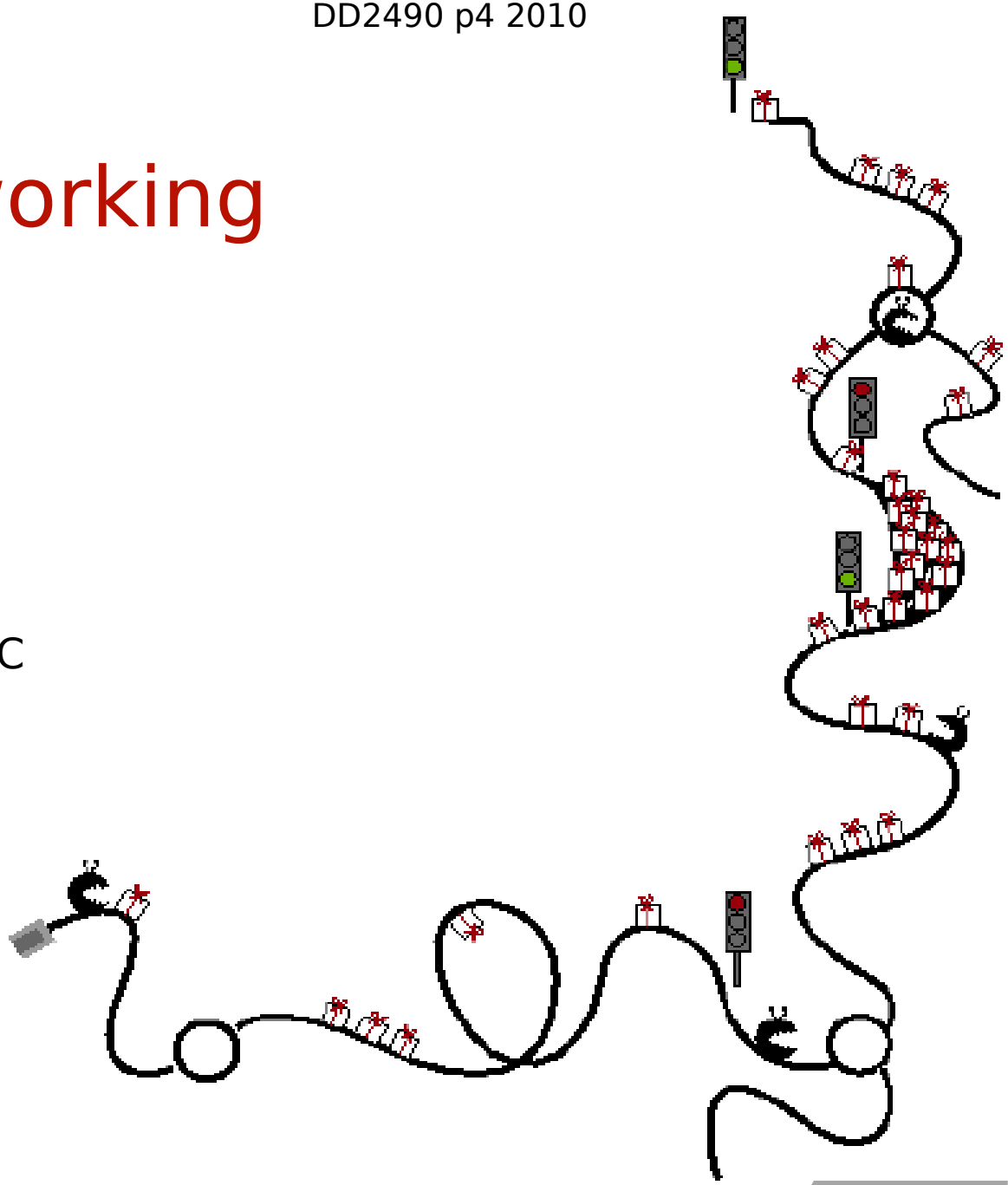
Layer 2 networking

Olof Hagsand KTH CSC

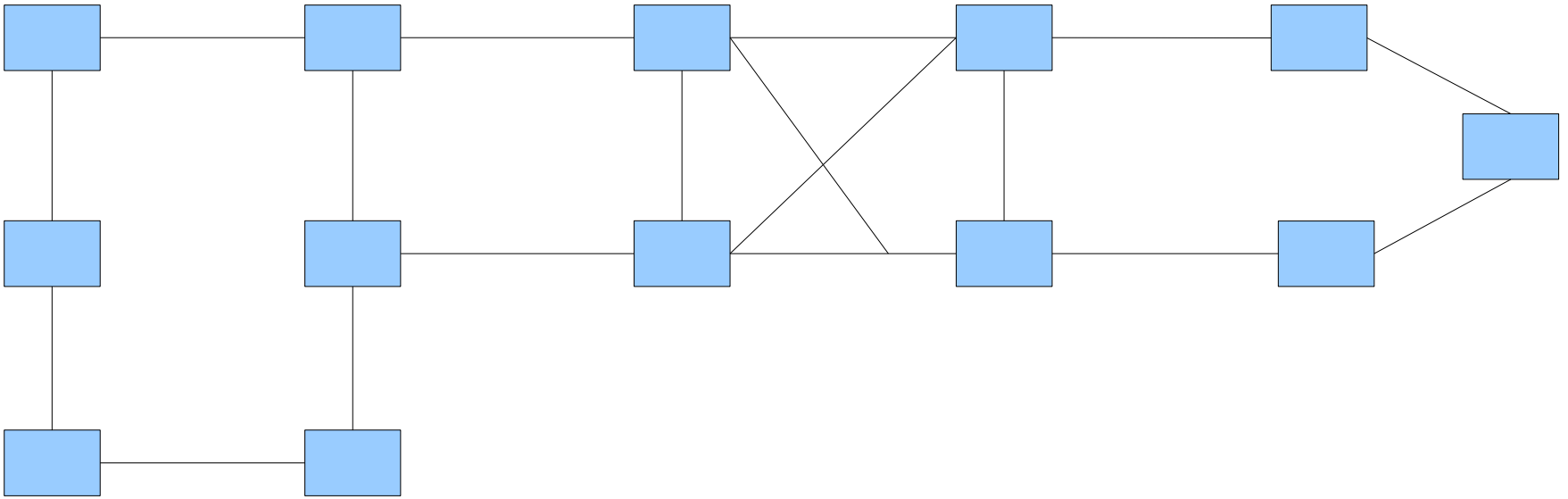


Literature

–Radia Pearlman ”Interconnections - Bridges, Routers, Switches and Internetworking Protocols”, Addison-Wesley. Section 3: Transparent bridges

(Handouts)

Building a network: routing or bridging?



- Scaling differences: traffic, state in nodes
- Cost
- Ease of configuration
- Addressing
- Management and fault detection
- Local-area / Wide-area

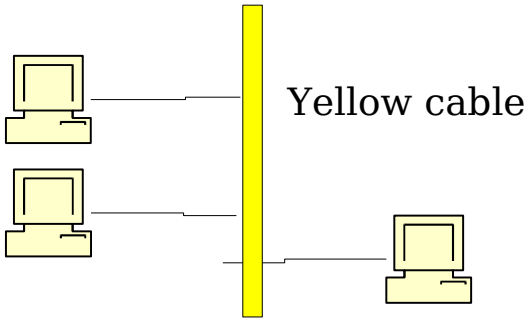
Routing vs bridging

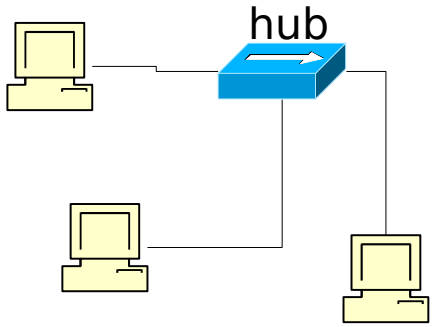
- Bridging - forwarding on layer 2
  - –A MAC address has a *flat* structure
    - many nodes -> large forwarding tables
    - broadcast reaches all nodes
  - –Simple to configure and manage, cheaper
  - –Loops detected by spanning tree protocol
- Routing – forwarding on layer 3
  - –The netid of the IP addresses can be *aggregated*
    - many nodes -> smaller forwarding tables than bridging
    - routers partition broadcast domains
  - –Routing is more difficult to configure
  - –Loops detected by routing protocols and TTL decrementation

Ethernet/IEEE 802.3 Frame (RFC894) Format

- DST: Destination Address, 48-bit MAC address
- SRC: Source Address, 48-bit MAC address
- Type: Type of data carried, e.g., IP packet
- Data: min size is 46 bytes, max size is 1500 bytes
- CRC: Cyclic Redundancy Check (or FCS – Frame Check Sum)
- 48-bit MAC address normally written 12:34:56:78:9a:bc

Ethernet / IEEE 802.3: Bus and Hubs (L1)



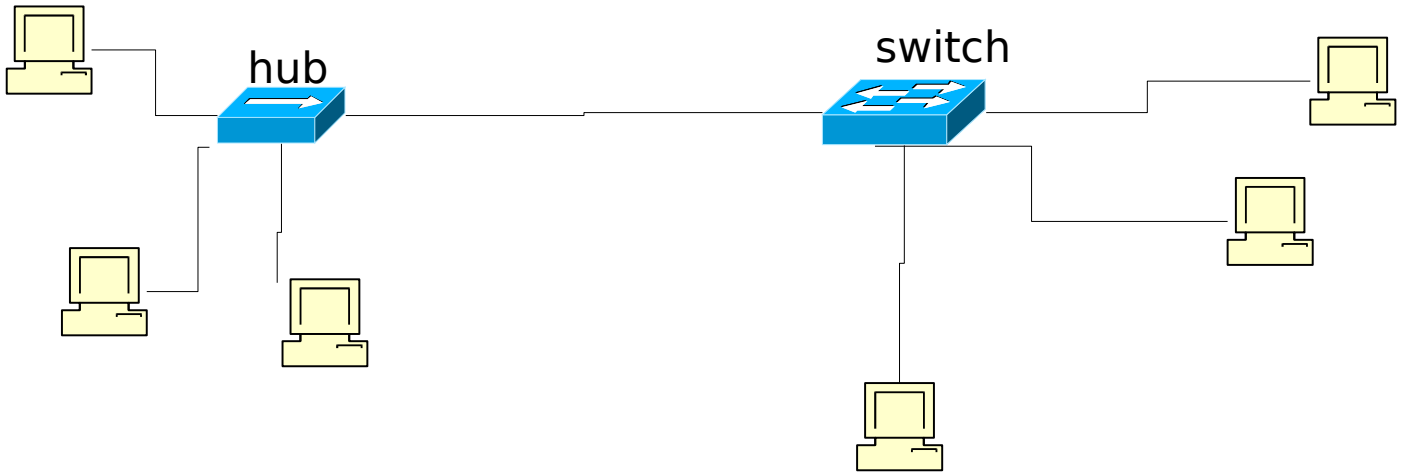


- Cables and Hubs form one *collision domain* (Layer 1)
  - –All packets *broadcast* on network using CDMA/CD

Bridging (L2)

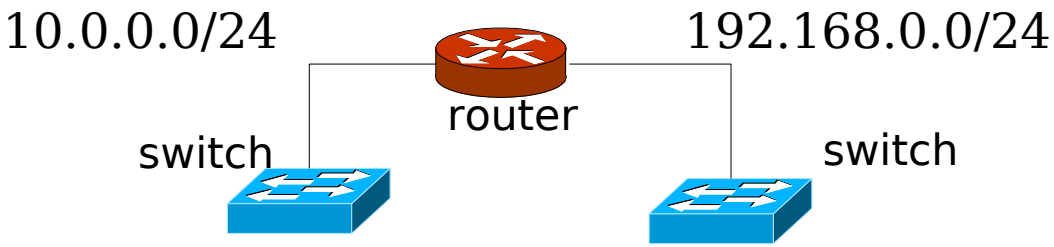
- Bridges interconnect segments forming a *broadcast domain*
- Bridges operate in the
  - link layer (according to TCP/IP)
  - physical and link layer (according to OSI)
- Bridges do not modify the structure or content of a frame
- Bridges contain logic allowing them to keep the traffic for each segment separate
- A packet entering a bridge is copied only to the segment to which the receiving address belongs
- A multiport bridge can act as a switch to connect two or more segments, e.g., as in a *switched Ethernet*

Switches / Bridges



- Switches partitions the LAN into *segments*, each segment is one collision domain
- Switches (actually *learning bridges*) learn on which port hosts are connected
  - –Tries to forward packets to where the host actually is
- The whole LAN is a *broadcast domain*
  - –Some packets are sent to all other stations (not learned, broadcast, multicast)

Routers (L3)



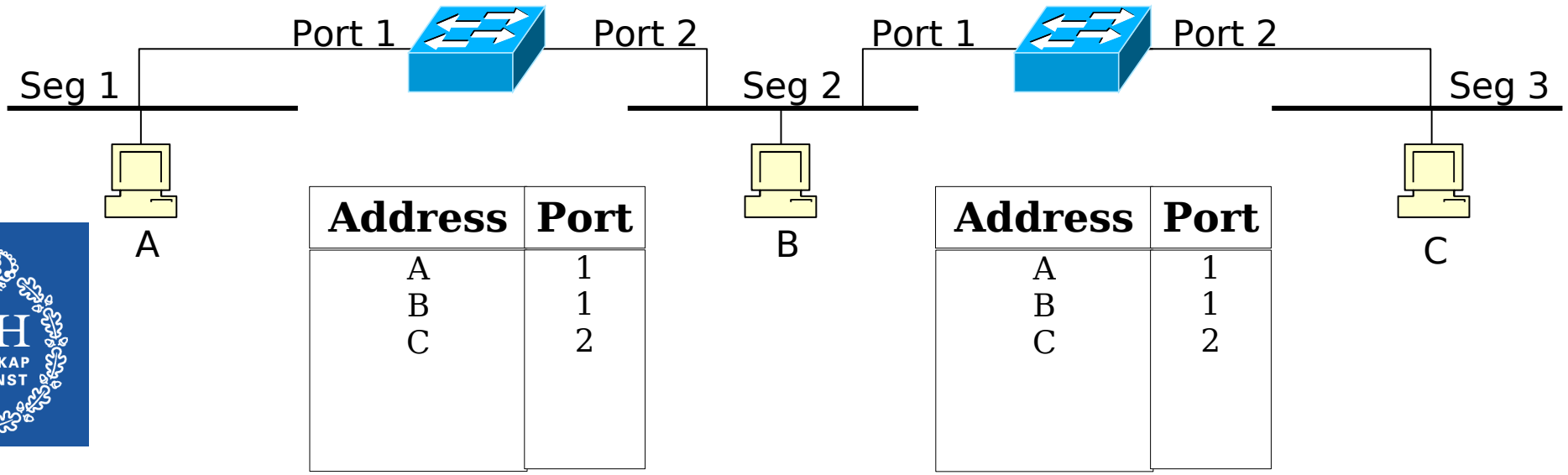
- Each LAN is still a broadcast domain with respect to IP (Layer 2)
  - It forms one IP subnetwork.
- To transmit traffic between LANs, packets pass through a router (Layer 3)
- Some see a router simply as a box that limits broadcast

The Learning Bridge

Strategy of a learning bridge:

1. Listen promiscously, receiving every packet
2. Store the source address of every packet in a *learning table* (station cache)
   - reset aging
3. For each packet, check destination address with learning table
   - If not found, *flood*: forward on all (except receiving) ports
   - If addr found, forward only on port specified in learning table.
     If specified port == receiving port, drop packet
4. Bridge ages each entry in the learning table.
   - Deletes entry if no traffic is received from that source after a certain period of time

Learning Bridges Example



The learning bridge concept works for any number of ports

And for any *tree* (loop-free) topology

Populating the learning table: Multicast and broadcast

- Multicast (`*1:**:**:**:**:**`) and Broadcast (`ff:ff:ff:ff:ff:ff`) are flooded on all ports
  - –But bridges can be extended with multicast (eg *IGMP snooping*) to keep track of where members are.
  - –But the source address is still learnt
- This means that multicast/broadcast frames typically populate the learning tables (they reach the whole broadcast domain)
- In an IPv4 sub-network, hosts send *ARP requests* on broadcast as soon as they communicate with another host,
  - –This populates all bridges with their src adress
  - –And may cause broadcast storms in a large switched network
- A second common cause is *DHCP requests*:
  - –0.0.0.0 -> 255.255.255.255 to ask for an address.
- A third is *IGMP membership reports* common in IPTV

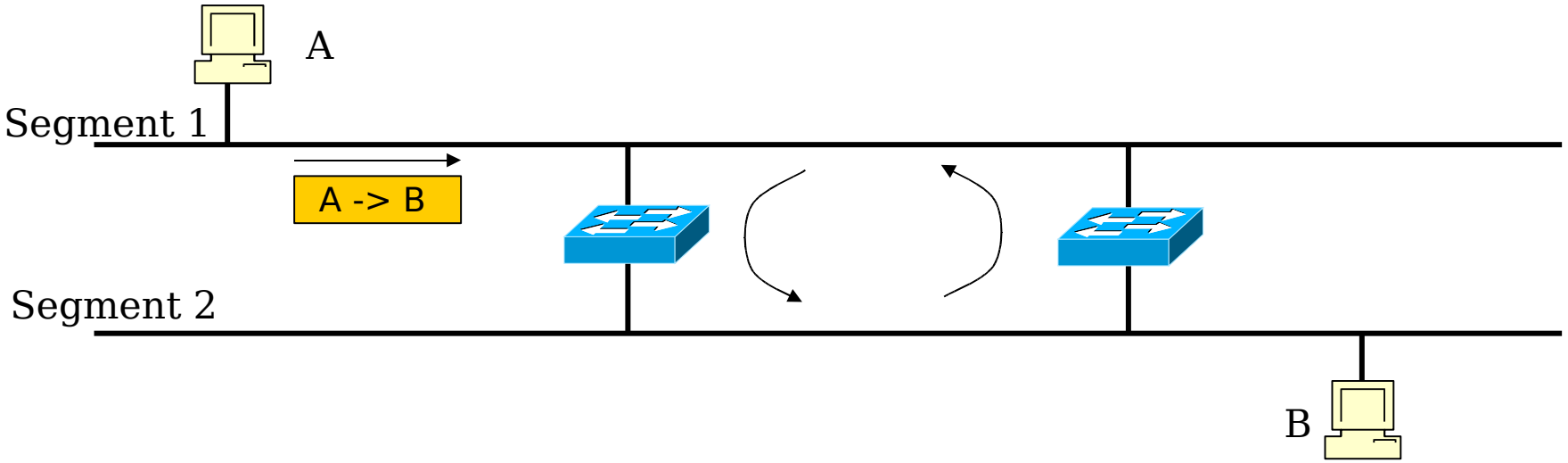
Bridging Limitations

Can a bridged network be of an arbitrary size?

No, it 'does not scale' because of:

- Addressing
  - –No hierarchy in the MAC addresses → no aggregation can be used
- Table sizes (size of the forwarding table) grows linearly with the number of sources
- Traffic congestion due to broadcast increases linearly
  - –Broadcast and multicast spans the whole broadcast domain
  - –Broadcast domains must be kept at a reasonable size

Learning Bridge – Loop Problem



- Learning bridges are unaware of hosts until they receive a packet from them.
- The bridges are not aware of the existence of other bridges.
- Example: If the bridges are unaware of B, and A sends a frame to B, the frame may start looping.
- Solution: Spanning Tree Protocol.

Spanning Tree

Purpose:

- Bridges dynamically discover a loop-free topology subset (a tree)
- The tree has just enough connectivity so that there is a path between every pair of segments where physically possible (the tree is spanning)

Basic Idea:

- All bridges exchange configuration messages, called bridge protocol data units (BPDUs), allowing them to calculate a spanning tree

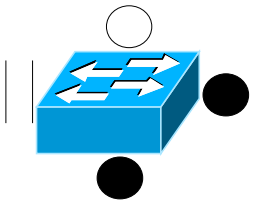
Spanning Tree Algorithm

An ID number is assigned to each bridge, and a cost to each port.

Process of finding the spanning tree:

1. The bridges choose a bridge to be the *root bridge* of the tree by finding the bridge with the smallest ID.
2. Each bridge determines its *root port,* the port that has the least *root path cost* to the root. The root path cost is the accumulated cost of the path from the port to the root.
3. One *designated bridge* is chosen for each segment
4. Select ports to be included in the spanning tree (root port plus designated ports)

Data traffic is forwarded only to and from ports selected for inclusion in the spanning tree



root port

designated port

forwarding

blocked/listening/learning

STP operation

- A bridge initially assumes itself to be the root (cost = 0)
- A bridge continously receives configuration messages on each of its ports and saves the ”best” configuration message from each port
- If a bridge receives a better configuration message on a segment than the configuration message it would send, it stops sending configuration messages
- When the algorithm stabilizes, only one bridge on each segment (Designated Bridge) sends configuration messages on that segment

Spanning tree poem

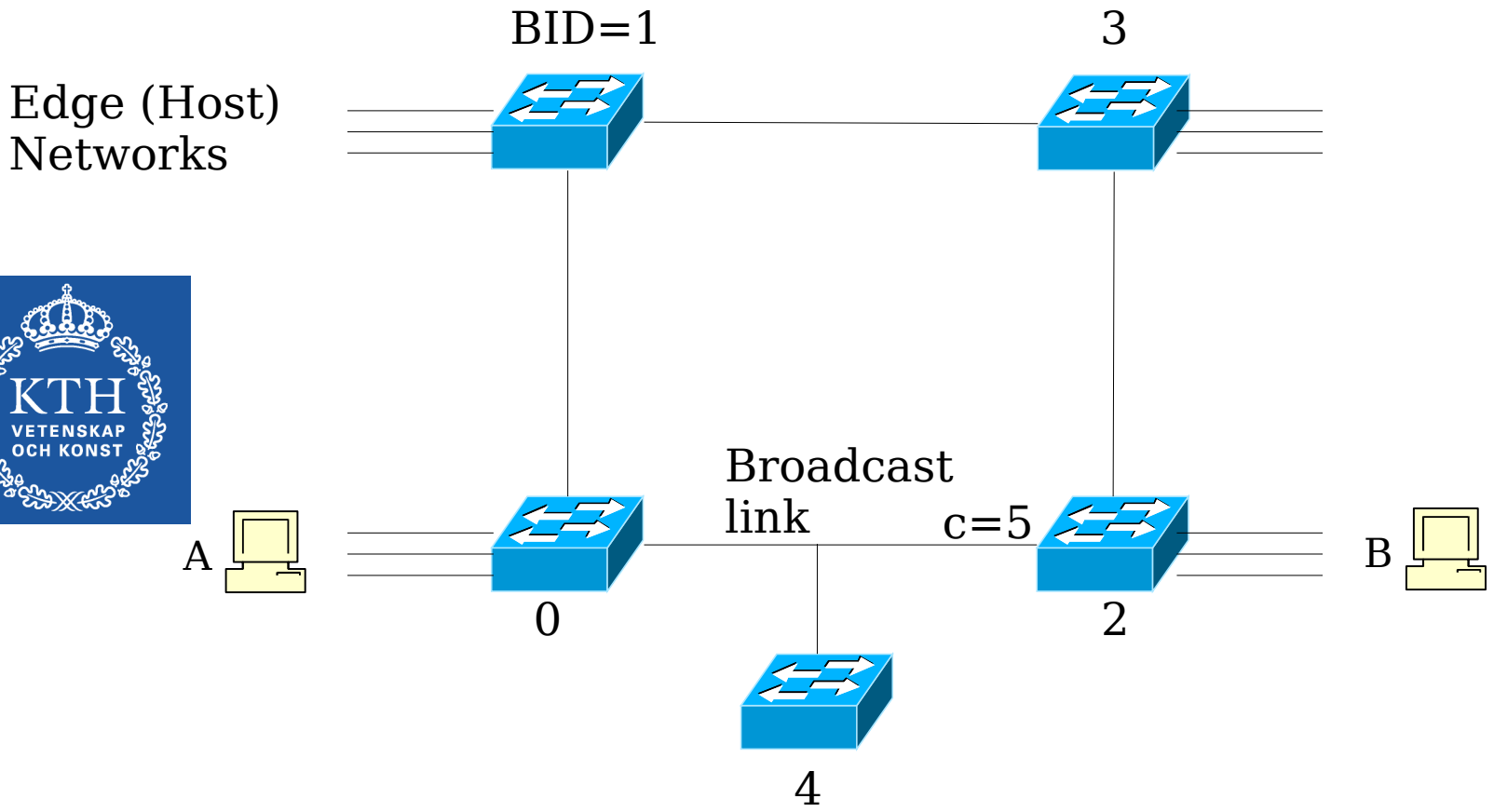
I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths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

Radia Pearlman

Priority vector

- The bridges construct a priority vector from:
  - –<Root ID, Root Path Cost, Bridge ID>
  - –These are used to construct the spanning tree
- Lower value --> Higher priority
- Example:
  - –<1,5,7> is better than <3,0,0>
  - –<4,7,2> is better than <4,7,3>
- Actually, both sending and receiving ports are also a part:
  - –<Root ID, Root Path Cost, Bridge ID, snd port, rcv port>

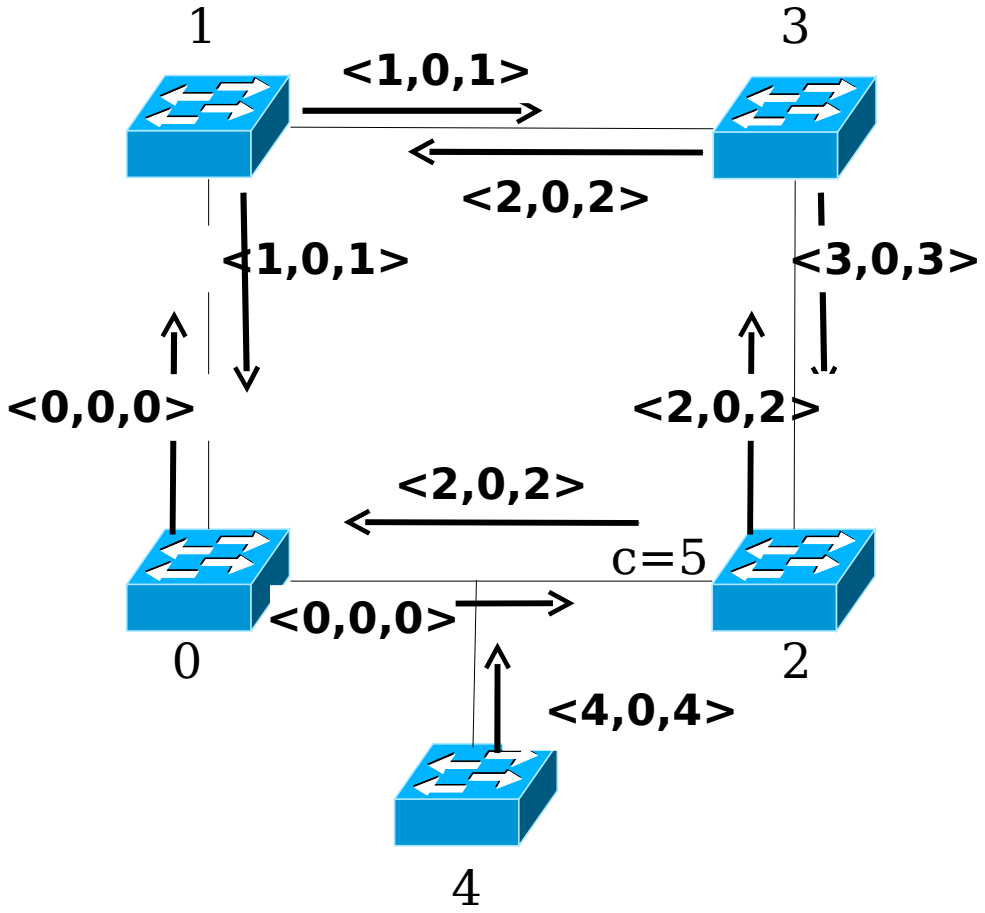
Simple bridged network



BID – Bridge Id: <prio>.<macaddress>
Example: `8000.0:14:6c:e0:65:39`

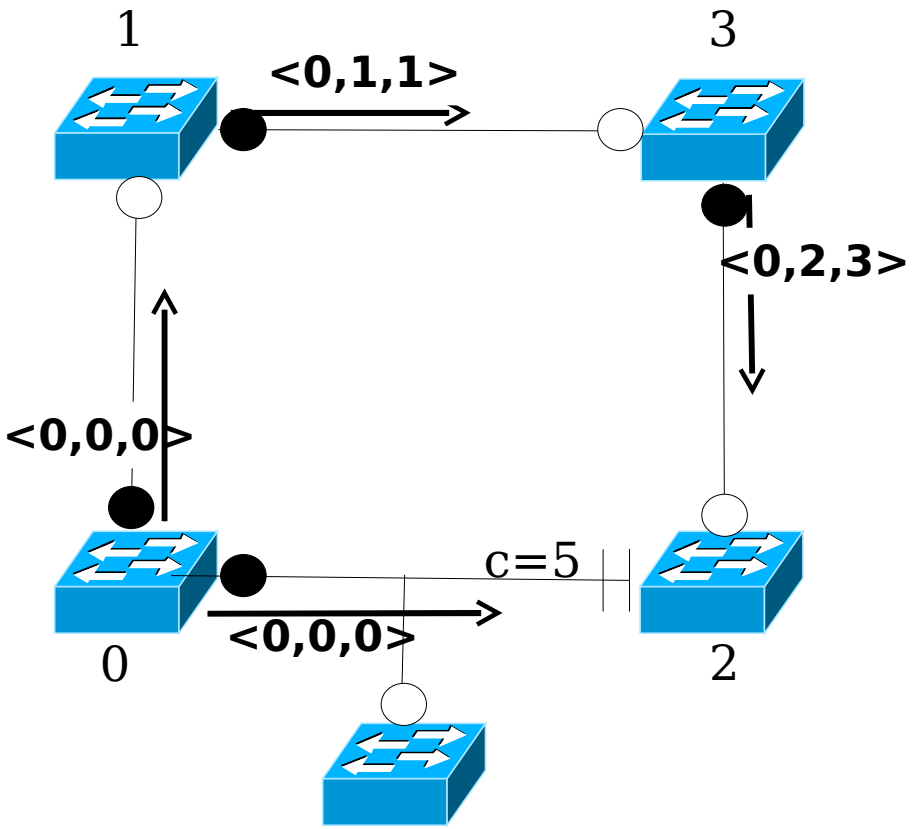
`Note: bridge 2 has assigned cost 5 on its left link`

Initial State:
All bridges assume they are root

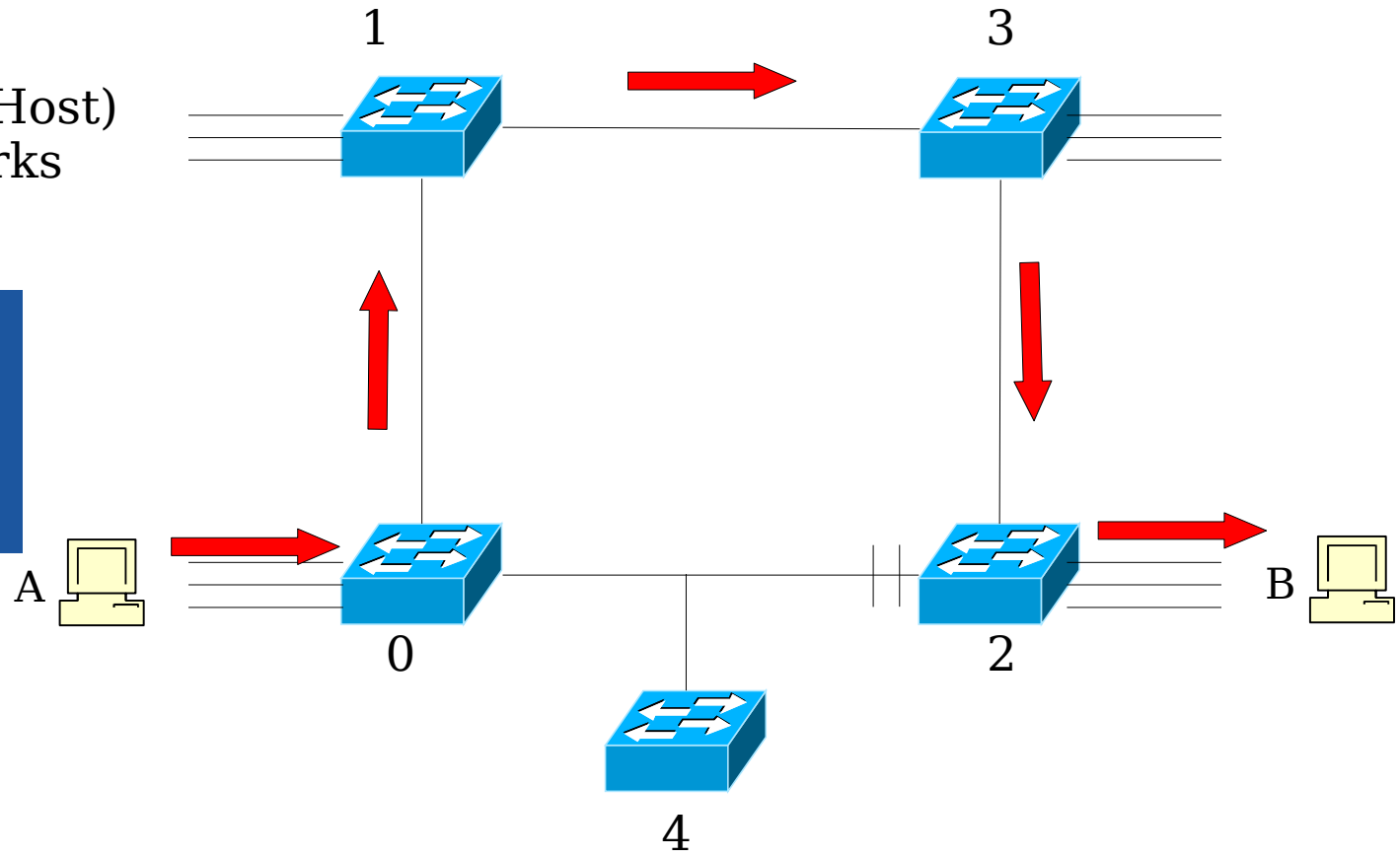


All ports are blocked

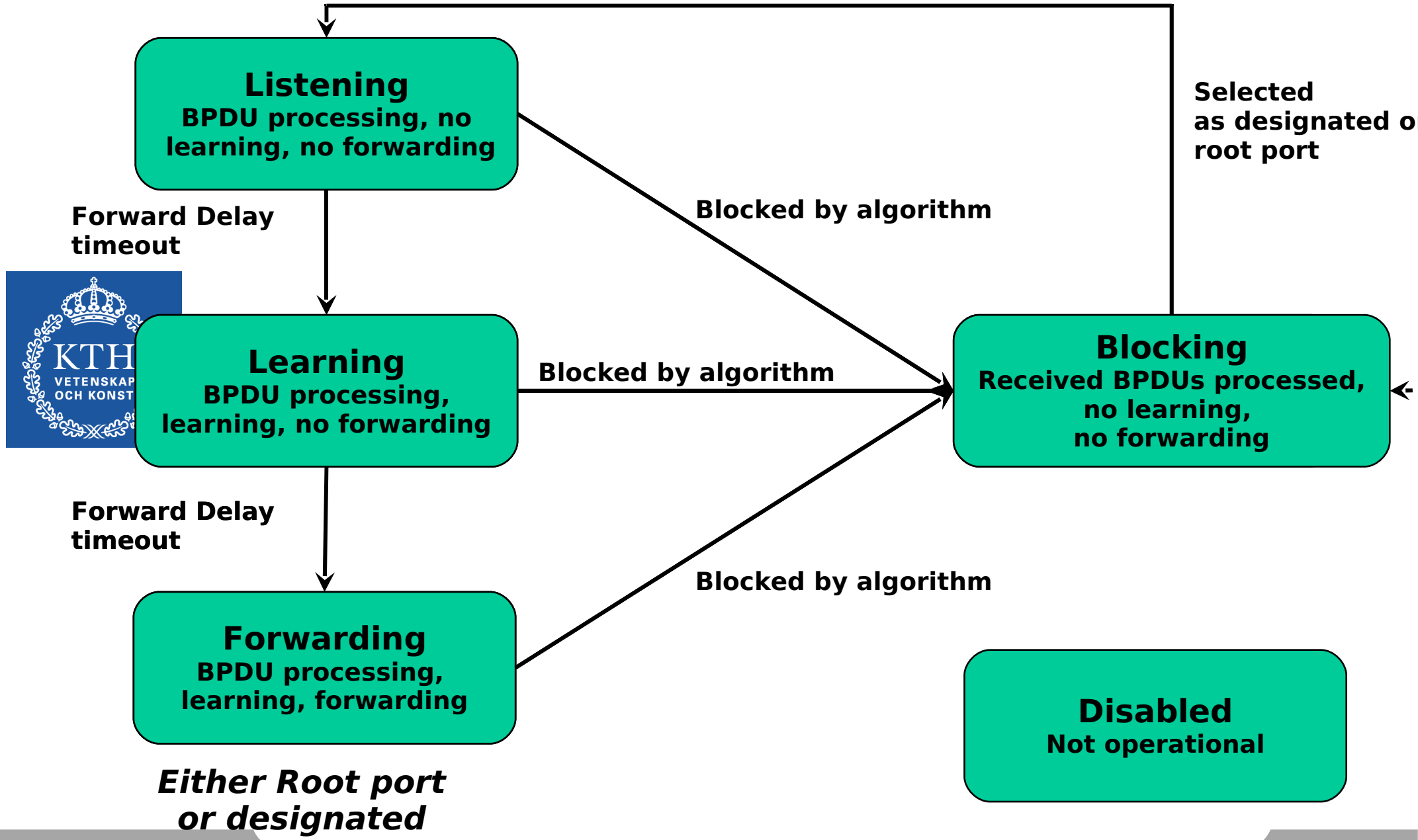
Final state



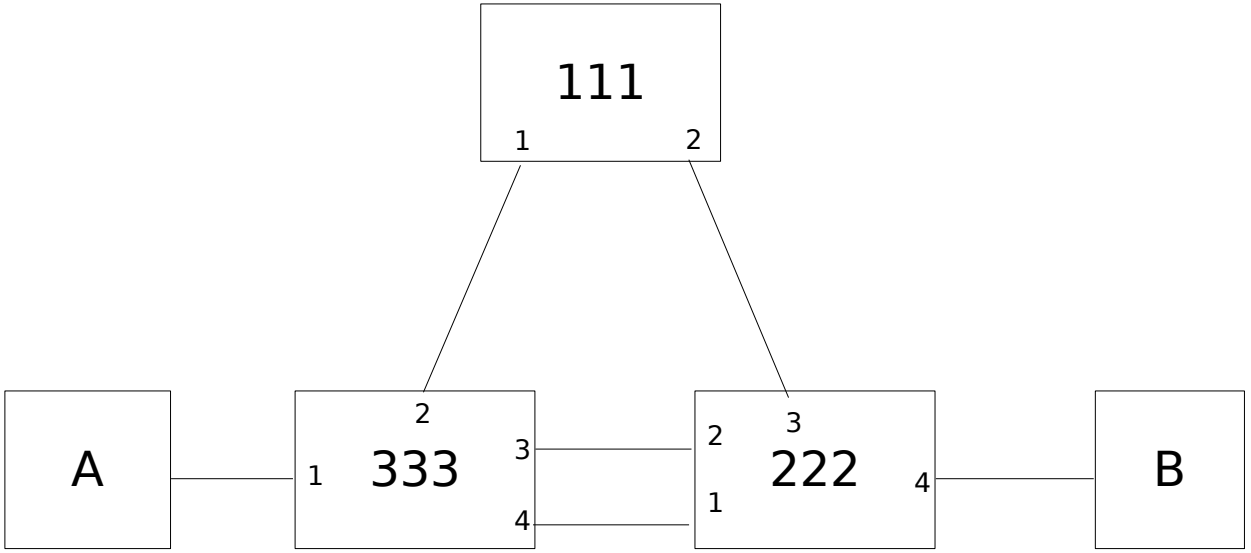
Example: traffic path from A to B



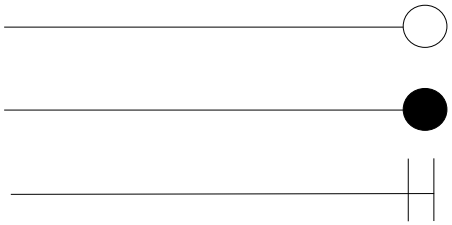
Port States


Listening
BPDU processing, no learning, no forwarding
Forward Delay timeout
Learning
BPDU processing, learning, no forwarding
Forward Delay timeout
Forwarding
BPDU processing, learning, forwarding
Either Root port or designated
Blocked by algorithm
Blocked by algorithm
Blocked by algorithm
Blocking
Received BPDUs processed, no learning, no forwarding
Selected as designated o root port
Disabled
Not operational

Exercise



- Traffic sent between A and B
  1. Learning table after stabilization?
  2. What happens when link between 111 and 222 fails ?



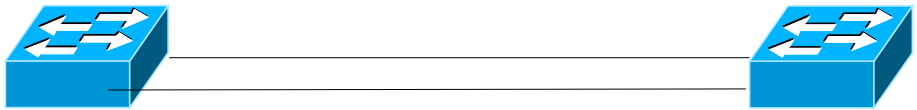
root port

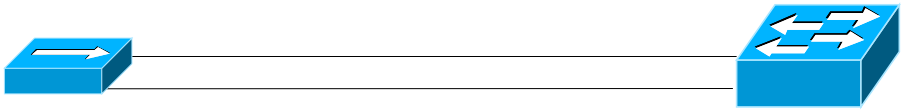
designated

blocked/listening/forwarding

Why are ports a part of the priority vector?

<Root ID, Root Path Cost, Bridge ID, *snd port, rcv port*>

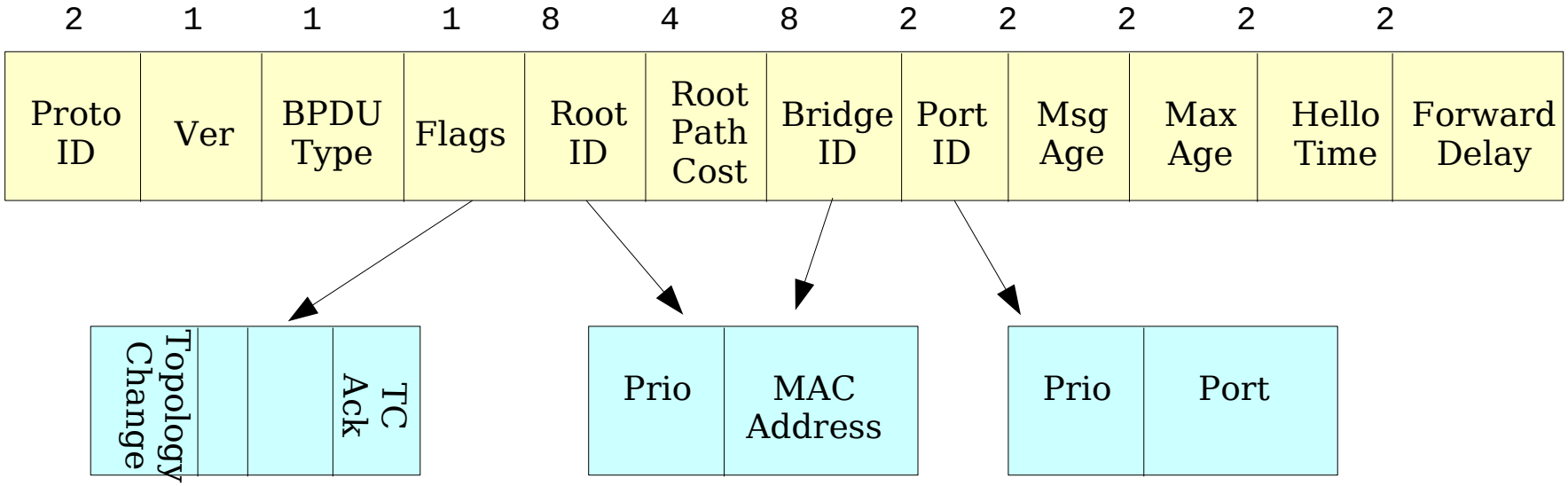




hub

- Many links connecting two switches
  - –One link is chosen
  - –Note that this leads to: no load-balancing
- Many ports to same LAN

BPDU format



- Bridge/Root priority was originally 2 bytes, but now (from 2004) only 4 bits.
- Port priority was originally 1 byte, but is now (from 2004) 4 bits.

```
# tcpdump -ni iwi0
15:20:50.964978 802.1d config root=8000.0:14:6c:e0:65:39
rootcost=0x0 bridge=8000.0:14:6c:e0:65:39 port=0x8002 age=0/0
max=20/0 hello=2/0 fwdelay=0/0
```

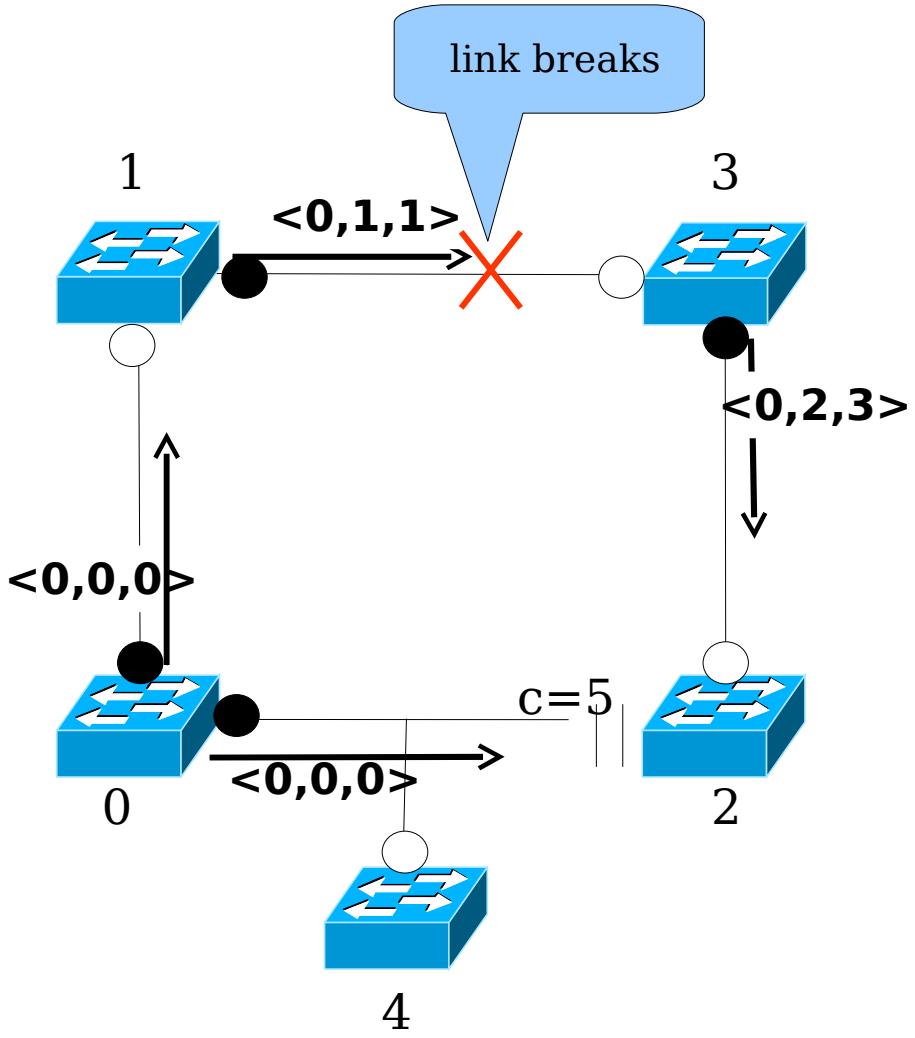
BPDU Fields

- Proto ID: 0
- Version: 0
- BPDU type: 0 (Configuration message)
  128 (Topology Change Notification)
- Flags:
  - TC: Topology Change flag
  - TCA: Topology change notification acknowledge
- RootID: Macaddr+prio of bridge that is root
- Root Path Cost: Total cost to root
- Bridge ID: Macaddr+prio of bridge that sent message
- Port ID: Prio + port nr of port where message was sent
- Message age: time since root sent message (in 1/256th second)
- Max age: When message should be deleted (20s)
- Hello time: Time between generation of config msgs by root bridge (2s)
- Forward delay: (1) Time to stay in intermediate states before port is forwarding; (2) entry timeout when topology change. (15s)

Failure

- A topology change occurs due to:
  - –Failed link
  - –Failed switch
  - –New link cost
- Message aging times out (Maxage ~20s). Possible actions:
  - –Bridge selects a new root (or itself)
  - –Becomes designated bridge of a segment
  - –Blocks a port
- Ports changing state enter Listening state. Takes 2* Forwarding delay until port starts forwarding (~30s)
- Notifies root of change: Topology Change Notification
  - –Continue until Configuration Message with TCA is received
- Root sends Configuration Message with TC set
  - –This will cause all aging entries to use short cache timeout (forwarding delay) instead of long (eg 5min -> 15s)

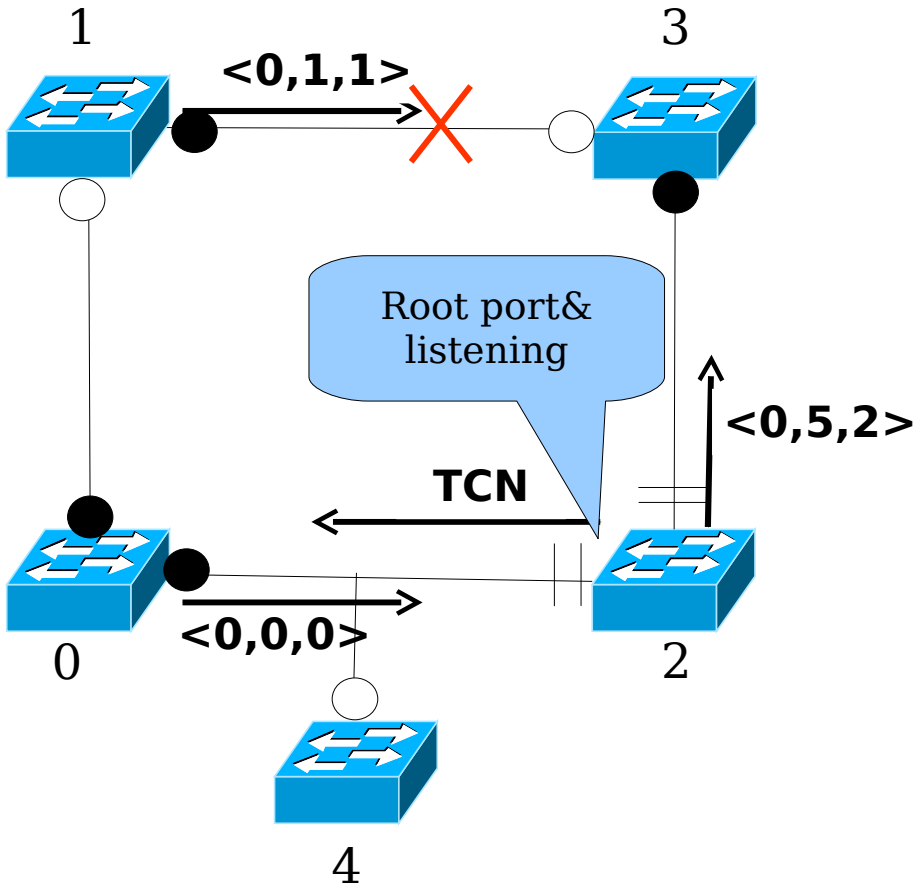
Example: A change occurs



<RootID, Root Path Cost, Bridge ID>

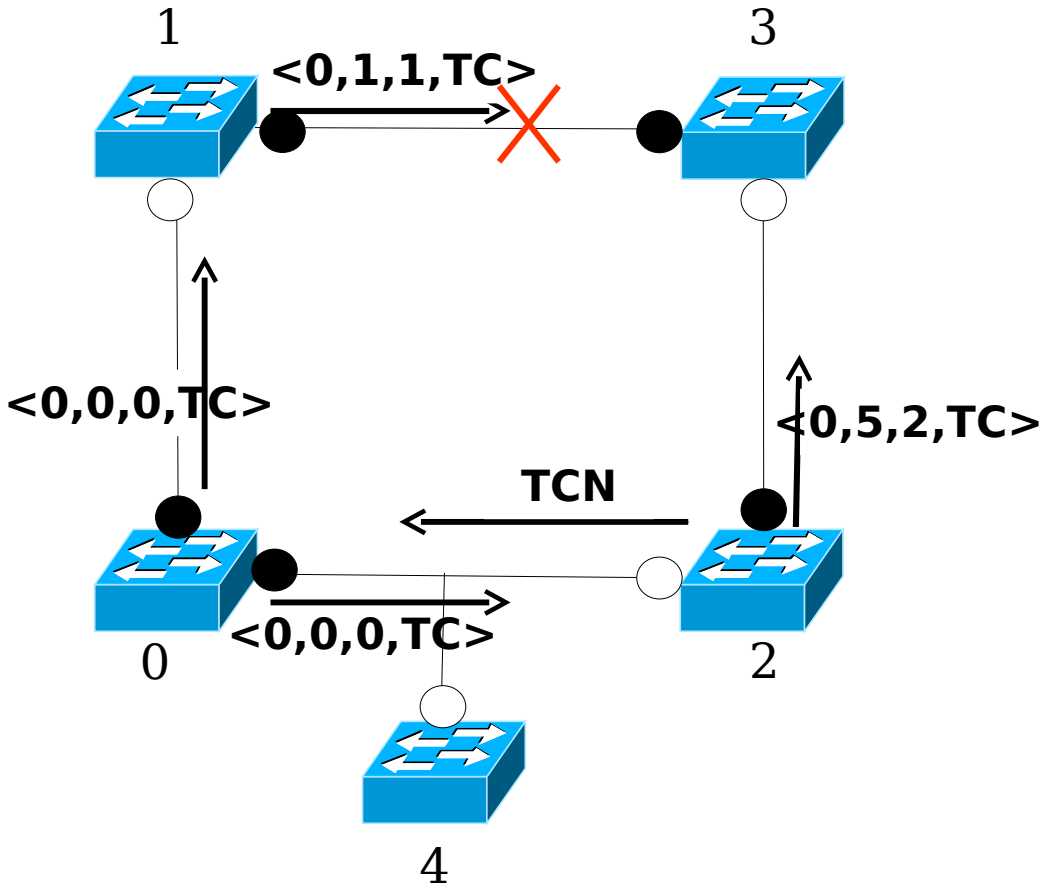
Bridge 2 times out

- Receives no config messages
- Timeout is MaxAge (~20s)
- Bridge 2 elects new root port and sends Topology notification upstreams and sets port in listening state



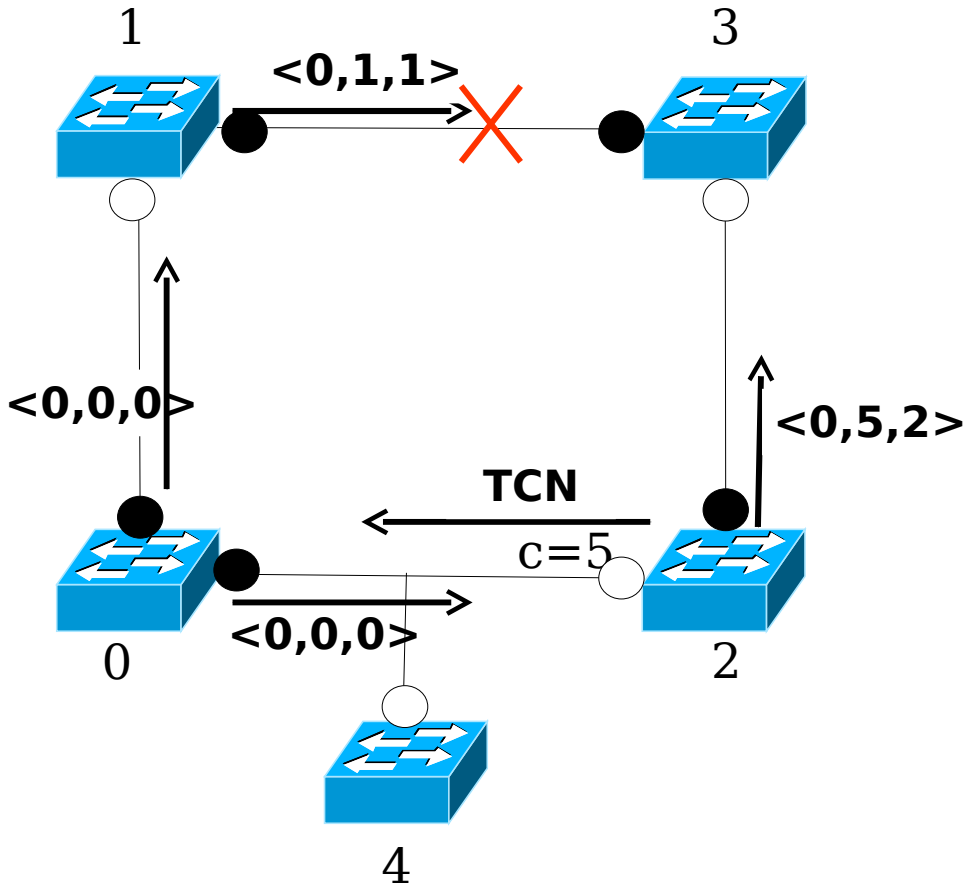
Root notifies that topology has changed

- Root sends TC (Topology Changed) flag
- Causes all bridges to use shorter cache timeout
  - –Forward Delay: ~15s

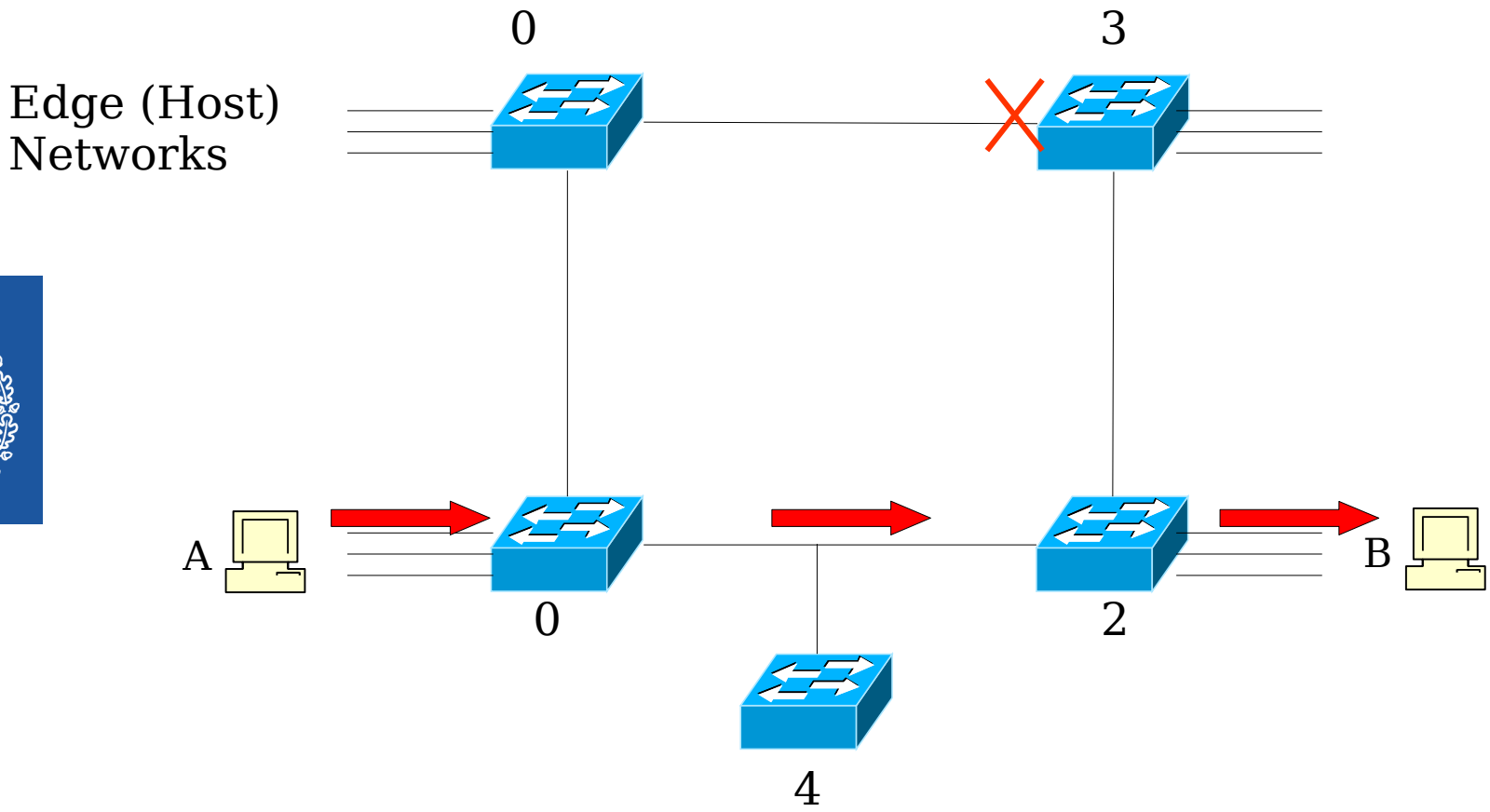


Bridge 2 ports in forwarding state

- Bridge 2's port is in forwarding state (after ~15s + 15s)
- *Total time for re-configuration: ~50s*

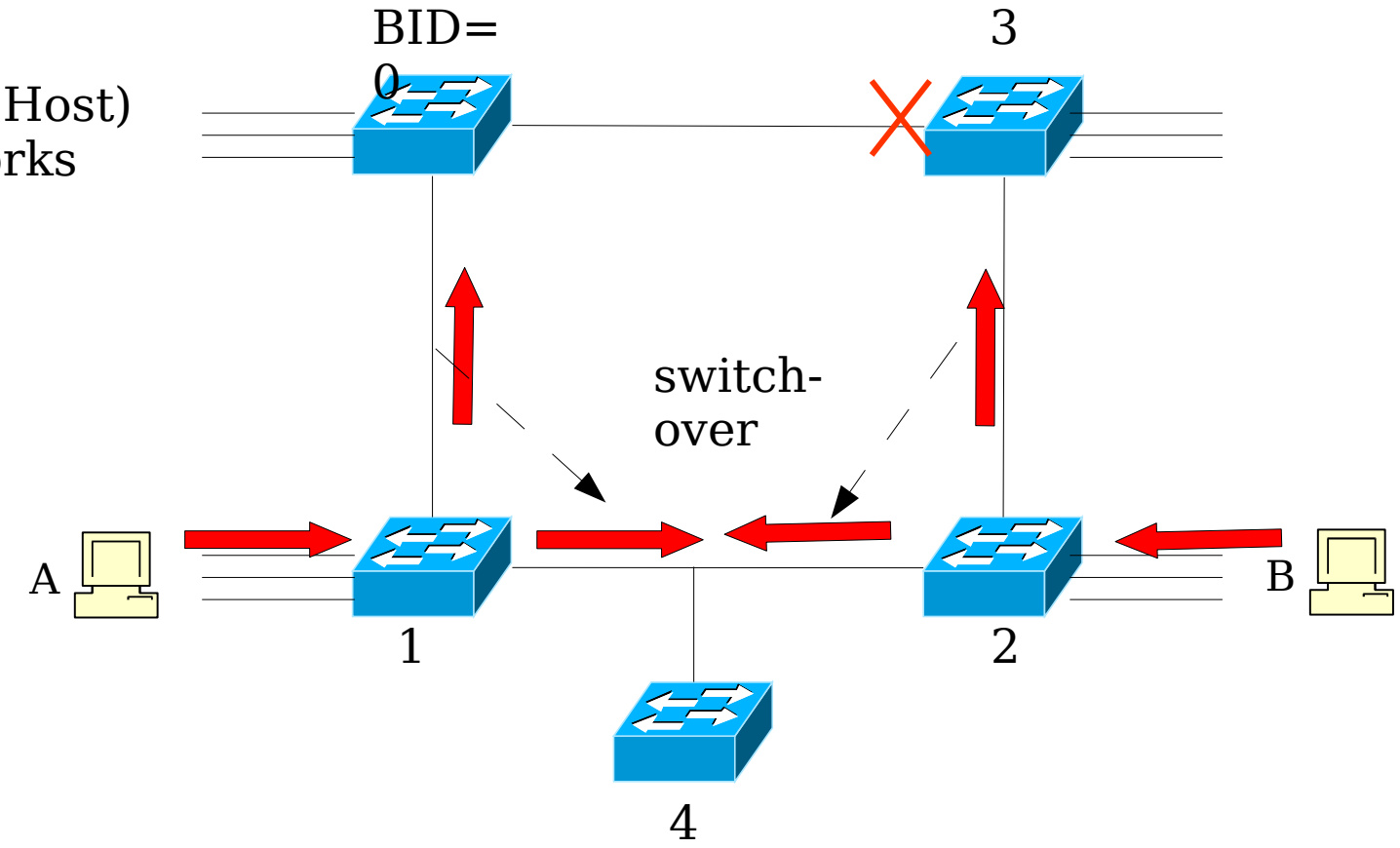


Example: traffic path from A to B



Why TCN?

If no TCN -> bridges could keep their old cache timeout (~5min) and old forwarding paths could remain.
TCN ensures that old state is flushed.
But if network is large, TCNs will occur all the time and cause frequent re-learning (and flooding)



STP Parameters

Parameter	Default	Range	Description
Hello time	2s	1-10s	Time between (root) config msgs
Max Age	20s	6-40s	Max age of BPDU messages
Forward Delay	15s	4-30s	(1) State transition timeout (2) entry timeout when topology changes (Short cache timer)
Bridge prio		0 – 7	Priority for selecting root
Port prio		0 – 7	Port priority
Long cache timer	5min		Timeout of table entry
Path cost	200K (100Mbps)	1-200M	Cost added to root path

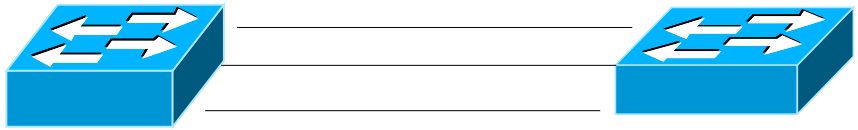
Rapid Spanning Tree

- Rapid Spanning Tree Protocol (RSTP)
- RSTP converges around a second
  - –First defined in IEEE 802.1w
  - –Replaces STP after 2004 in 802.1D
  - –Full-duplex mode - No shared links
  - –Backwards compatible with STP
- RSTP has two more port designations
  - –Alternate port – backup for root port
  - –Backup port – backup for Designated Port on the segment
- In RSTP all bridges send BPDUs automatically
  - –In STP, root triggers BPDUs (hello timer)
- In RSTP, bridges act to bring network to convergence
  - –In STP, bridges passively wait for time-outs before changing port state

Multiple Spanning Tree Protocol (MSTP)

- Separate trees for separate VLANs
- Part of IEEE 802.1Q
- Network organized into *regions*
- Regions have their own spanning tree topologies
  - –VLANs are associated with each topology
- One common spanning tree for the entire network
- MSTP based on RSTP

Link aggregate group (LAG)

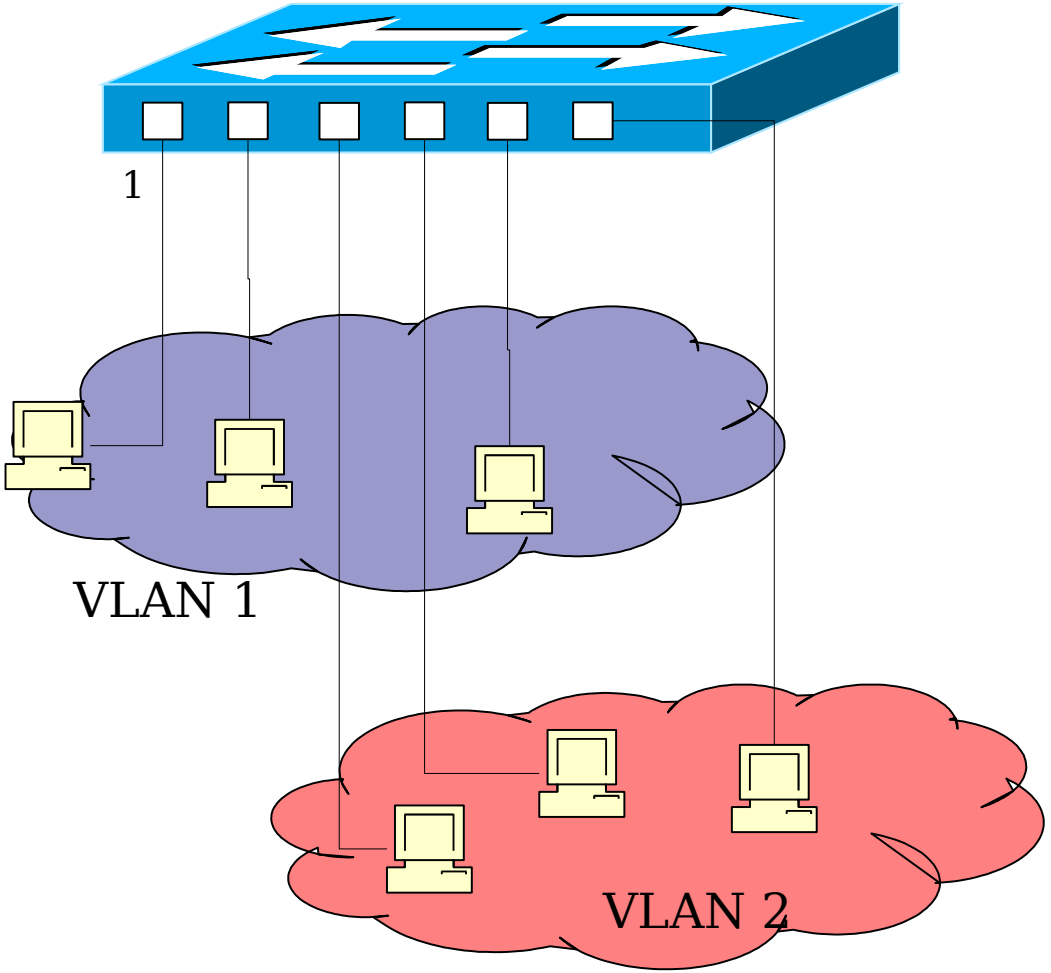


- Ethernet trunking / Link Groups
  - IEEE 802.1AX / Old: IEEE 802.3ad
- Suppose your 1Gbps link is overloaded
- But you cannot afford a 10Gbps link (10G cards may be expensive)
- Then you can group many links into a LAG (or Link Group)
- They are physically (L1) different ports
- But logically (L2), the same link.
- Packets forwarding use hashing to send packets on different links
  - So that packets from same flow is not re-ordered
- But this means that some links can be overloaded, and others not!

IEEE 802.1Q: Virtual LANs (VLANs)

- Need a way to divide the LAN into different parts
  - –Without physical reconfiguration
- Moving stations without reconfigurations
- Create virtual workgroups
- Keep broadcasts isolated
- Keep different protocols from each other
- Requires a router to communicate between VLANs

VLANs: Example



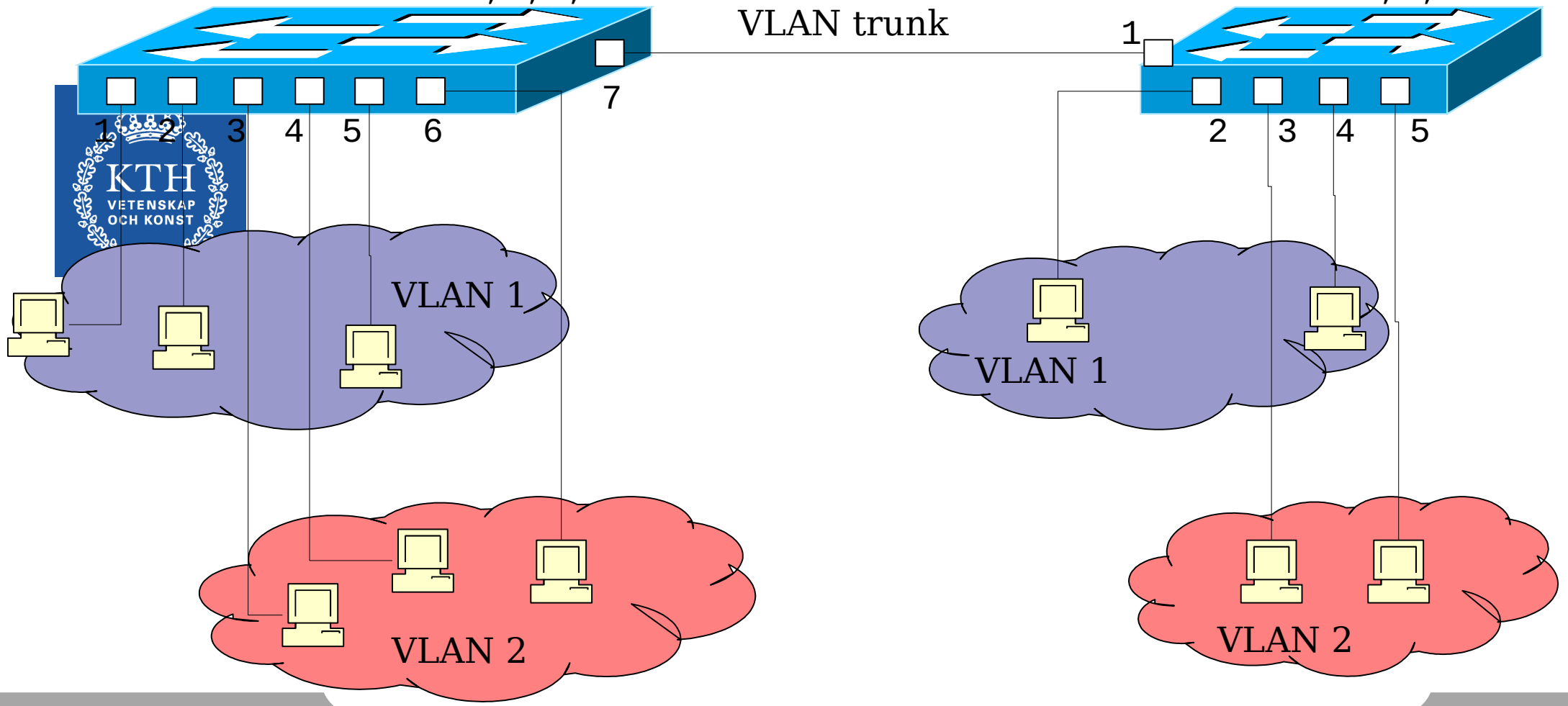
VLAN grouping

- How is VLAN membership determined?
  - –Port numbers (portvid) for incoming traffic
    - Ports 1, 2, 5: VLAN 1
    - Ports 3, 4, 6: VLAN 2
  - –VLAN membership for outgoing traffic
- VLAN membership can be done per MAC address (uncommon)
- Frame tagging - explicit VLANs
  - –VLAN *trunking*
  - –Multiplex many VLANs over the *same* link
  - –Frames belonging to different VLANs can be sent on the same trunk by VLAN encapsulation using tagging.
  - –Trunks can interconnect any number of bridges in any topology

VLAN Tags and Trunks

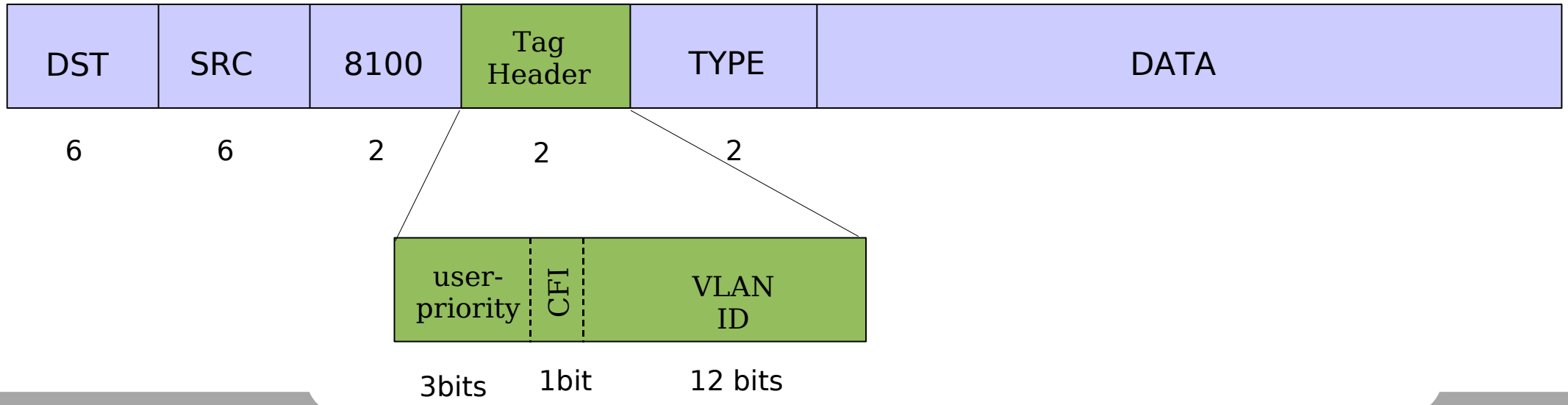
```
pvid 1: 1,2,5
pvid 2: 3,4,6
tagged: 7
VLAN 1 members: 1,2,5,7
VLAN 2 members: 3,4,6,7
```

```
pvid 1: 2,4
pvid 2: 3,5
tagged: 1
VLAN 1 members: 1,2,4
VLAN 2 members: 1,3,5
```



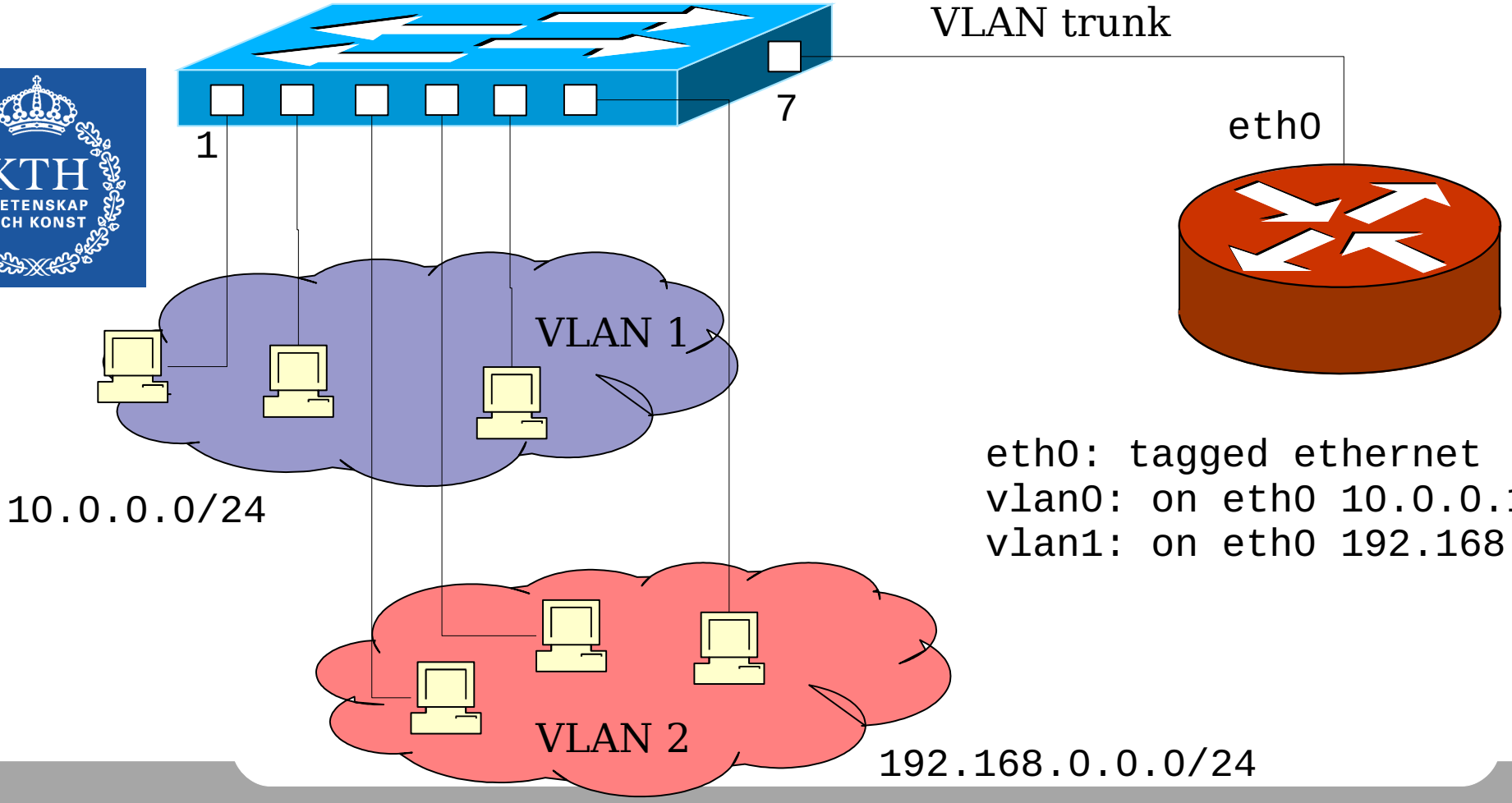
VLAN Encapsulation

- A VLAN *tag or VID* (12 bits) has been added to the Ethernet frame in order to distinguish between the VLANs
  - –4096 VLANs
- IEEE 802.1Q
- Special payload type: 0x8100
- User priority field (3 bits)



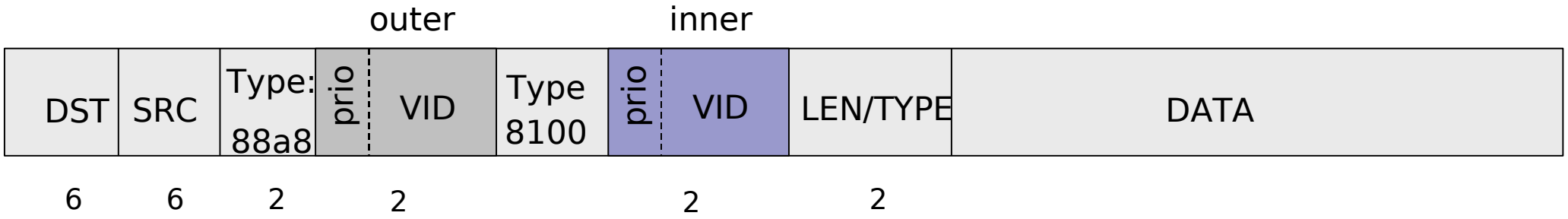
Routing between VLANs

- Router sees every VLAN as a separate interface
- Every VLAN is one IP subnet
- Many switches have this routing capability internally:
  - –router-switch

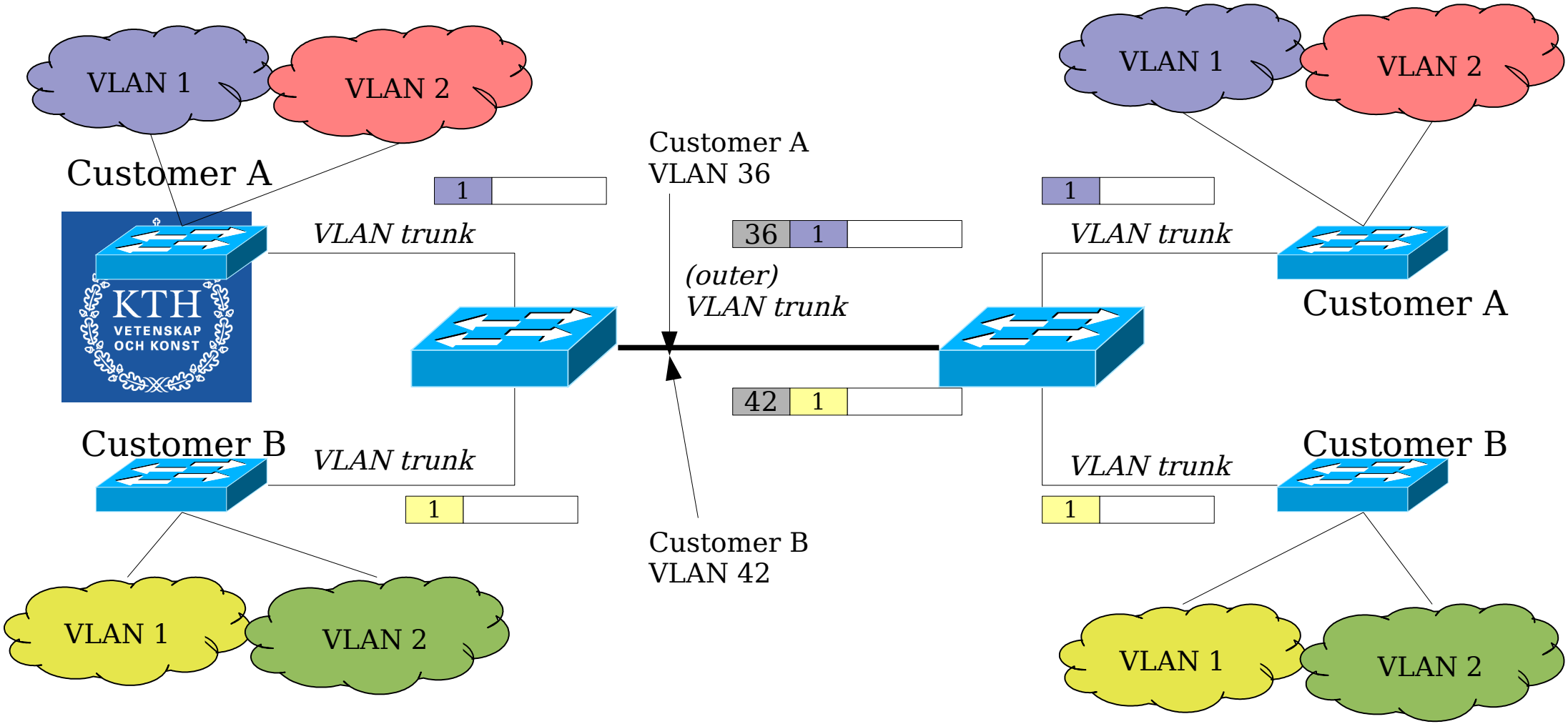


IEEE 802.1ad: VLAN stacking: QinQ

- Only 4096 VLANs.
- You may want to have several VLAN namespaces multiplexed on one link.
  - –Customers have their own VLAN space within provider VLANs
  - –Provider Bridging
- VLAN stacking puts a VLAN trunk header inside another one
  - –(similar to MPLS label stacking)
- But you still do learning using the same MAC address (SA)



VLAN stacking - Example



IEEE 802.1ah: Provider backbone bridges

- QinQ still uses the original MAC address for learning and spanning tree.
- PBB introduces tunneling of MAC frames in MAC frames, not just another tag
- One set of MAC addresses for the backbone. End-user mac addresses not used in backbone
  - Smaller backbone learning tables
  - Separate address domains (customer and back-bone)

IEEE 802.1ag: Operations and management

- Many of the new Ethernet standards are engineered for Ethernet as a MAN or WAN technology: Metro Ethernet
- But Backbone technologies have much higher requirements on management than have LANs
- O&M includes fault management:
  - loopback: 'Ethernet ping',
  - link-trace: 'Ethernet traceroute'
  - continuity check
  - delay measurement