Repeaters and bridges

Fulvio Risso

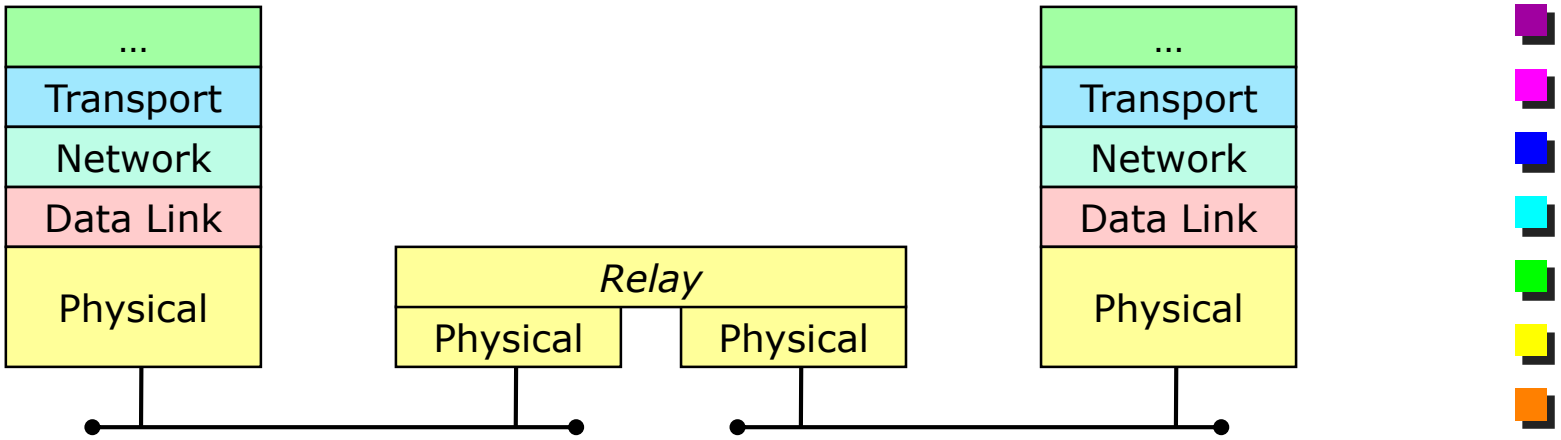
Politecnico di Torino

LAN devices in brief

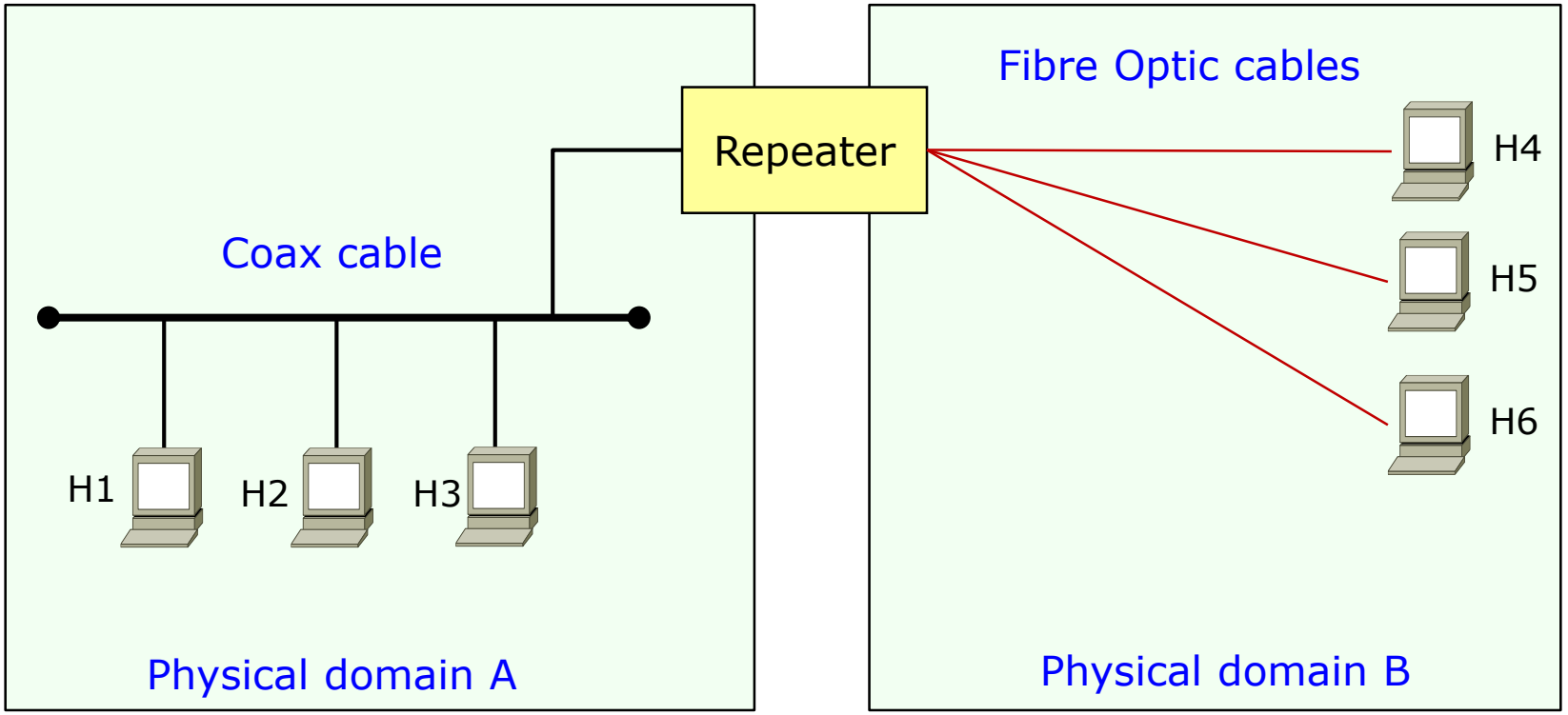
- L1: Repeater
  - Hub
  - Separate physical domains, same collision domain
- L2: Bridge
  - Switch
  - Separate collision domains, same broadcast domain
- L3: Router
  - L3 switch
  - Separate broadcast domains
  - Not really specific for LANs
  - Not covered in the current slides

Repeater

- Interconnection at the physical layer
  - Receives and propagates a sequence of bits
- Used for
  - Interconnecting networks having the same MAC
    - I.e., all ports must have the same speed
    - E.g., Ethernet 10Mbps fiber to copper
  - Recovering signal degradation (long cables), allowing larger distances

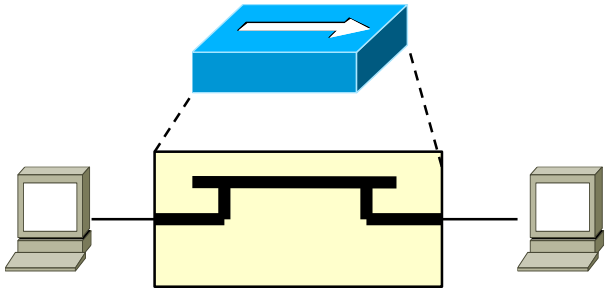


Repeater: example



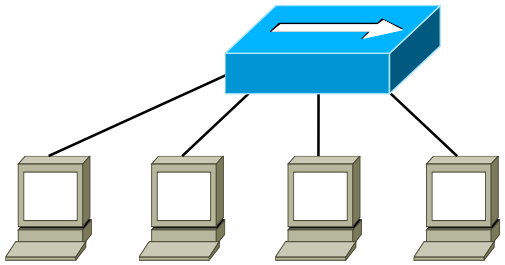
Repeater: characteristics

- Functions
  - Signal Amplification
  - Signal Symmetry
  - Signal Retiming
  - Carrier Sense and Data Repeat
  - Collision Detection and Jam Generation
  - Test functions
- Active device
  - Not just a “passive” backplane (i.e., signal amplification)
- No longer used (at least on Ethernet LANs)



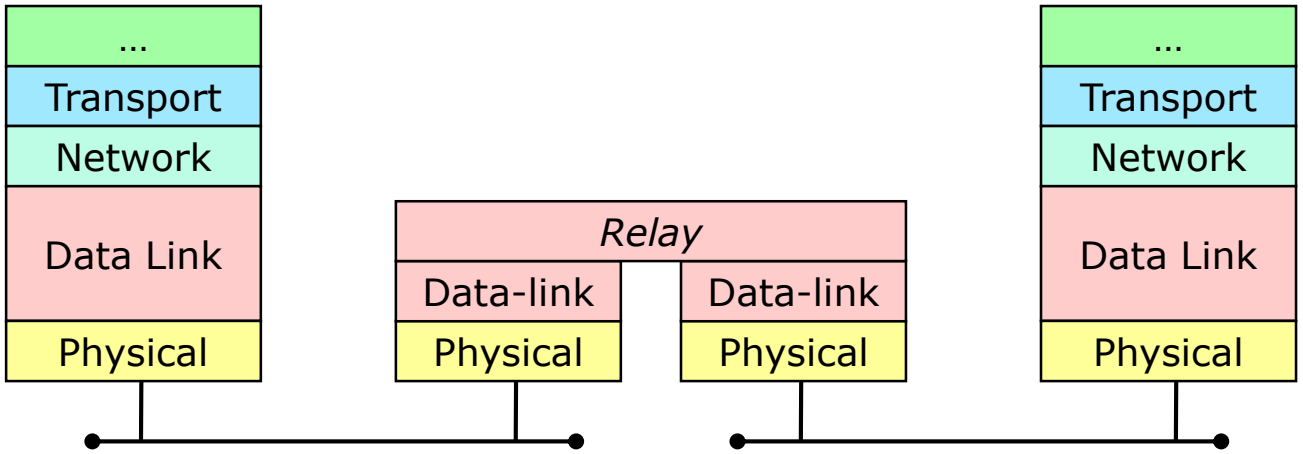
Multiport repeaters: Hubs

- Hubs are multiport repeater
  - Repeater with more than 2 ports
- Required for twisted pairs and fiber cabling (hub-and-spoke topology)
  - It became common with the adoption of structured cabling
  - More flexible (and robust) than the old coax cable
- On Ethernet, it allows reaching (almost) the theoretical collision domain
  - Overcomes limitations of physical cables (e.g., 100m on 10BaseT)
- No longer used



Bridge

- Introduced by DEC in 1983 (LANBridge 100)
  - Pure software
  - 2 ports (mainly for economic reasons)
- Interconnection at the data-link layer
  - E.g. Ethernet to WiFi, Ethernet to Fast Ethernet
  - Different MACs (medium access mechanism, framing)

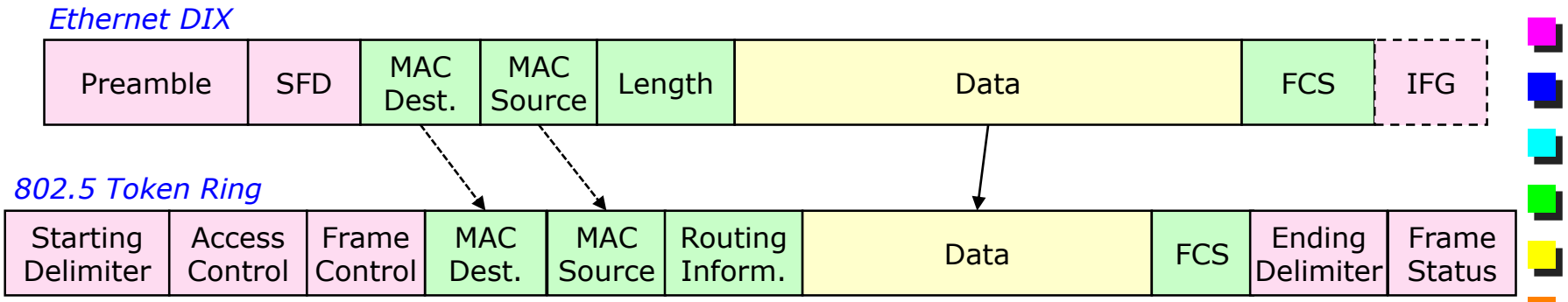


Bridge: objectives

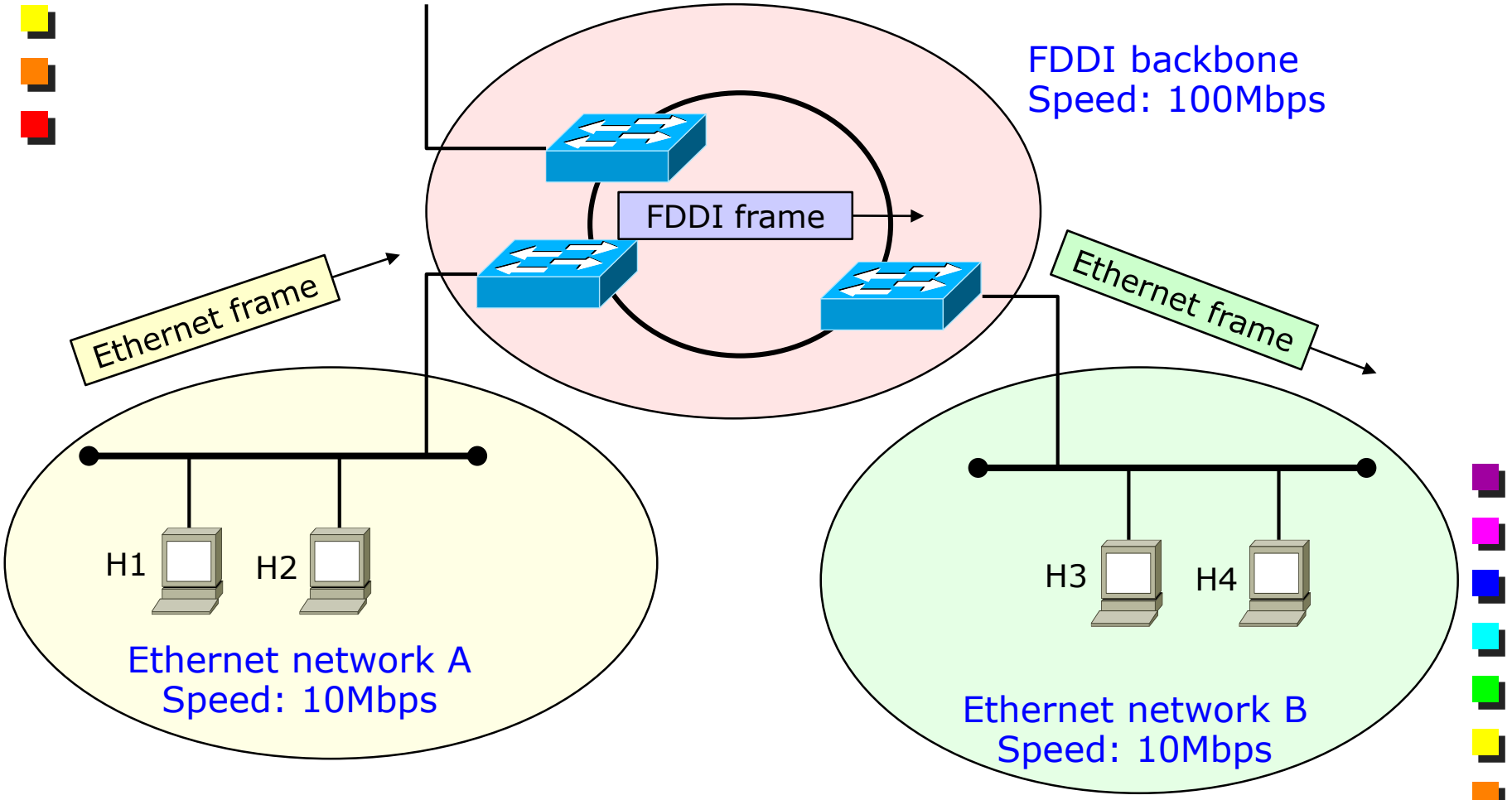
- Interconnection between different LANs using different technologies
  - E.g., Ethernet and WiFI
  - In practice it is often impossible due to maximum frame size issues (data-link does not have fragmentation)
- LAN extension (total diameter)
  - Especially useful for FastEthernet and upper speed (200m)
    - Collision domain issues
- Currently, bridges are often used for different reasons
  - Mainly speed (details will follow)

Bridge: operations

- Works by receiving and re-transmitting (later) a frame
  1. Store the frame (*store and forward* mode)
  2. Modify the frame (e.g. Ethernet to Token Ring)
  3. Send it out
- When a frame crosses a bridge
  - The L1 portion will be created from scratch
  - The L2 (MAC) portion will be regenerated (e.g., MAC conversion)
  - LLC and upper layers will transit unchanged

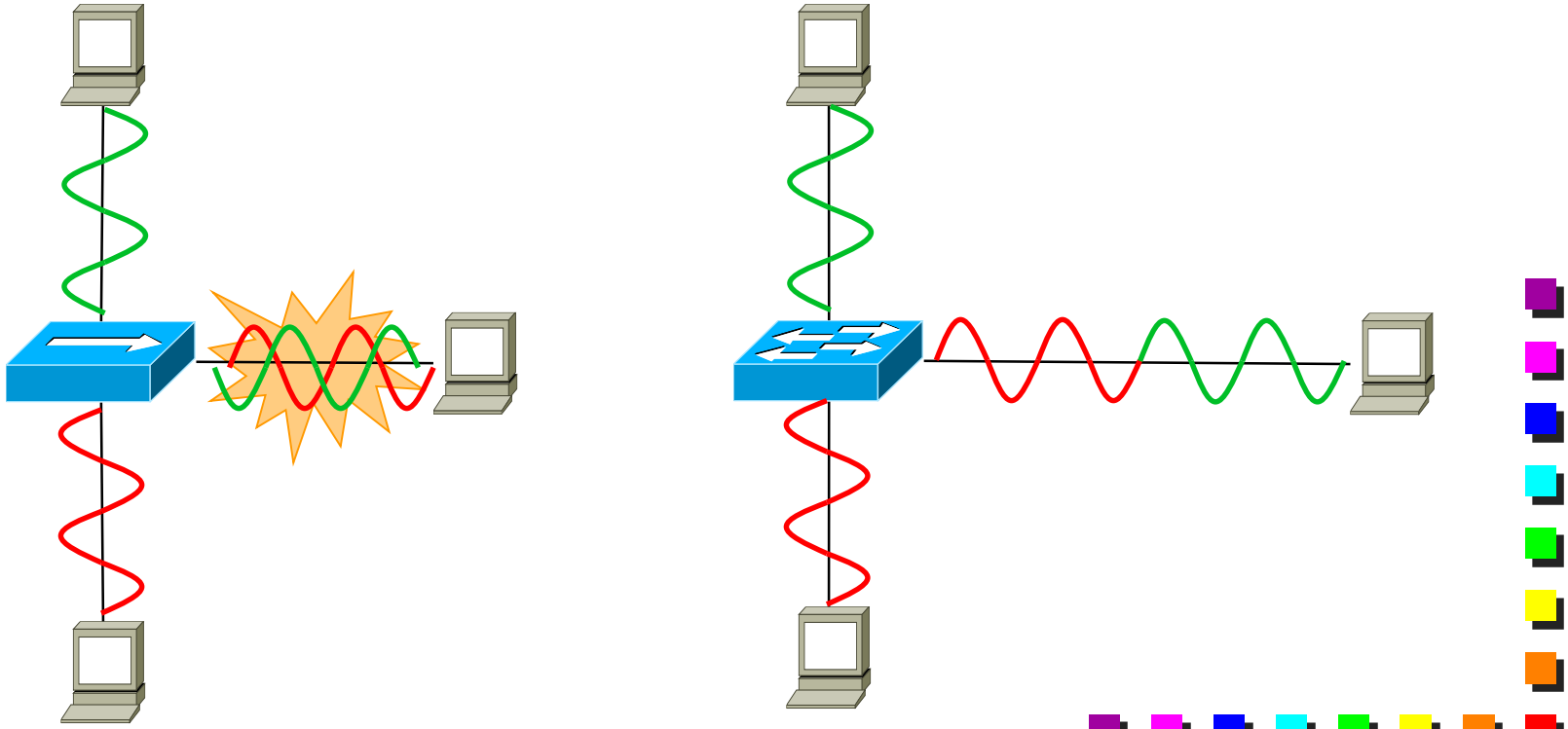


Bridge: example of interconnection



Bridges and collisions (on Ethernet)

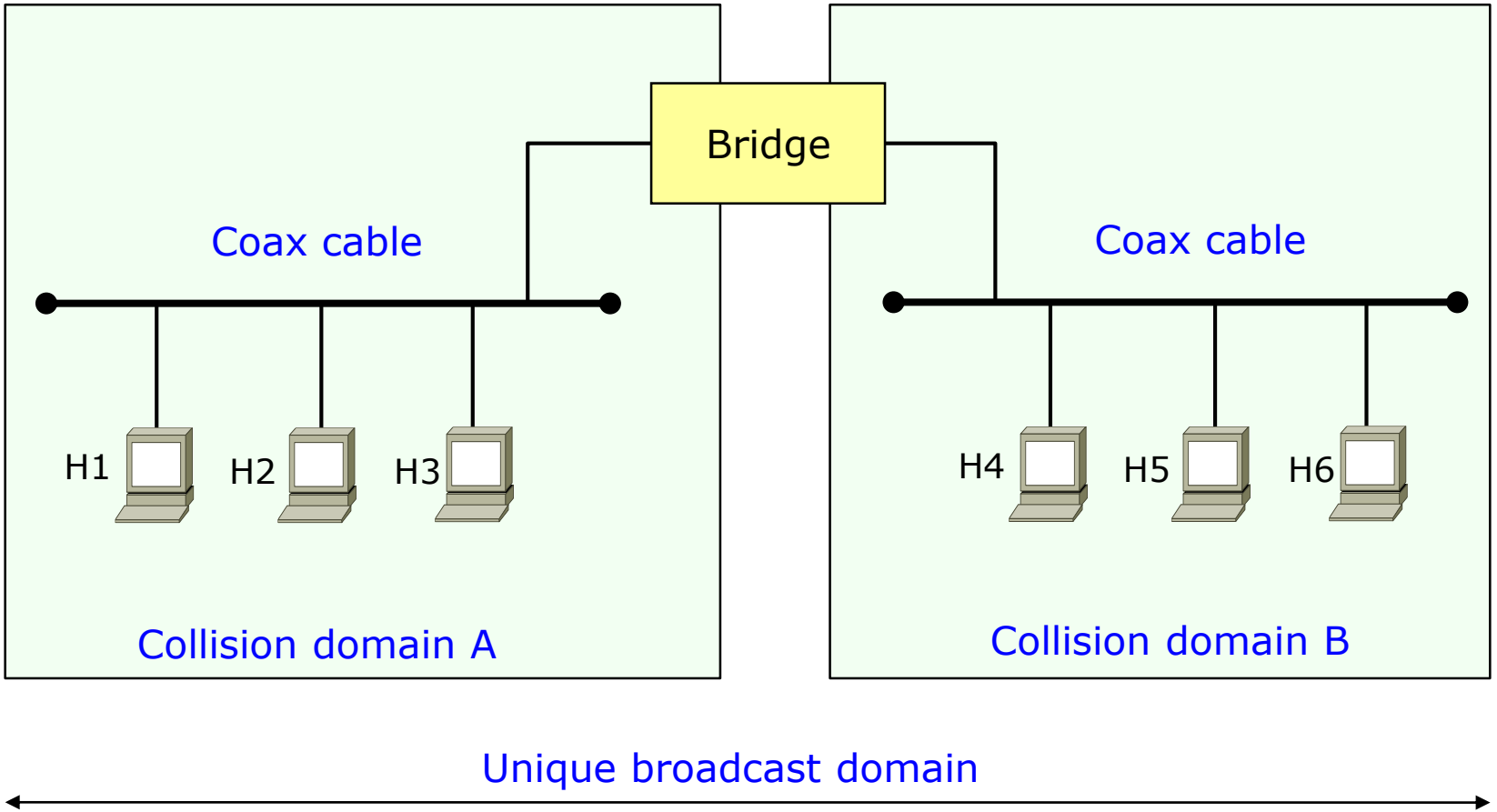
- "Store and forward" allows smarter sending of data on output interfaces
- Bridges **decouples collision from broadcast domain**
  - Collision domain no longer a limitation



Collision and broadcast domains (1)

- **Collision domain**: area where a single instance of the access control algorithm (e.g., CSMA/CD) operates
  - I.e., the area covered by a *single "physical" link*
  - Frames are immediately propagated over all the links (possibly through repeaters)
  - Also called **network segment**
- **Broadcast domain**: area where frames can be propagated
  - I.e., the area on which a *LAN* operates
  - Can include several collision domains
  - Frames can be stored and later propagated over le other collision domains
- So, from this point on:
  - LAN = *logical LAN* (i.e., *broadcast domain*)
  - *Physical LANs* are now referred as "*collision domains*"

Collision and broadcast domains (2)

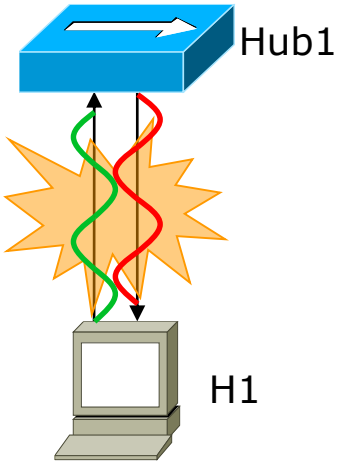


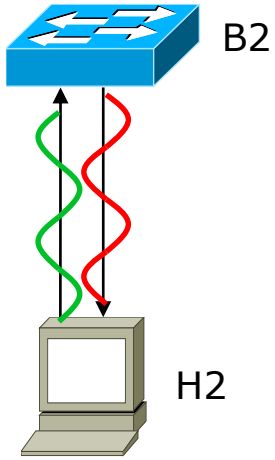
Collision and broadcast domains (3)

- Repeaters "extend" the collision domain
  - In fact, it is not actually "extended"; it allows the collision domain to reach its theoretical limits, despite cable limitations
- Bridges **create different collision domains** and **extend** the broadcast domain
  - I.e., bridges decouple broadcast domain from collision domain
- This is a very important feature of bridges, that comes out from their "store and forward mechanism"

Half and Full duplex mode

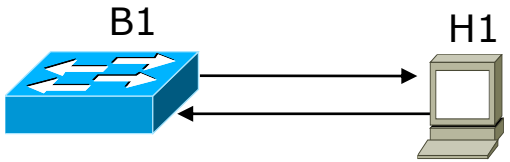
- Half Duplex mode
  - Standard operating mode of network interfaces (NICs)
  - RX and TX cannot happen at the same time
  - RX+TX activity is seen as collision
- But… if we have two physical links, do we really have a collision?





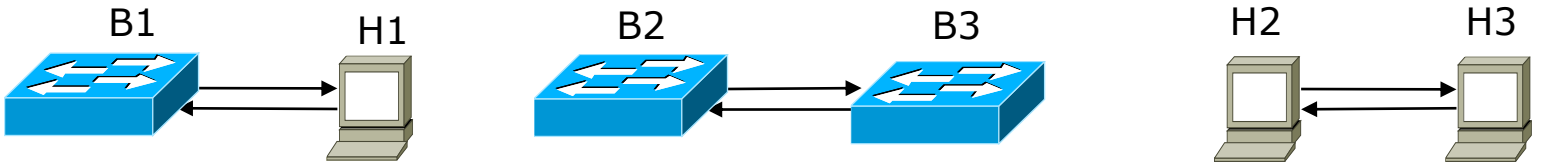
Full duplex (1)

- Full Duplex
  - Possibility, for a NIC, to send and receive data at the same time
  - Alternative operating mode than Half Duplex
- Two main reasons open the door to devices operating in "full duplex" mode
  - The "store and forward" operating mode of bridges
    - Collision on one port is not propagated to the other ports
  - Modern network cables use one wire for TX, one for RX
    - Twisted pair, fiber optic
- Can use both wires at the same time!
  - No problems of collisions, due to the splitting between collision and broadcast domains



Full duplex (2)

- Introduced with Fast Ethernet (part of 802.3x)
- Available whenever *the other party* can *temporarily store* the frame
  - Not just host ←→ bridge
  - Available when the other party stores the frame, instead of repeating (immediately) the received bits on the other ports, such as a repeater does
- Examples: host ←→ host, host ←→ bridge, bridge ←→ bridge

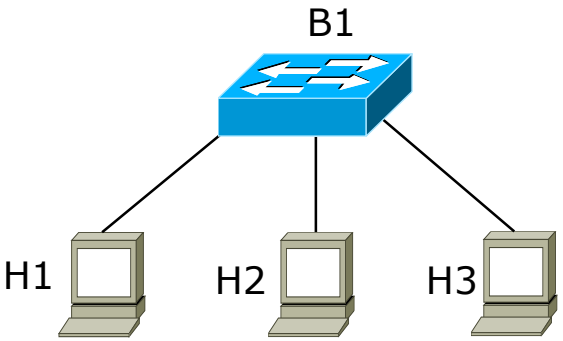


Full duplex: advantages

- Bandwidth
  - In theory, throughput x2
  - In practice, minimum advantage for clients and servers
    - Clients tend to saturate downlinks, servers uplinks
  - May be interesting for bridges on the backbone
    - More symmetrical bandwidth
- CSMA/CD
  - No longer needed, since collisions are no longer possible
    - With CSMA/CD, TX and RX together are used to detect collisions
  - Advantages
    - No requirement for min frame size for Ethernet
    - No limits on the network size on Ethernet (no collision domain)

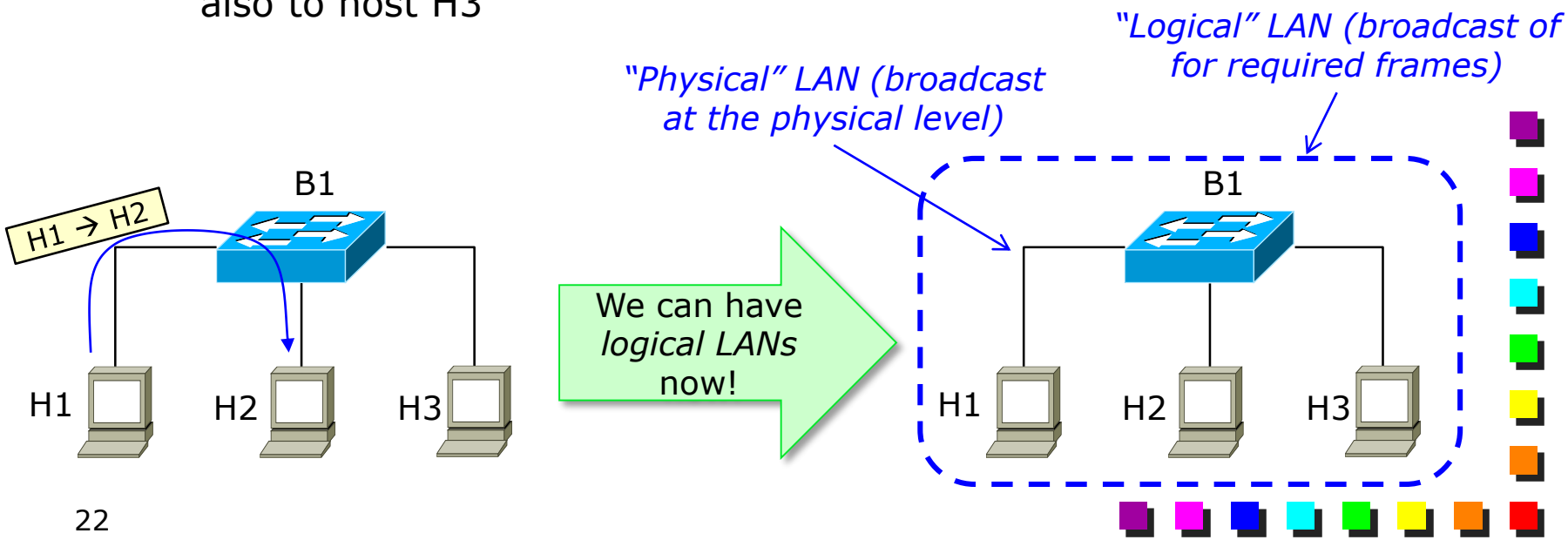
Full duplex and switches

- Modern networks are heavily based on full duplex
- Hub-and-spoke topology
  - Point-to-point connections between hosts and the “bridge”
  - No collision domain
- Multiport bridges are called “switches”
  - Same functions, different internal architecture

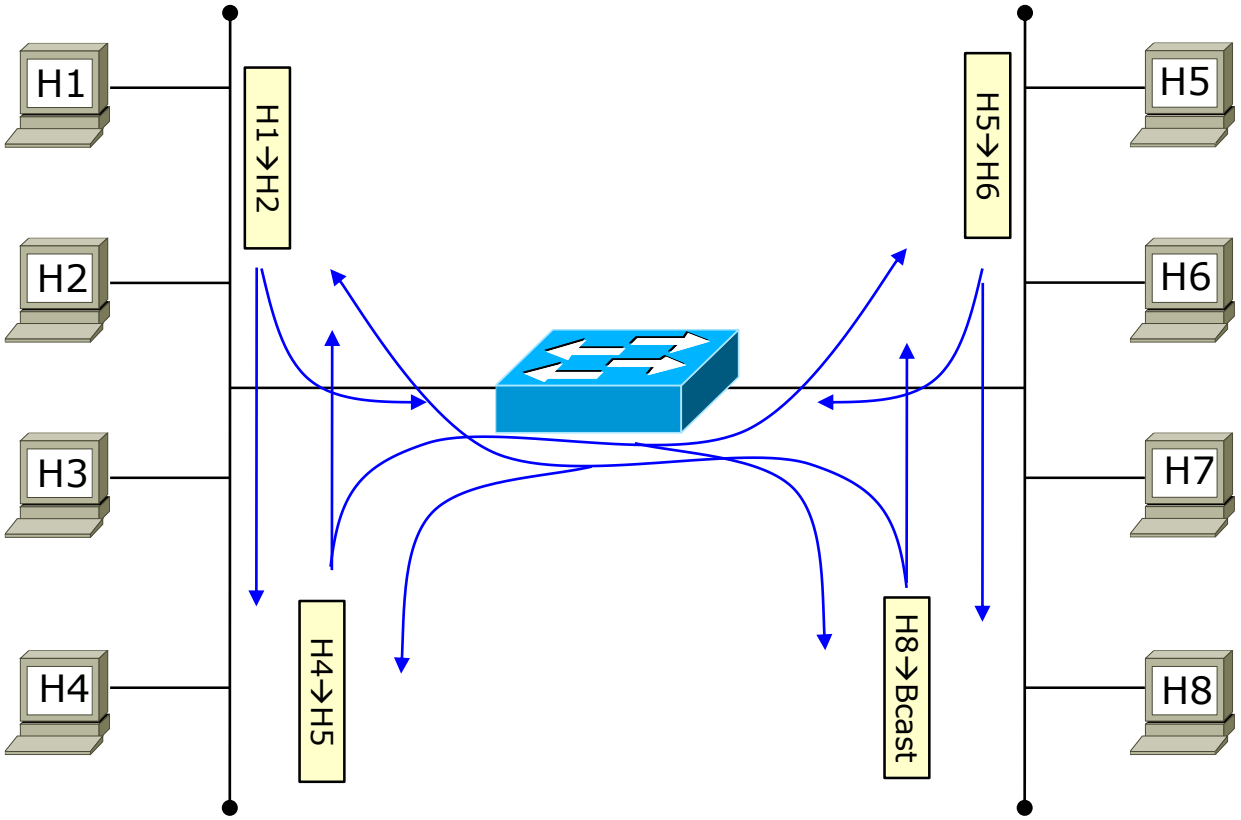


Should we introduce more intelligence in bridges?

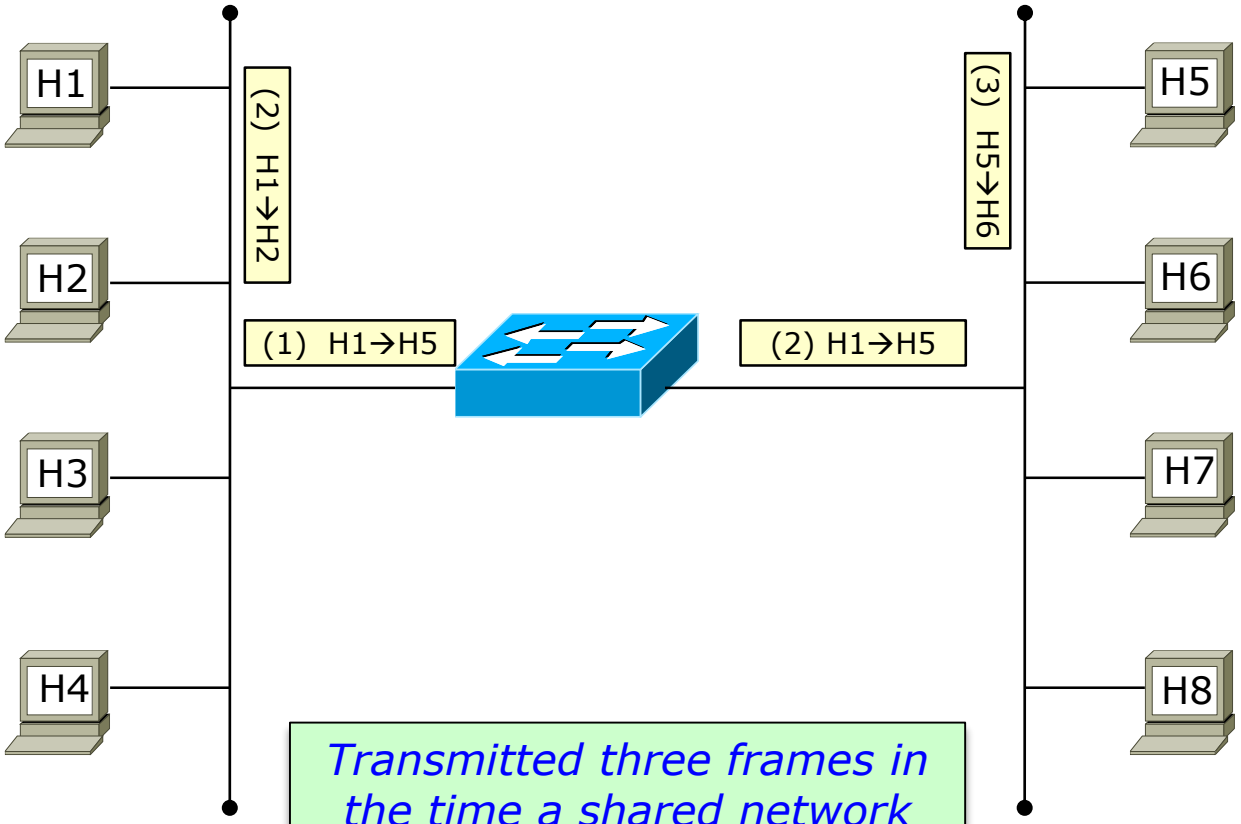
- By definition, a LAN is a *shared communication medium*
  - But… not necessarily at the physical layer
  - This feature can be emulated at logical level
- What is important is that each frame will arrive to the legitimate recipient
  - If a frame is directed to host H2, is not important that arrives also to host H3



Smart forwarding process (1)



Smart forwarding process (2)



Bridge: smart forwarding process (3)

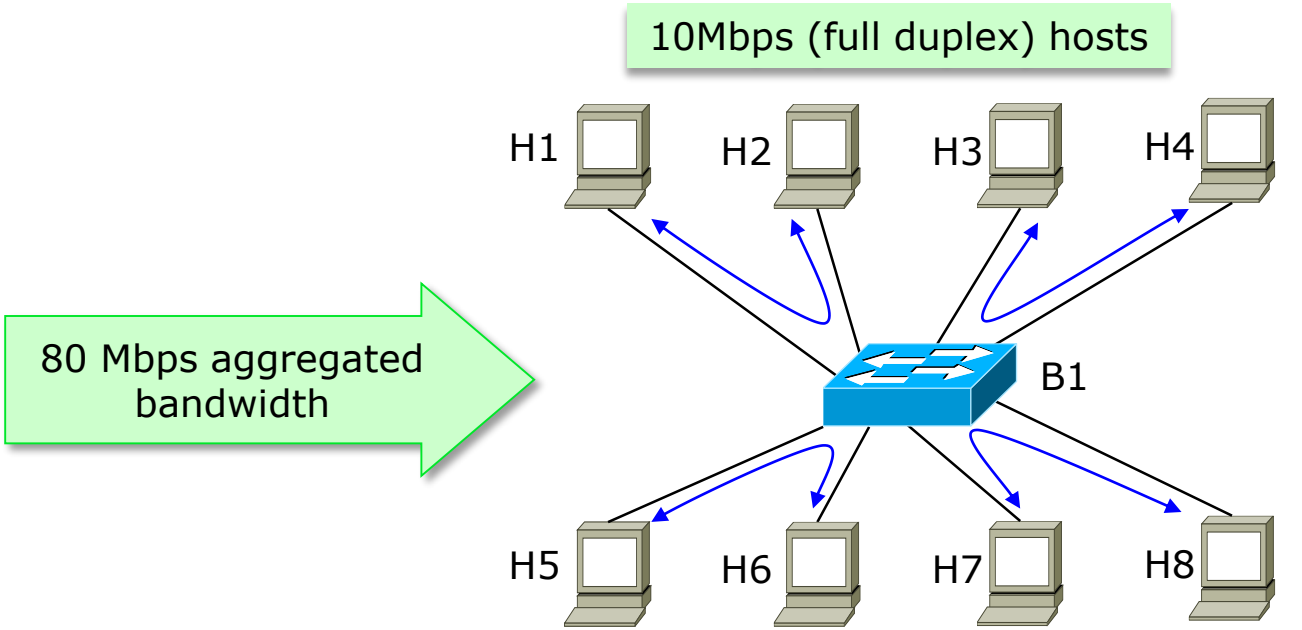
- Bridges, in principle, should propagates each frame in the entire network
  - Not really useful in practice
    - Apart from being able to sniff my friend's traffic
- What about implementing a smarter forwarding process, forwarding to everyone only the frame we really care about?
  - E.g., broadcast frames
- Let's modify the bridge operations so that
  - Receives a frame on one interface
  - Stores the frame into a local buffer
  - Analyzes the destination address
    - Why is the MAC Dst before the MAC Src in Ethernet frames?
  - Forward it on the right port (if needed)

Smart forwarding process (4)

- But… ports belong to different collision domain, therefore a bridge can send/receive traffic at the same time over different ports
  - Bridges have buffers in order to absorb bursts and to wait for the proper transmission slot
- Very important by-product: if the bridge has a smart forwarding process, it can implement traffic segregation
  - Increase the aggregate bandwidth of the network
  - Forwarding technique based on MAC Destination address
- Right now, the most important reason for using bridges!

Smart forwarding process (5)

- Single-link bandwidth does not change
- *Aggregate* bandwidth increases
- Why did we say that the "shared medium" is more efficient?
  - Well… in 1980 it was more efficient, but we are now in 2010+
  - Now, "switching" is possible at very limited costs



Smart forwarding process (6)

- Smarter forwarding rules
  - Unicast: only on the port toward we can reach the destination (Destination MAC-based forwarding)
  - Multicast, broadcast: flooding
    - All ports except the port on which the frame has been received (*flooding*)
- A MAC forwarding table must be available locally
  - Filtering database (more details later)
- A note about flooding
  - Frames are sent on all ports (expect on the incoming port), but **may not be sent at the same time** (delayed forwarding)
  - Hubs send data in flooding at the same time (*bits* forwarded immediately)

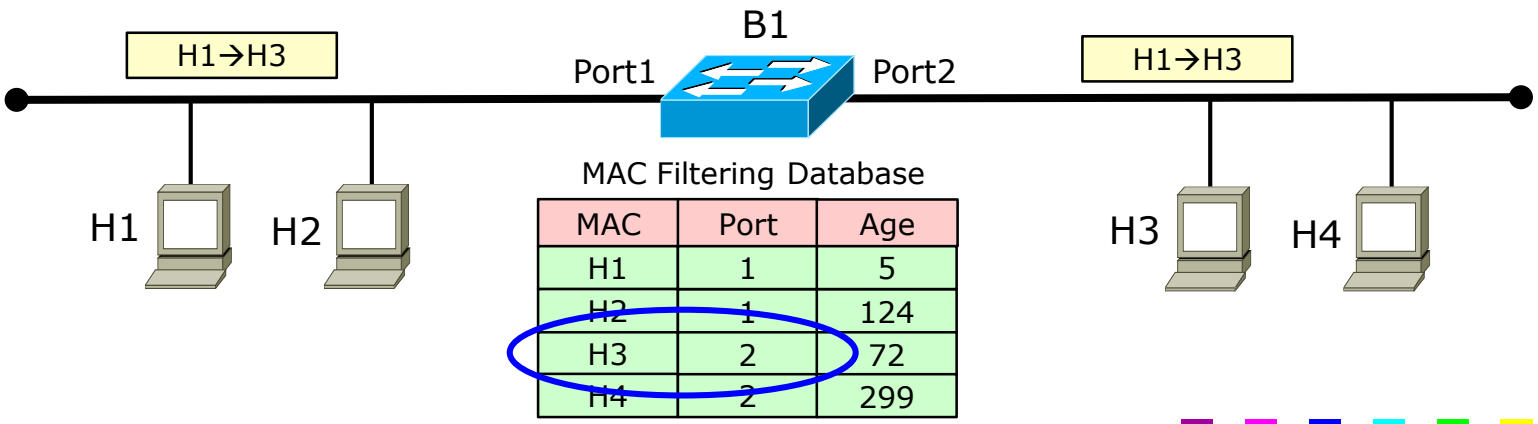
New components in “smart” bridges

- In order to operate successfully, a “smart” bridge requires three additional components:
  - A local forwarding table (*filtering database*)
  - Stations auto-learning (*backward learning*)
  - Loop detection (*spanning tree algorithm*)

- The ultimate goal: the bridge should be able to do its job without any explicit configuration from the network admin
  - Really “plug and play”
  - By-product: stupid network admins believe they are really smart just because their networks work properly

Filtering Database (1)

- Table with the "location" of any MAC address found in the network
  - MAC address
  - Destination port
  - Ageing time (default expire after 300 s)
- "Filtering" database: in the old days, the smart forwarding process was perceived as a way to "filter out" unwanted traffic from a link



Filtering Database (2)

- Entry types
  - Dynamic
    - Populated and updated by the backward learning process
    - Max entries: 2 ÷ 64 K
  - Static
    - Not updated by the learning process
    - Usually < 1K entries
- Old dynamic entries are purged out of the filtering database
  - E.g., stations that do no longer exist on the network
  - Default: 300 seconds

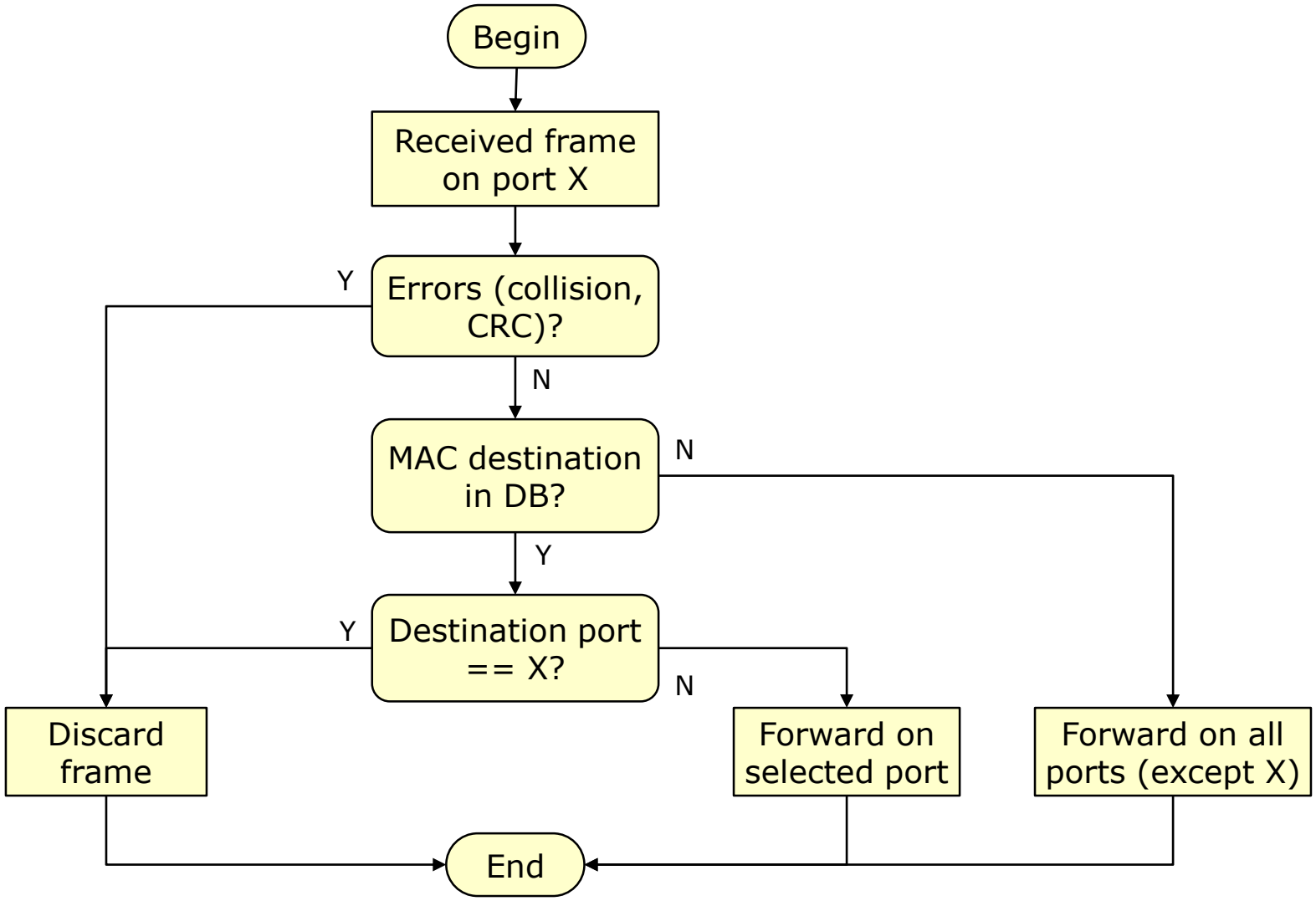
Filtering database: real example

```
Cisco-switch-1> show cam dynamic

* = Static Entry. + = Permanent Entry.
# = System Entry X = Port Security Entry

Dest MAC Address     Ports Age
------------------   ----- ---
00-00-86-1a-a6-44    1/1   1
00-00-c9-10-b3-0f    1/1   0
00-00-f8-31-1c-3b    1/2   4
00-00-f8-31-f7-a0    1/1   2
00-01-e7-00-e3-80    2/2   0
00-02-a5-84-a7-a6    2/1   1
00-02-b3-1e-b4-aa    2/1   5
00-02-b3-1e-da-da    2/5   1
00-02-b3-1e-dc-fd    2/4   2
```

Forwarding process



Forwarding process and transient

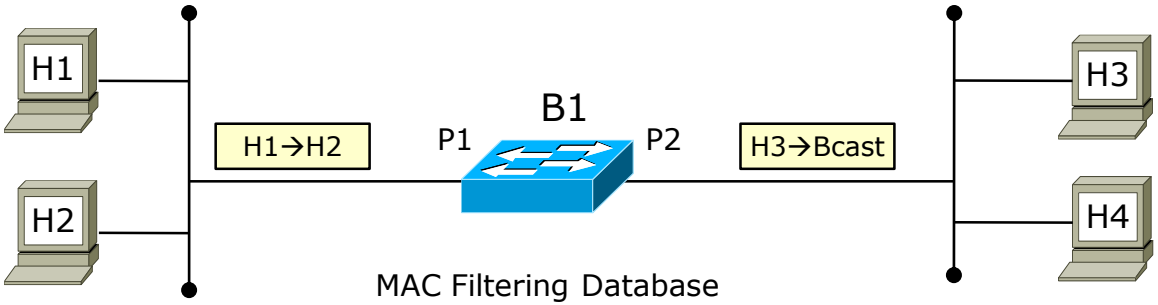
- What about if a MAC address is not present in the Filtering Database?
  - Bridge looks like an hub
    - Frame duplicated on all ports except the one on which it was received
- This situation is rather common and it is called “transient”
  - Bridges are plug-and-play and have an algorithm to learn the *location* of the hosts
    - Backward learning (presented later)
  - However, at the beginning, bridges do not know where an host is located
  - In this case the “MAC Flooding” algorithm is the only way to go

How do we populate the filtering database?

- 1) By hand
  - Possible on all modern devices, but not very handy
- 2) By means of a proper algorithm
  - Backward learning
    - The best choice, of course

Backward learning (1)

- The idea
  - If a bridge receives a frame whose source is host H1 from port P1, that host will be reachable through port P1
- Topology is learned by inspecting **received** frames
  - Analysis of MAC source address
  - The destination MAC address is ignored by this algorithm

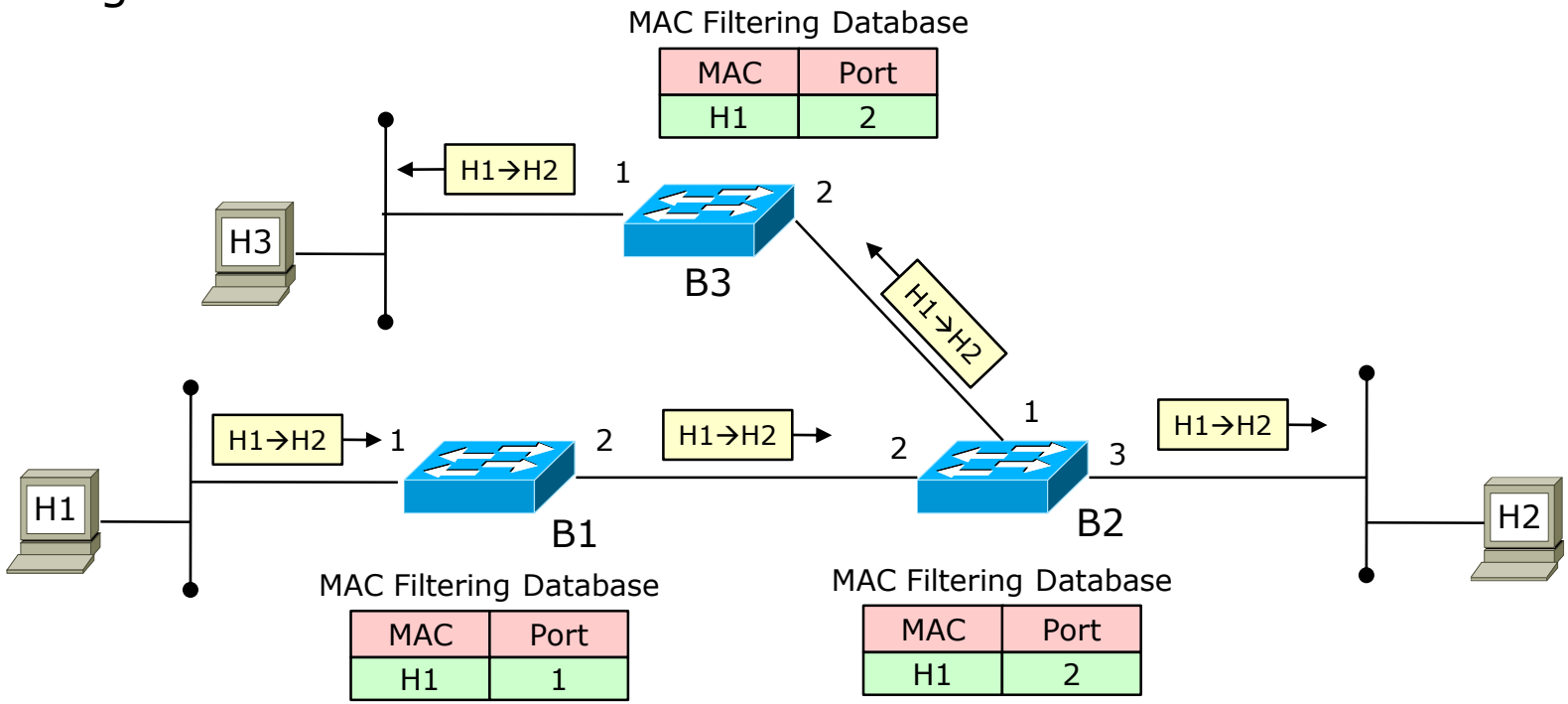


MAC Filtering Database

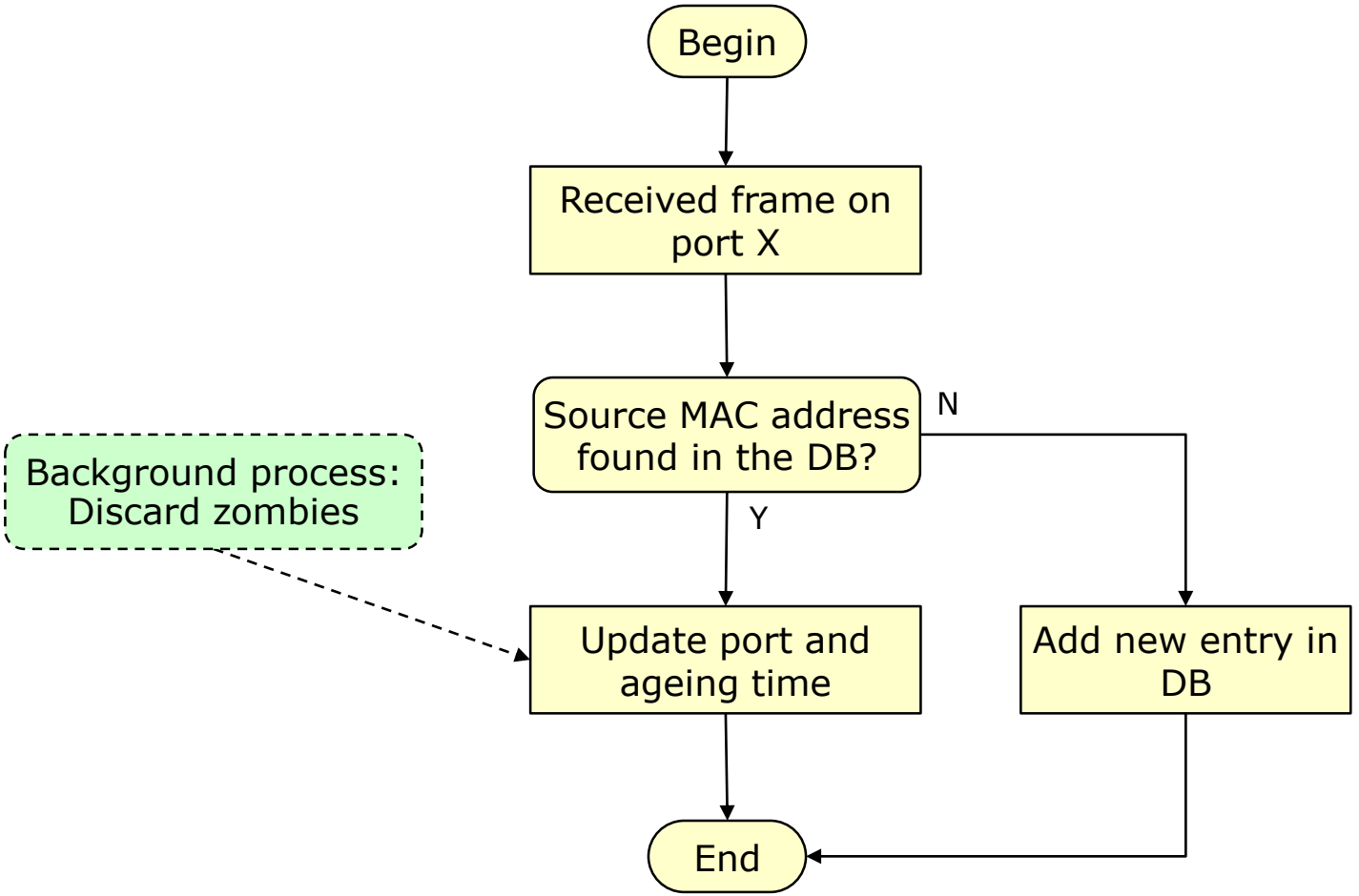
MAC	Port
H1	P1
H3	P2

Backward learning (2)

- Works also in presence of multiple bridges
  - Remote bridges learn the position anyway, even if the end-system is not connected locally
- Example: backward learning and frame forwarding taken together



Backward learning (3)

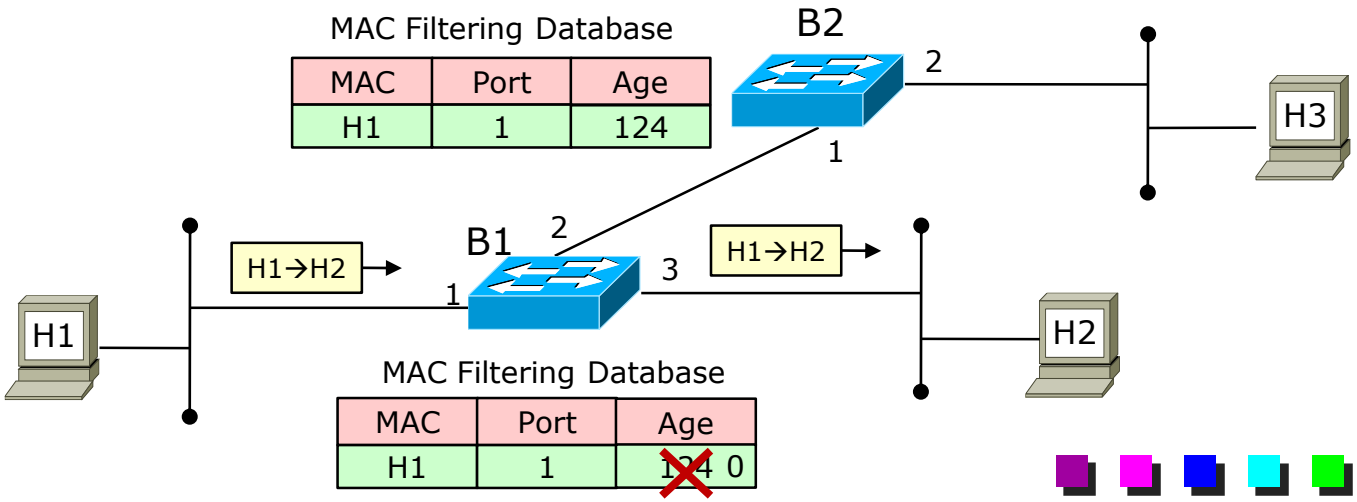


How do we keep the filtering DB up to date? (1)

- Update the Filtering database means…
  - Refresh “Age”, so that the entry keeps alive
  - Refresh “Port”, so that the host is updated with the new position

How do we keep the filtering DB up to date? (2)

- Broadcast (multicast) frames
  - Best option, since they are propagated across the entire network
  - E.g., solicitation/advertisement (e.g., ARP Request)
  - Multicast: take are of IGMP snooping or similar
- Unicast frames
  - E.g. loquacious hosts
  - Better than nothing, but they may not be propagated across the entire network



Some notes about filtering database

- Please note that…
  - An end-system whose MAC address is not in the DB is *always reachable*
    - Corollary: a frame sent to a non-existing host will always be forwarded in all the network
  - An end-system whose MAC address is in the DB may be *unreachable*
    - At most for *Aging Time*, in fact

Possible attacks to the filtering database

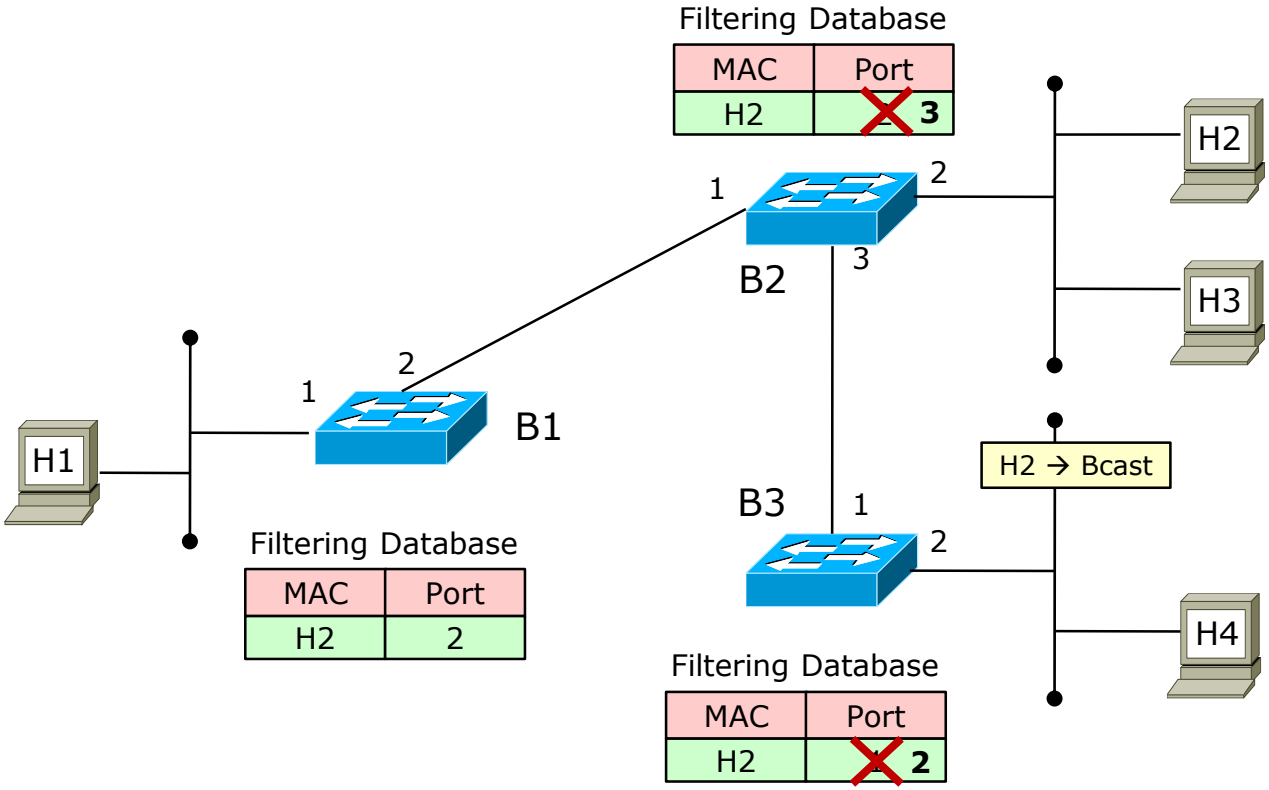
- MAC Flooding Attack
  - Generation of frames with random MAC sources
  - Filtering database gets full
  - Bridges will start flooding most of the frames
    - All the ones whose destination address is not present in the DB
  - Objectives
    - Forces bridges to operate like hubs, so that we can intercept traffic generated by other stations
    - Slows down the network
  - Some vendors give the opportunity to limit the number of MAC address learnt on each port

Possible attacks to the filtering database (2)

- Packet storms
  - Generation of frames to non-existing stations
  - Frames are always send to the entire network
  - Objective
    - Slows down the network

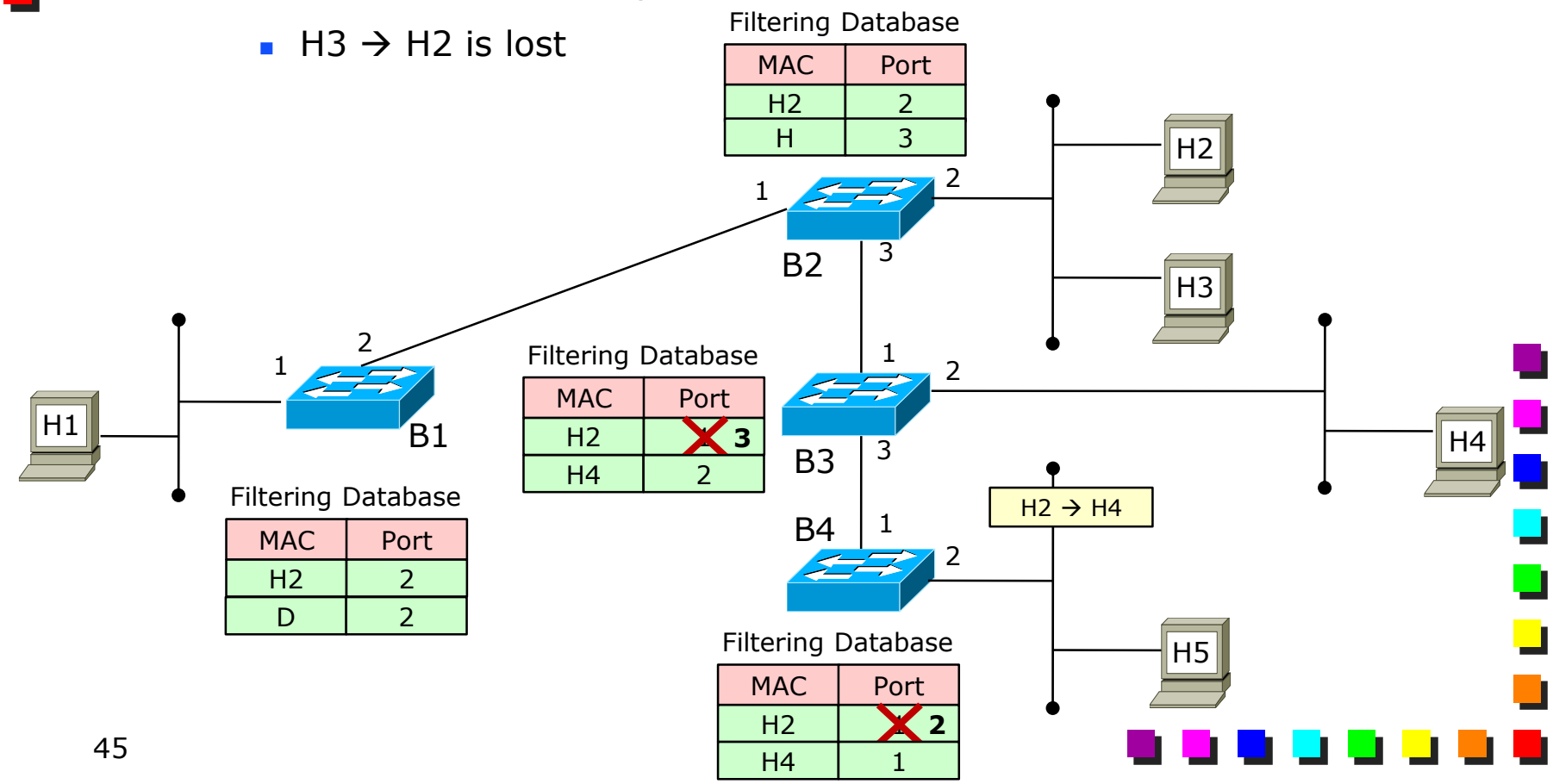
L2 networks and hosts mobility (1)

- If the end-system generates broadcast frame immediately
  - No problems



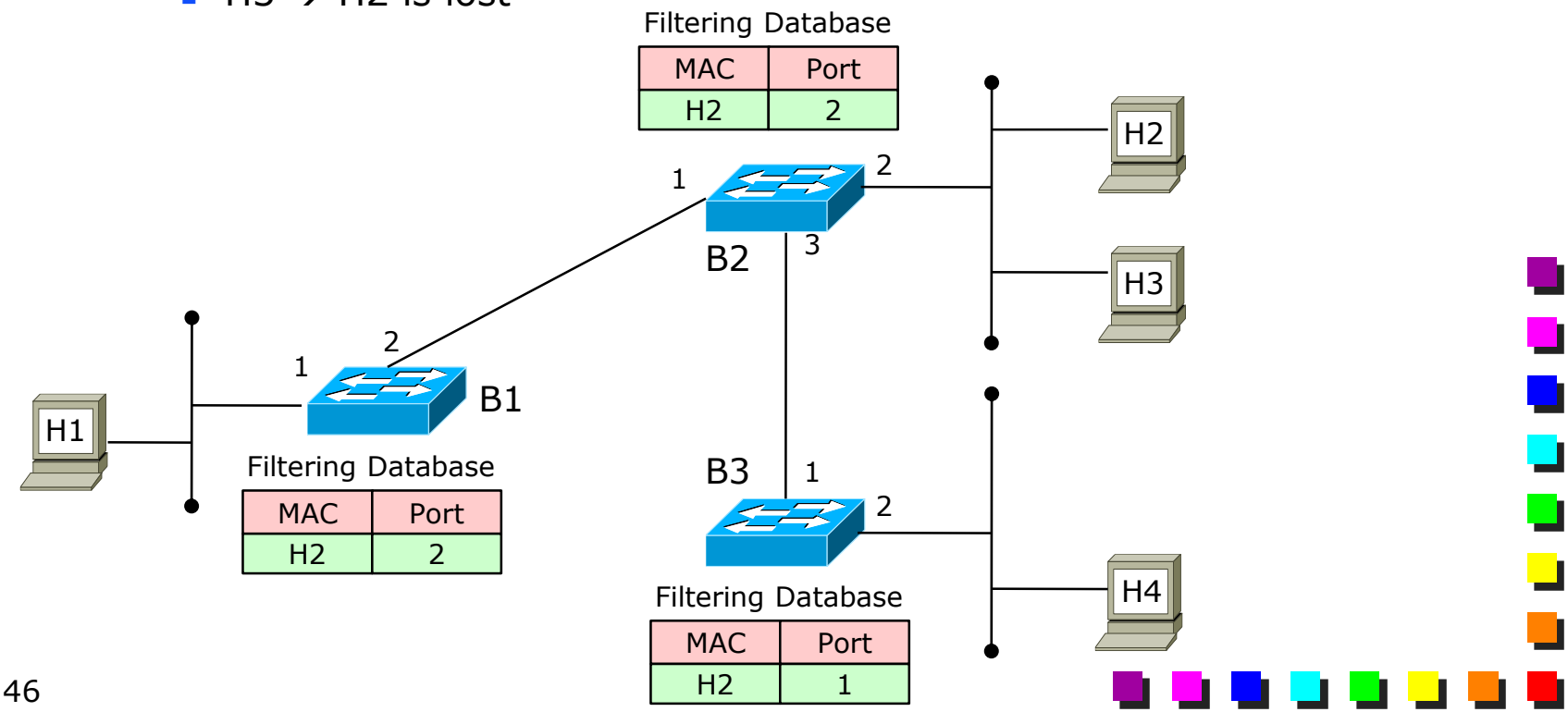
L2 networks and hosts mobility (2)

- If the end-system generates unicast traffic immediately
  - We may have forwarding errors
    - H4 → H2 is correctly delivered
    - H3 → H2 is lost



L2 networks and hosts mobility (3)

- If the end-system does not generate traffic at all
  - We may have forwarding troubles
    - H4 → H2 is correctly delivered
      - The frame is forwarded also to the original destination
    - H3 → H2 is lost



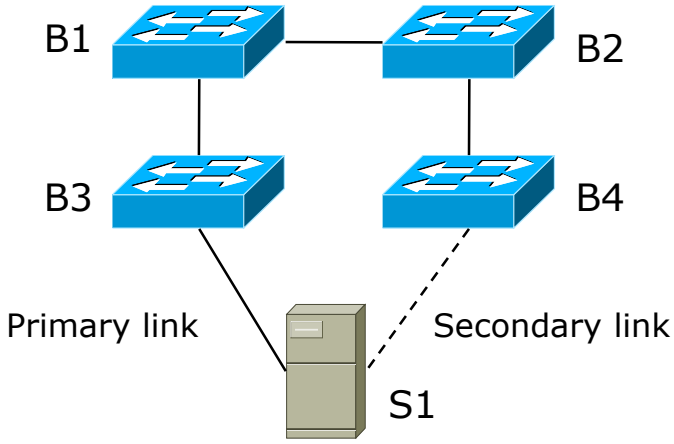
L2 networks and hosts mobility (4)

- Broadcast (multicast) frame
  - Receives all the network, therefore all the bridges update the location of the current station
- Unicast frame
  - Potentially reaches only a portion of the network, hence the rest may still have the old location of the station

- In the real world
  - Windows host typically generates a lot of broadcast
    - No problems when moving from one place to another
  - UNIX servers and virtualized hosts (e.g., Vmware) are often silent if not solicited
    - Need to wait for the aging time

L2 networks and hosts mobility (5)

- The aging time
  - Usually enough in order to cope with manual movements
    - A laptop moved from office to lab
  - Represents the worst-case black-out time for an end system
- Some problems may appear in specific environments
  - E.g. fault-tolerant NICs
    - We need to react much quickly than 5min
  - NIC driver has to generate an additional broadcast frame



L2 networks and mobility (6)

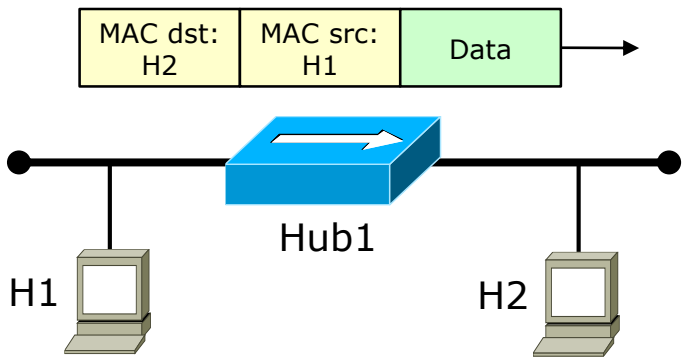
- L2 networks natively support mobility
  - Just enough to send some broadcast immediately
- Mobility at L3 level requires MobileIP, which is often not available on real devices
- Lesson learned
  - Much easier to move an host at L2 level and keep all the connection active (the IP address does not change)…
  - … than move an host at L3 level (which requires to change the IP address)

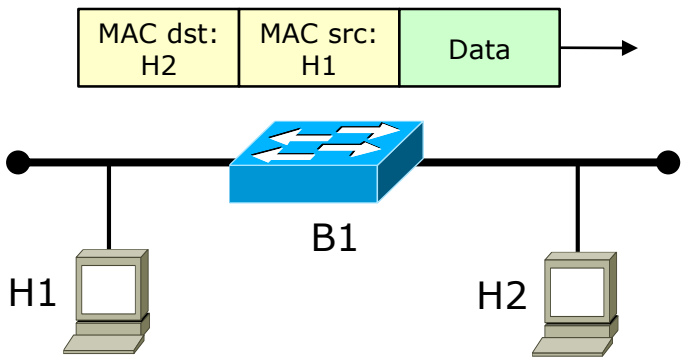
Transparent bridges

- Bridges that we presented so far are called **transparent bridges**
  - Originally proposed by Ethernet
  - Other (non transparent) bridges have been proposed in the past (e.g. Token Ring networks)
    - No longer in use
- Transparent bridges standardized by IEEE in 802.1D
- Transparency
  - Bridges should be plug-and-play and must not require any change in the configuration of the end systems
  - Performance (throughput, max distances) may be different from the original network, but functionalities are the same

Transparent bridges and end hosts (1)

- End systems must operate in the same way (same frames, some format, etc) with or without bridges
- In details
  - No changes at all in frames sent by end systems
    - Same frame, same src/dst MAC address, etc…
  - There may be some changes in **which** frames are received
  - No changes at all in the **format** of the received frame
    - Same source/MAC address, etc





Transparent bridges and end hosts (2)

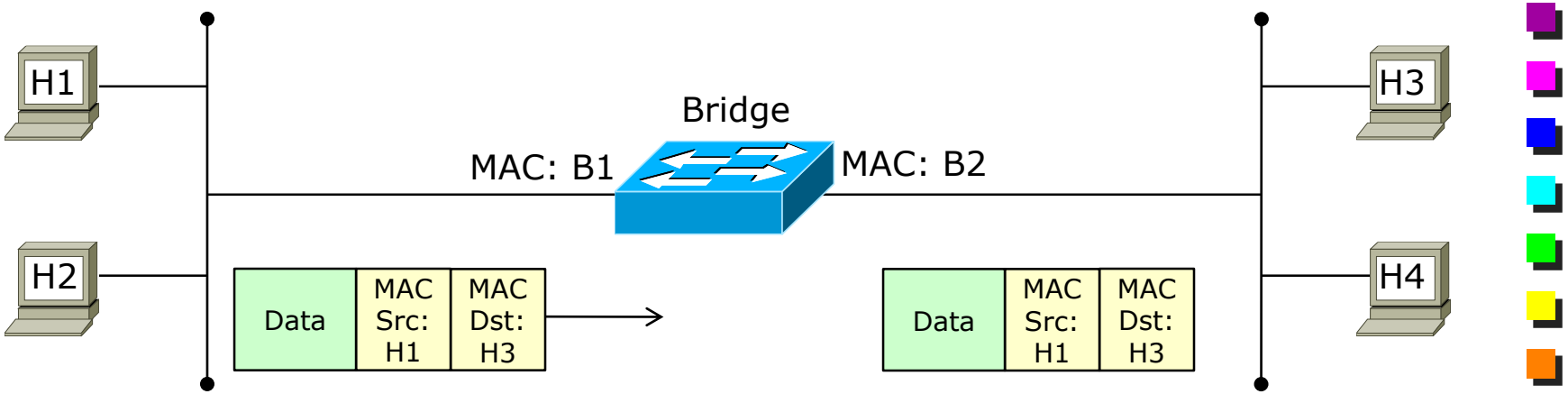
- Hosts do no longer receive all frames
- An host connected to a bridge will receive:
  - All the frames sent/received on the current network segment
  - All broadcast/multicast frames
    - IGMP snooping or such excluded
  - Unicast frames whose destination MAC address is equal to its own MAC address
- An host with a point-to-point connection to a bridge
  - All broadcast/multicast + unicast frames directed to it
- This is valid if filtering database is fully updated
  - No transient

Transparent bridges and end hosts (3)

- What about frames received by the *OS* on the host?
  - No changes at all
  - Bridges filters frames that were previously filtered by the NIC
    - The result at the OS level is the same
- May become less important
  - MAC filtering on the NIC
    - Still useful during the bridge transient period and for multicast MAC addresses
  - NIC in promiscuous mode

Transparent bridges and port addresses

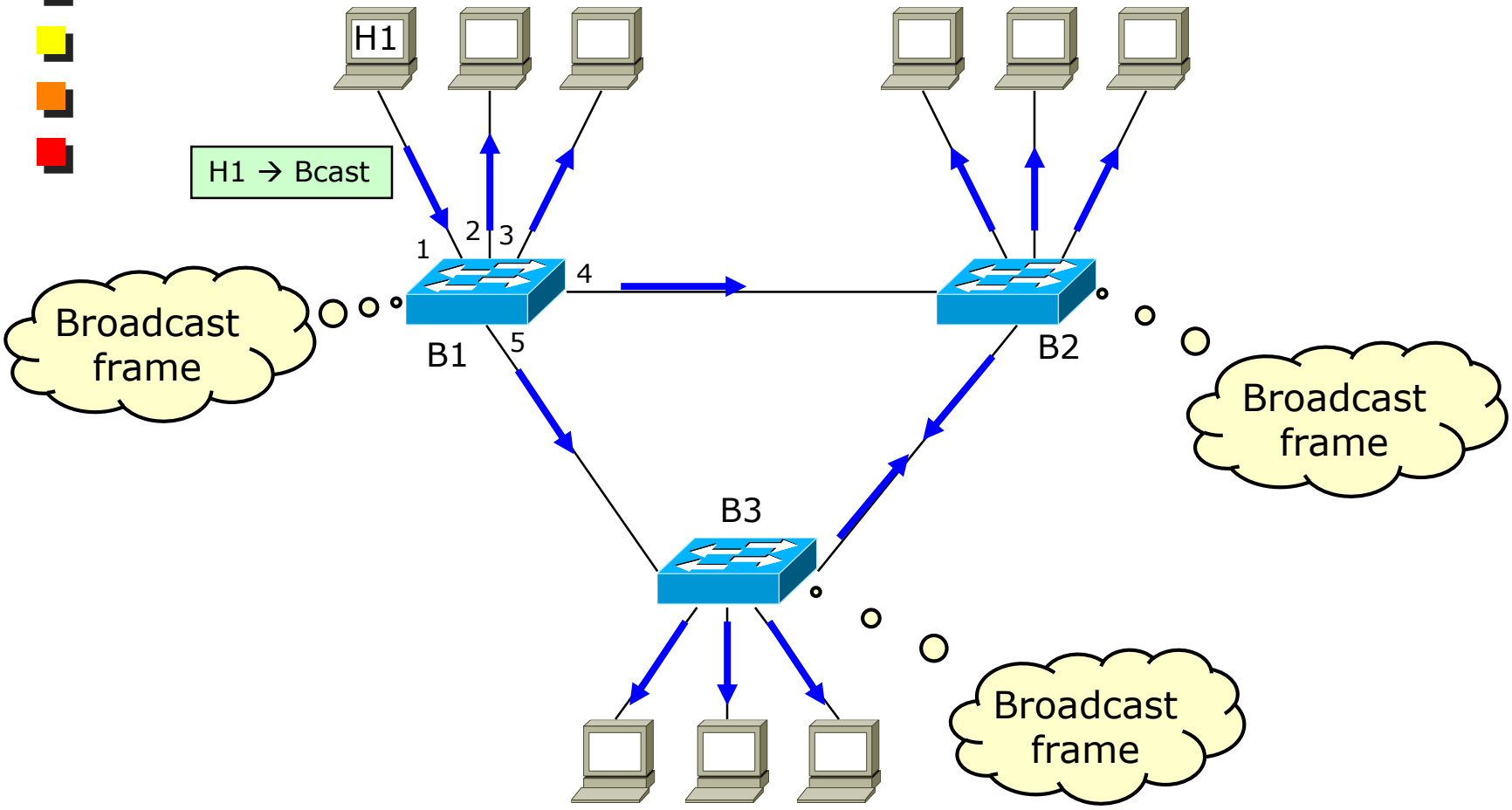
- Each port of a bridge has a MAC level and therefore **it has** a MAC address
  - That MAC address is never used when forwarding data frames
  - It is used when frames are generated/received by the switch itself
    - E.g. management frames



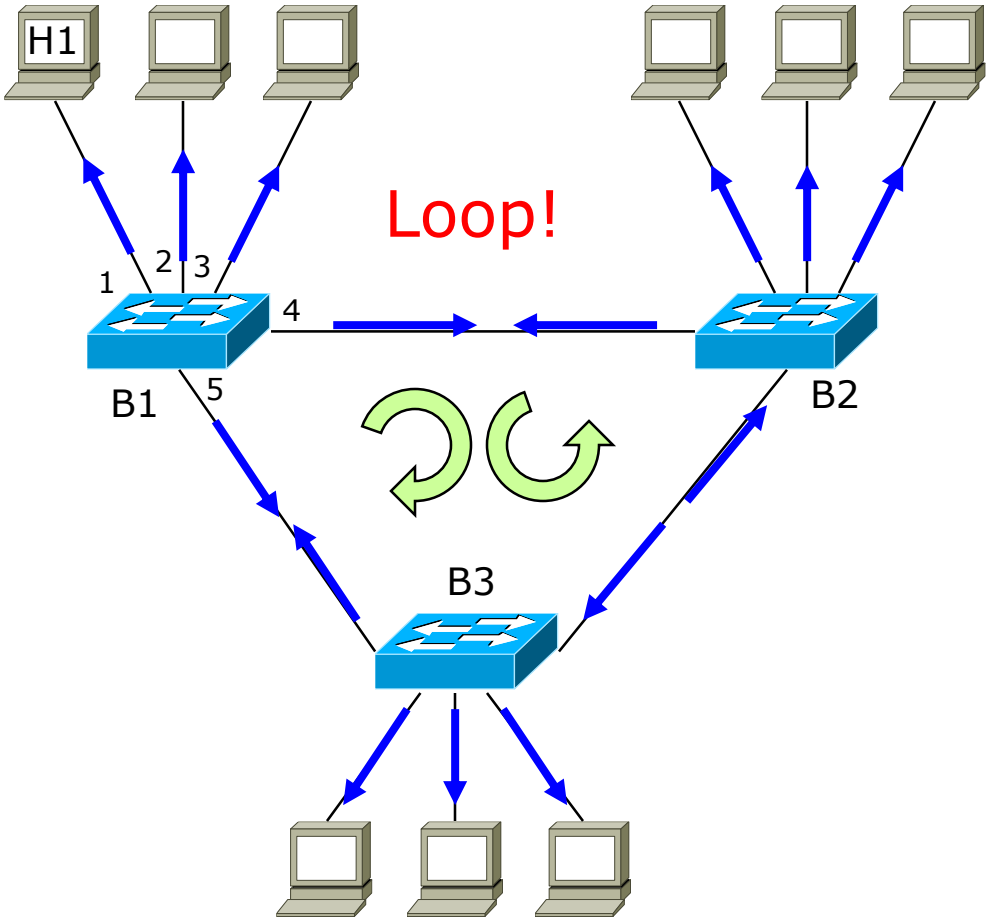
Bridges and meshes

- Two problems
  - Frames can enter in a loop
  - Backward learning no longer able to operate
- It’s now the time to present the third component (i.e. “Spanning Tree”) after the ones we presented earlier
  - “Filtering Database” and “Backward Learning”

Bridges and meshes: the loop problem



Bridges and meshes: the loop problem



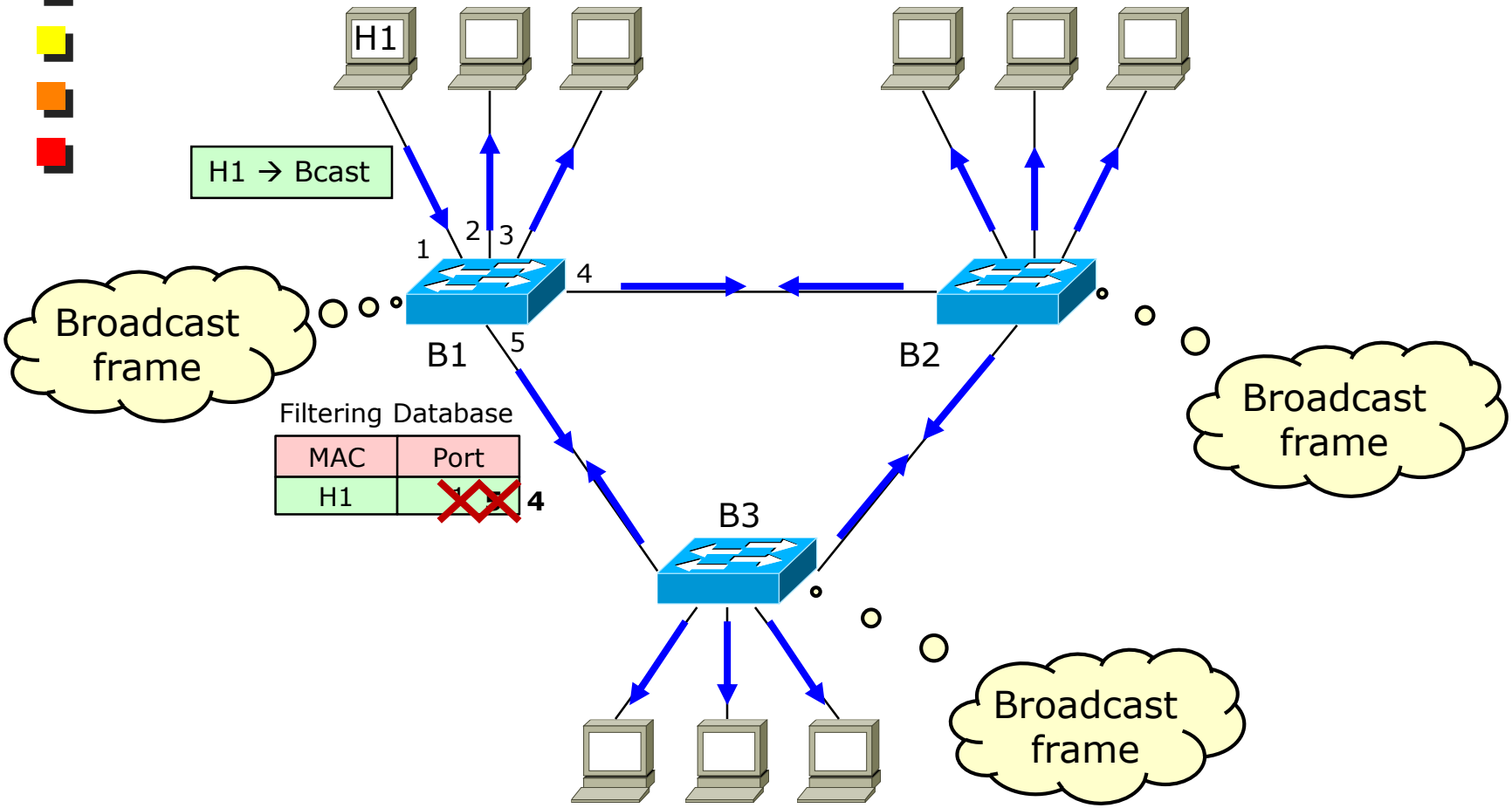
Which frames can generate a loop?

- Multicast/broadcast frames
  - Very common
- Frame to a non-existing station
  - MAC address not present in the filtering DB (e.g. non existing station)
  - Problem that may happen rarely (unless under attack)
    - IP sends an ARP before contacting an L2 station
    - If the station does not exist, the ARP will never get a reply and the destination MAC address is unknown
    - Therefore, no MAC frames will be sent to that station intentionally

The Broadcast Storm

- Massive load due to broadcast/multicast traffic on a LAN
- One of the most dangerous problems at data-link layer
- No solutions, except for disabling (physically) loops
  - E.g., detach a cable from a bridge
- Network operators are almost impotent in such this case
- Due to the lack of a “time-to-live” field in L2 frames
- L3 networks can tolerate transient loops
  - TTL available on L3 packets
- Can be used to create a low-cost traffic generator sending frames at line-rate

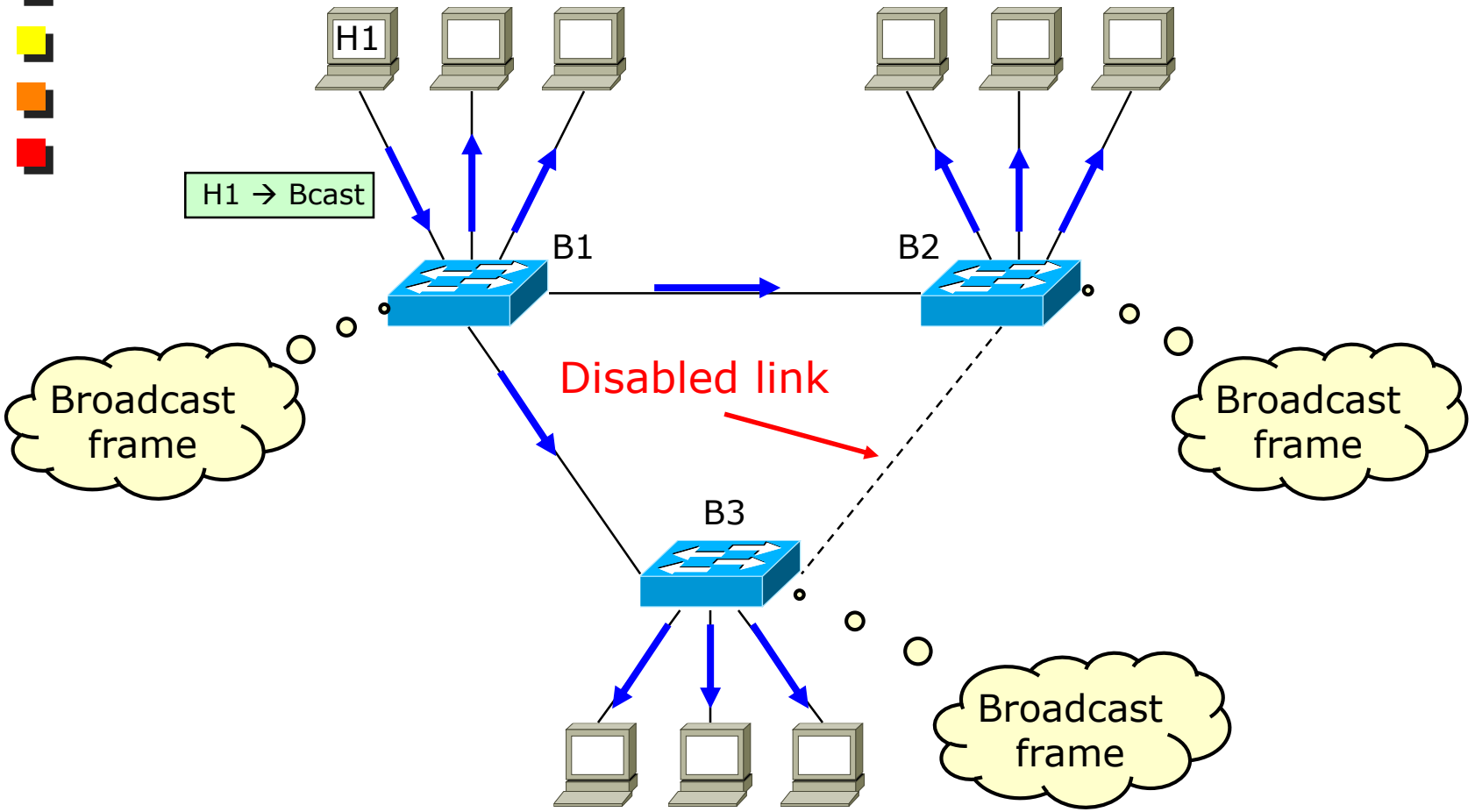
Bridges and meshes: the learning problem (1)



Bridges and meshes: the learning problem (2)

- Backward learning problem
  - Switches may have inconsistent filtering database
  - An entry in the filtering database may change the port indefinitely
    - An entry may not able to reach a stable state
    - Transient loops can be created among back-to-back bridges
      - B1 forwards to B2 that forwards to B1,…
      - Larger (B1-B2-B3-B1) loops may occur as well

The Spanning Tree idea: no loops in the network



Spanning Tree

- In order to avoid troubles, you must avoid loops in the physical network
  - Either create loop-free networks
    - Discouraged; not robust
  - Or define al algorithm that disables (temporarily) loops
- 802.1D
  - Original idea from Radia Perlman, PhD @DEC
- Meshes detected and disabled; the network becomes a tree
  - Unique path between any source and any destination
- Operates periodically (every second)
  - Decides which port set to forwarding state and which port set to blocking state
- More details in another set of slides

Bridges: pro

- Transparent to the network stack (and to the application)
  - Work with all L3 protocols
- Automatic configuration (although not optimal)
  - Backward learning
- Allow automatic reconfiguration in case of faults
  - Spanning Tree
- Increase performance (different collision domains)

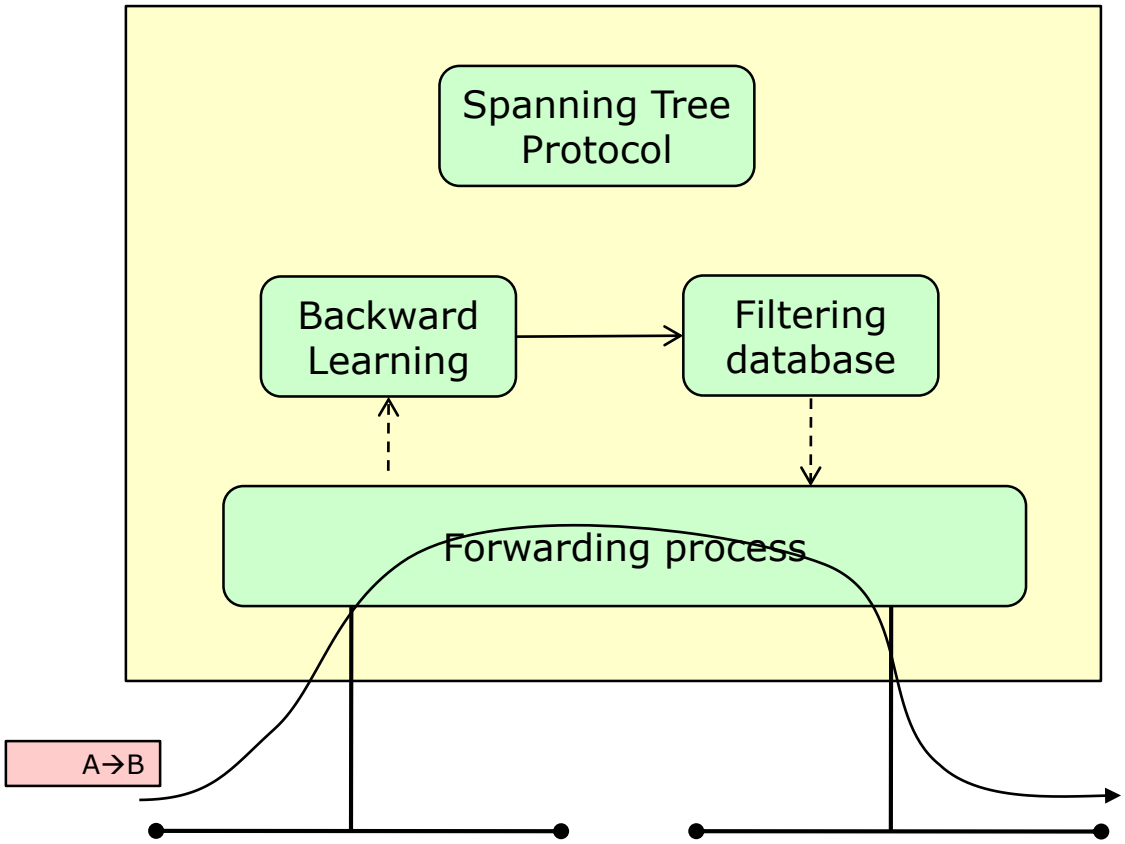
- Plug and play!
  - Often installed by technicians who do not know anything about the network

Bridges: cons

- Not suitable for complex networks (e.g. WAN)
  - Broadcast traffic, Spanning Tree blocked links
- No filtering for broadcast frames
- No load balancing over multiple parallel links

- Spanning Tree configuration is, in practice, required in complex network

Bridge architecture



Bridges and switches (1)

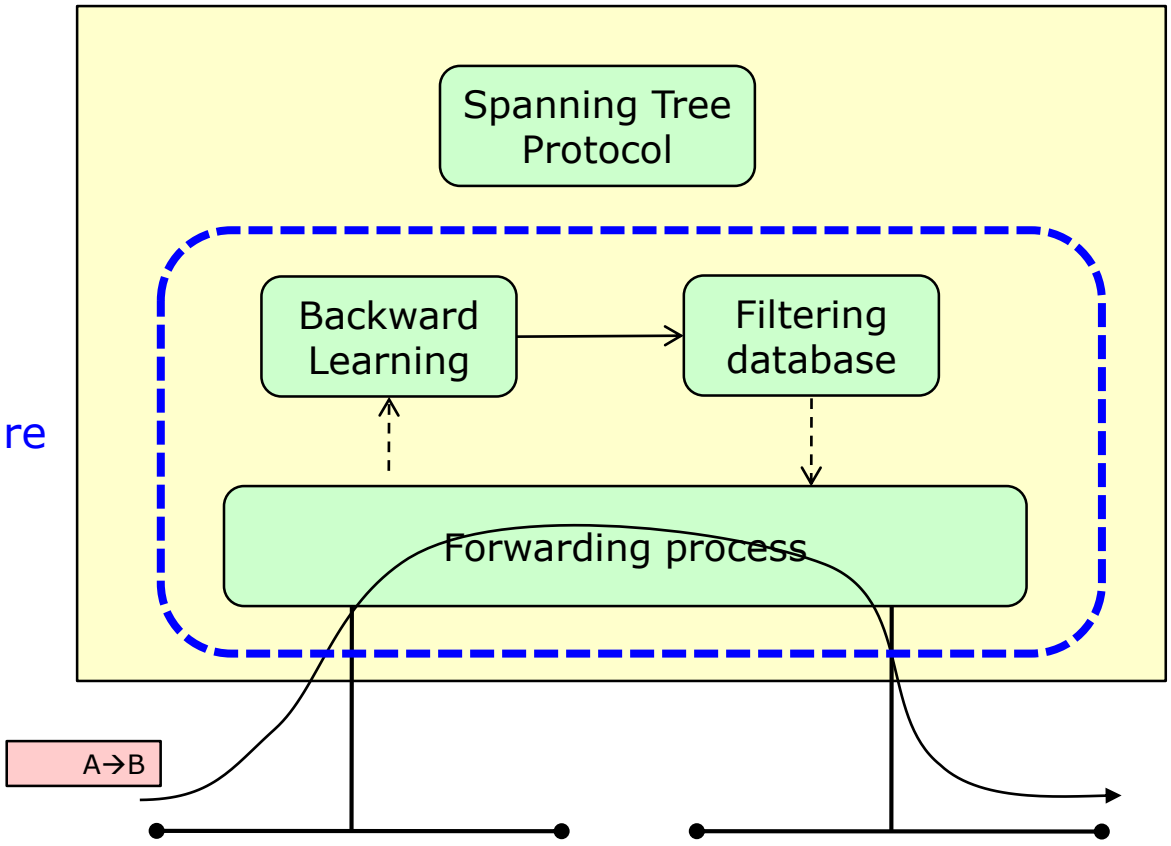
- Bridge
  - Originally 2 ports, then more
  - Software-based architecture
  - No longer used in real networks
  - Still some PC-based implementations
    - For research or some special purpose
  - WiFi access points are bridges

Bridges and switches (2)

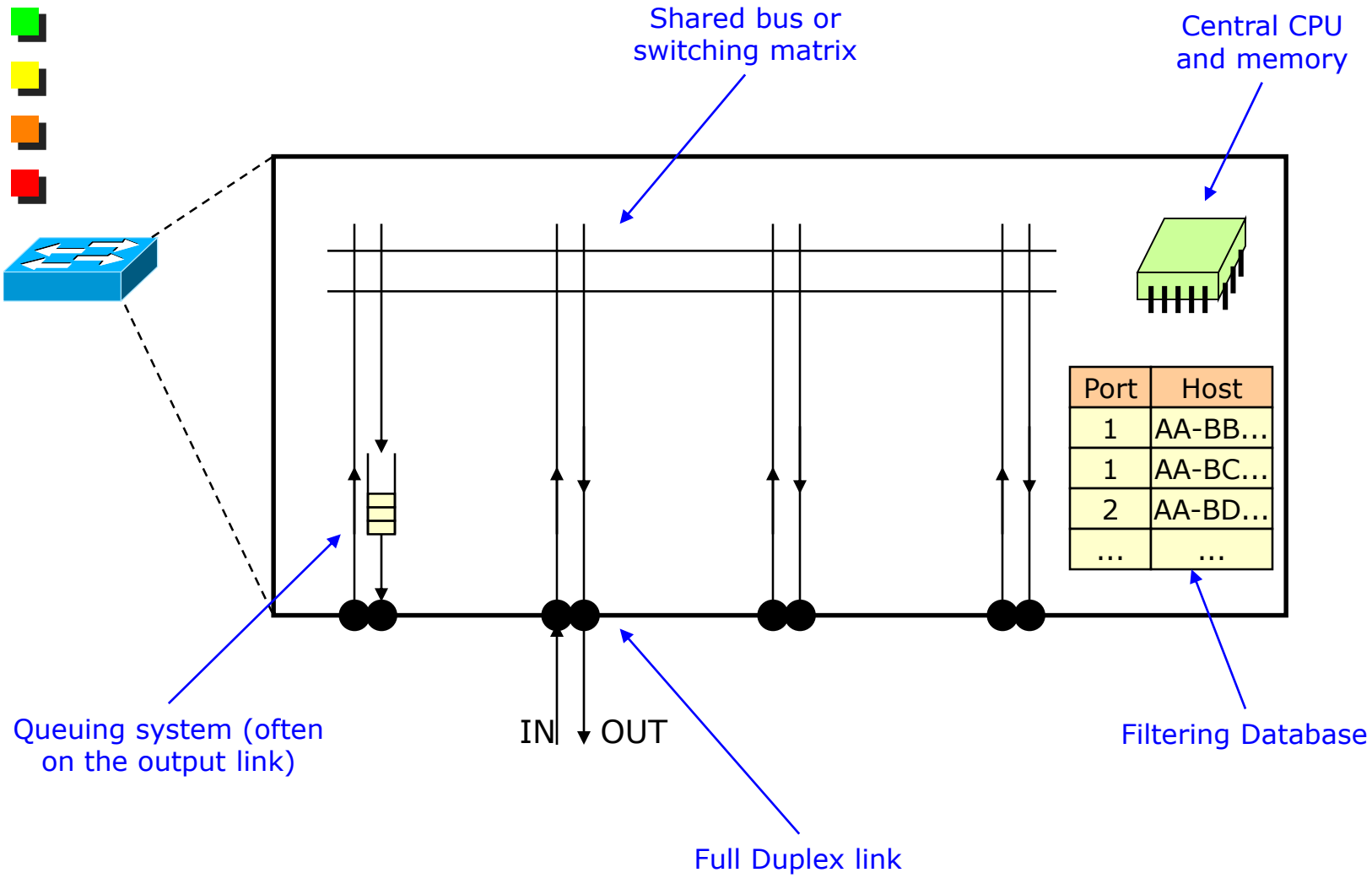
- Switch
  - Same device, different technology
  - Hardware based forwarding and learning
  - Lookup through CAMs (Content Addressable Memories)
  - Spanning Tree in software
    - Convergence time in several seconds, hence hardware implementation is useless
  - Can implement a "cut-through" forwarding technology
    - A frame can be forwarded on the target port immediately after receiving the Destination MAC
      - The destination port must be free at that time
    - Faster than "store and forward"
    - Requires all ports operating at the same speed

Switch architecture

Hardware



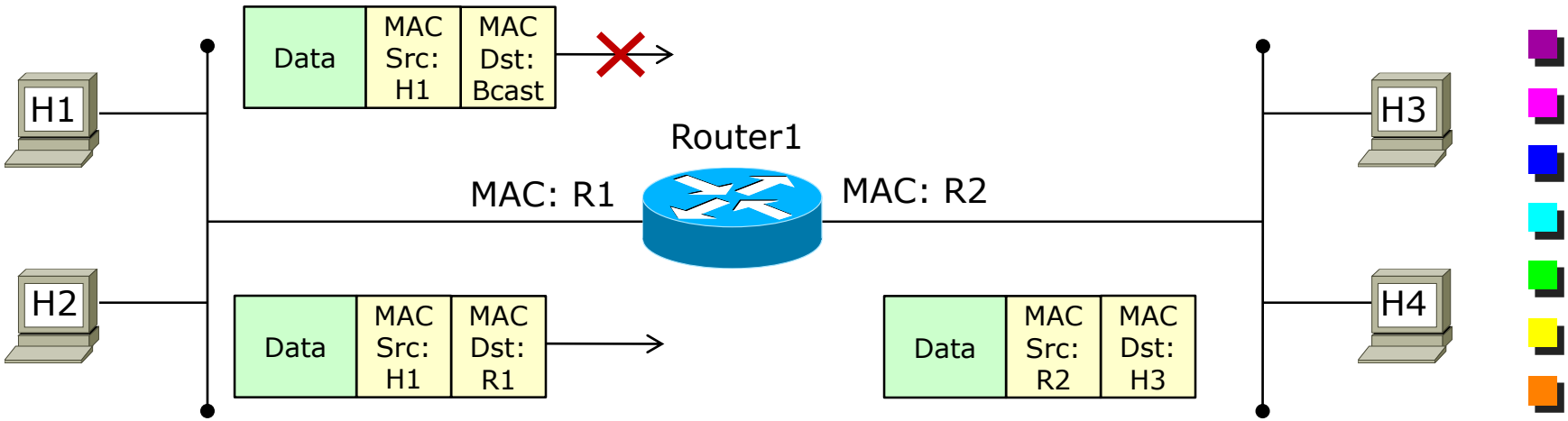
Switch internals



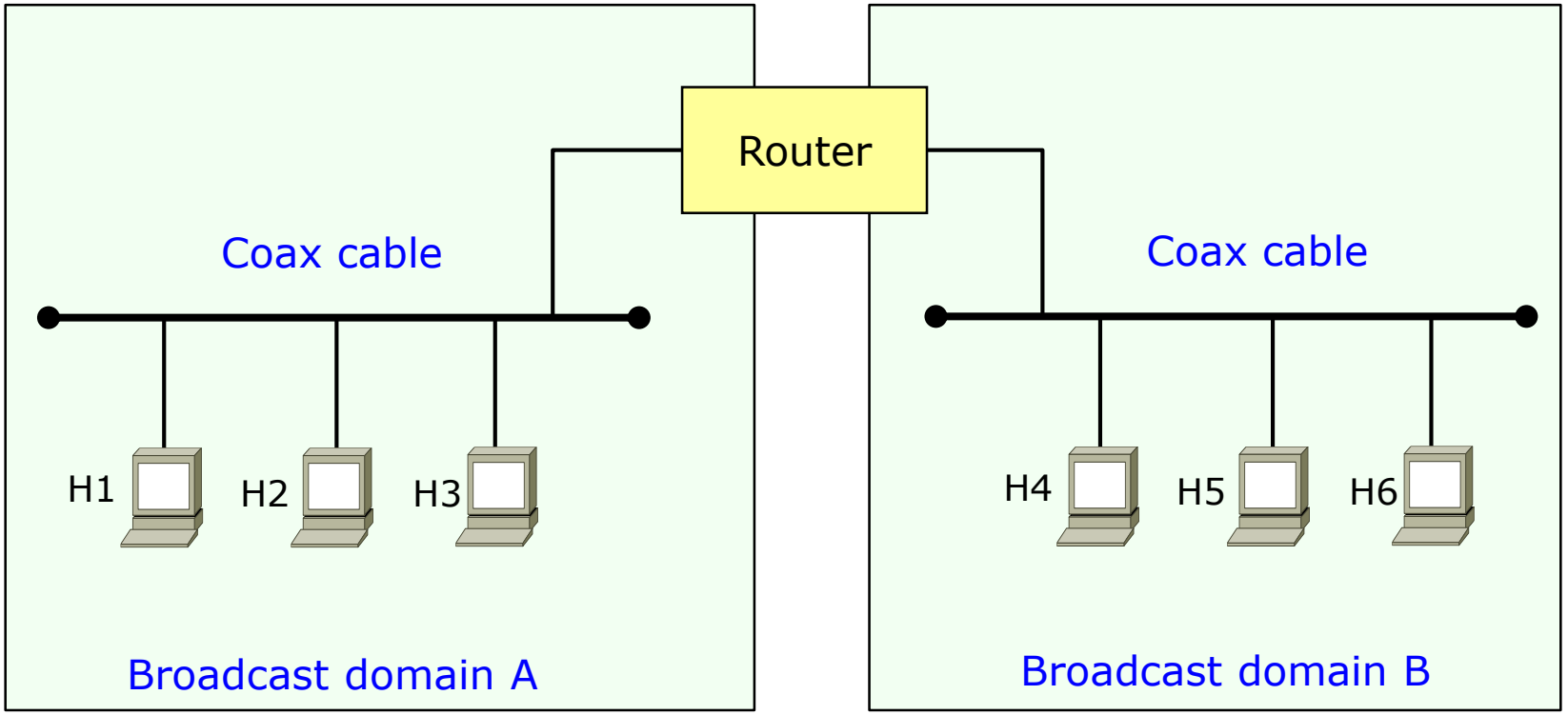
Routers

- Just a brief recap here for what concerts the L2 layer

- Routers are not transparent with respect to MAC addresses
- Routers separate **broadcast domains**



Routers and broadcast domains



Different IP networks on the two interfaces of the router

Conclusions

- Repeaters, Hubs and Bridges are historical
- Switches are really used
  - Most real network are now “pure switched networks”
  - Increased performance
  - Departure from some peculiar characteristics of a LAN
    - E.g., low error rate
    - No CSMA/CD, large diameters (in Ethernet)
- Components of a bridge/switch
  - Filtering Database
  - Backward Learning
  - Spanning Tree