Quality of Service in IEEE 802 LANs

Fulvio Risso

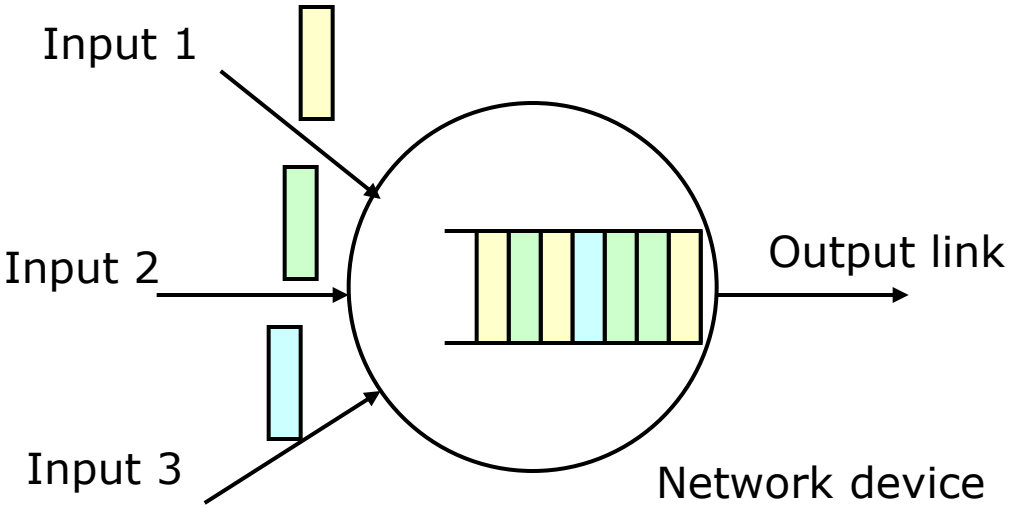
Politecnico di Torino

Based on chapter 8 of M. Baldi, P. Nicoletti, "Switched LAN", McGraw-Hill, 2002, ISBN 88-386-3426-2 and on an existing presentation of Mario Baldi and Piero Nicoletti

Copyright notice

The problem: QoS

- QoS in traffic forwarding is required when:
  - Limited amount of resources
  - The offered traffic exceeds the capacity of draining data ➔ congestion
- Many other QoS aspects not considered here
  - Resiliency of the network, etc.

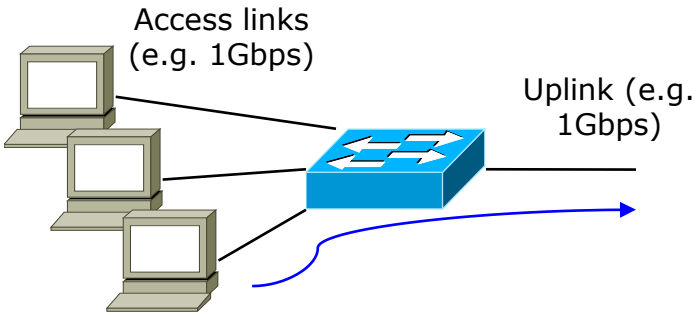


Do we need QoS in LANs?

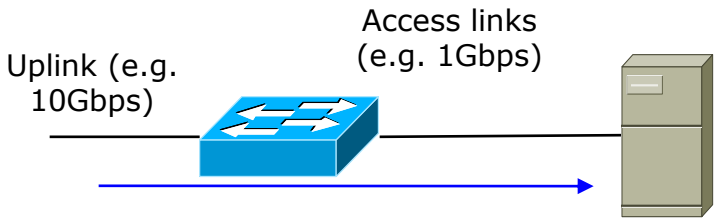
- Usually, LANs are over-provisioned
  - 30-40% of available bandwidth
  - Much cheaper to expand the network than enforce QoS
- Apparently, no congestion → no needs for QoS
  - So, apparently no problems

QoS in LANs (1)

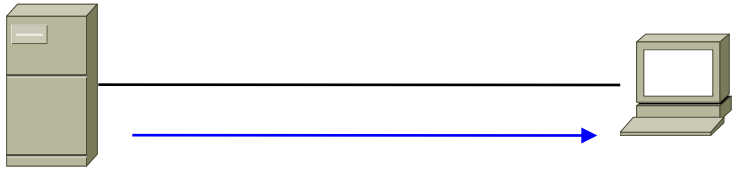
- Some possible scenarios in which we may have troubles



(1) Backbone not well dimensioned



(2) Data transfer from several hosts to a single server



(3) Data transfers from a fast host to a slow host

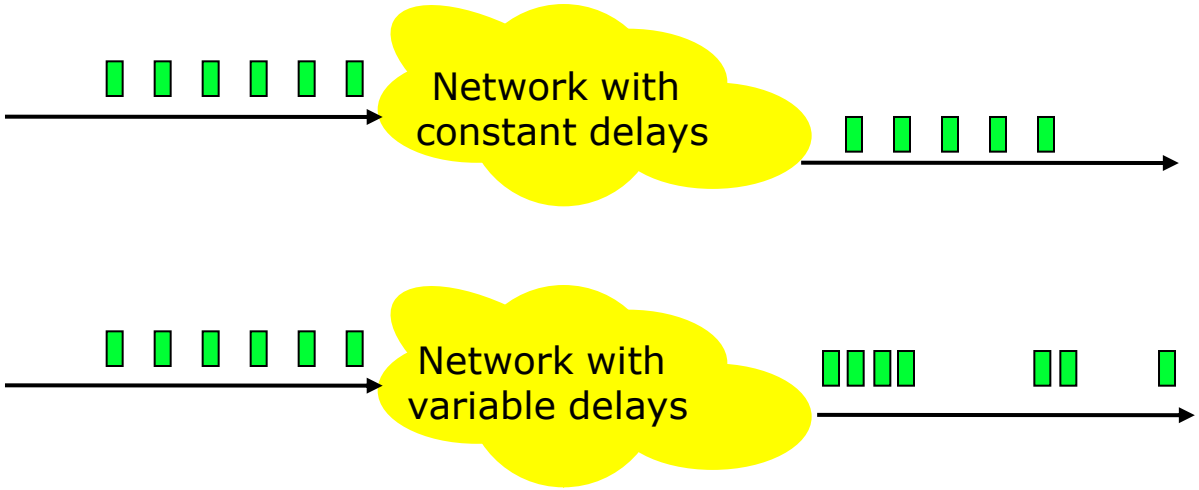
Fast/slow may refer to link speed (e.g. 1Gbps toward 100Mbps), CPU capacity, etc.

QoS in LANs (2)

- Example of congestions that may happen on LANs
  - Micro-congestions on uplinks
    - Emphasized when (uplink speed)/(access speed) ~ 1
    - Traffic from clients extremely bursty; may affect the traffic sent by other clients in short-term intervals
  - Temporary congestions on clients
    - Mainly present in the old days (long data transfers on slow links)
    - Multiplexing at packet level alleviates the problem
  - Persistent congestions on edge servers
    - Emphasized when (uplink speed)/(access speed) > 1

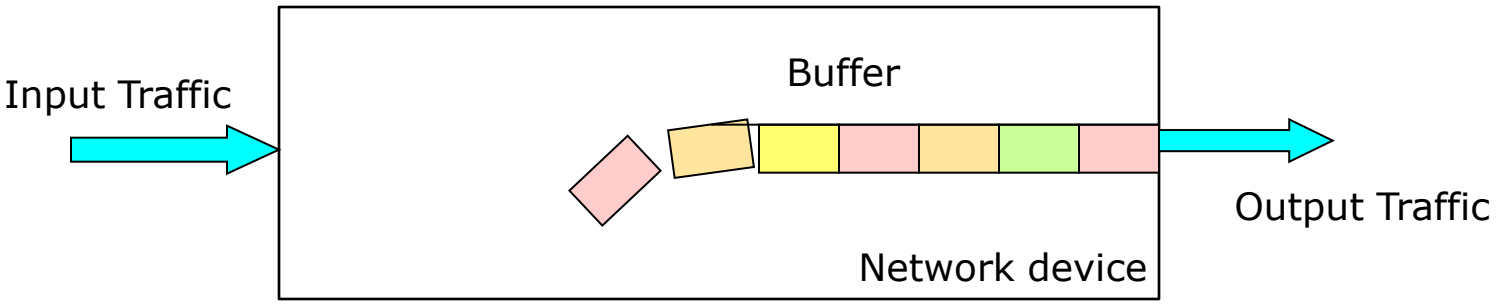
Possible effects when QoS is missing (1)

- Congestion creates delay variations
- Receiving timings influence the behavior of sensitive apps
  - Real-time (e.g., voice, telephony, music, video, videoconference)
  - Storage



Possible effects when QoS is missing (2)

- Congestion causes dropped packets
  - Small buffers in NIC of edge hosts
    - Isn't TCP Window enough to prevent congestion?
  - Small buffers in switches (e.g. some tens of KB) because
    - Fast memory is costly (here we need SRAM, not DRAM) and power hungry
    - We need to sell boxes at a very low price
    - People believe congestions do not happen and, in case some packets get dropped the transport-layer protocol will recover the error



Possible effects when QoS is missing (3)

- When packets are dropped
  - TCP can recover the error
    - Timeout and/or fast retransmit
    - Window halved (plus possible timeout)
    - Throughput declines
  - What about if we do not have TCP?
    - UDP (e.g. CIFS, NFS)
    - Other protocols (e.g. FCoE)
- Interesting, sometimes performance decrease after a network upgrade

LAN, QoS and Congestion

- Congestion depends on different factors
  - Link capacity
  - Network topology
  - Internal architecture of network devices and NIC cards
  - Operating systems (e.g. TCP flavor)
  - Applications and protocols used
- Congestion is possible in LAN environments
  - Usually micro-congestions, not massive congestions such as in WAN
- Do we need QoS mechanisms in LANs?
  - Yes, if we really care about QoS
  - Think carefully if you really need those features

Possible solutions to QoS Problems

- Frames dropped
  - Large buffers may reduce the problem
  - Better congestion control
- Variable delay
  - Queuing and scheduling algorithms
  - Select the next packet in the buffer in optimal way(?)
    - Sophisticated algorithms offer better control over delay
    - Normally we do not want complicated layer 2 switches
  - Limitation on number of frame contention (admission control)
    - Normally not used in layer 2 switches

Possible solutions in IEEE 802

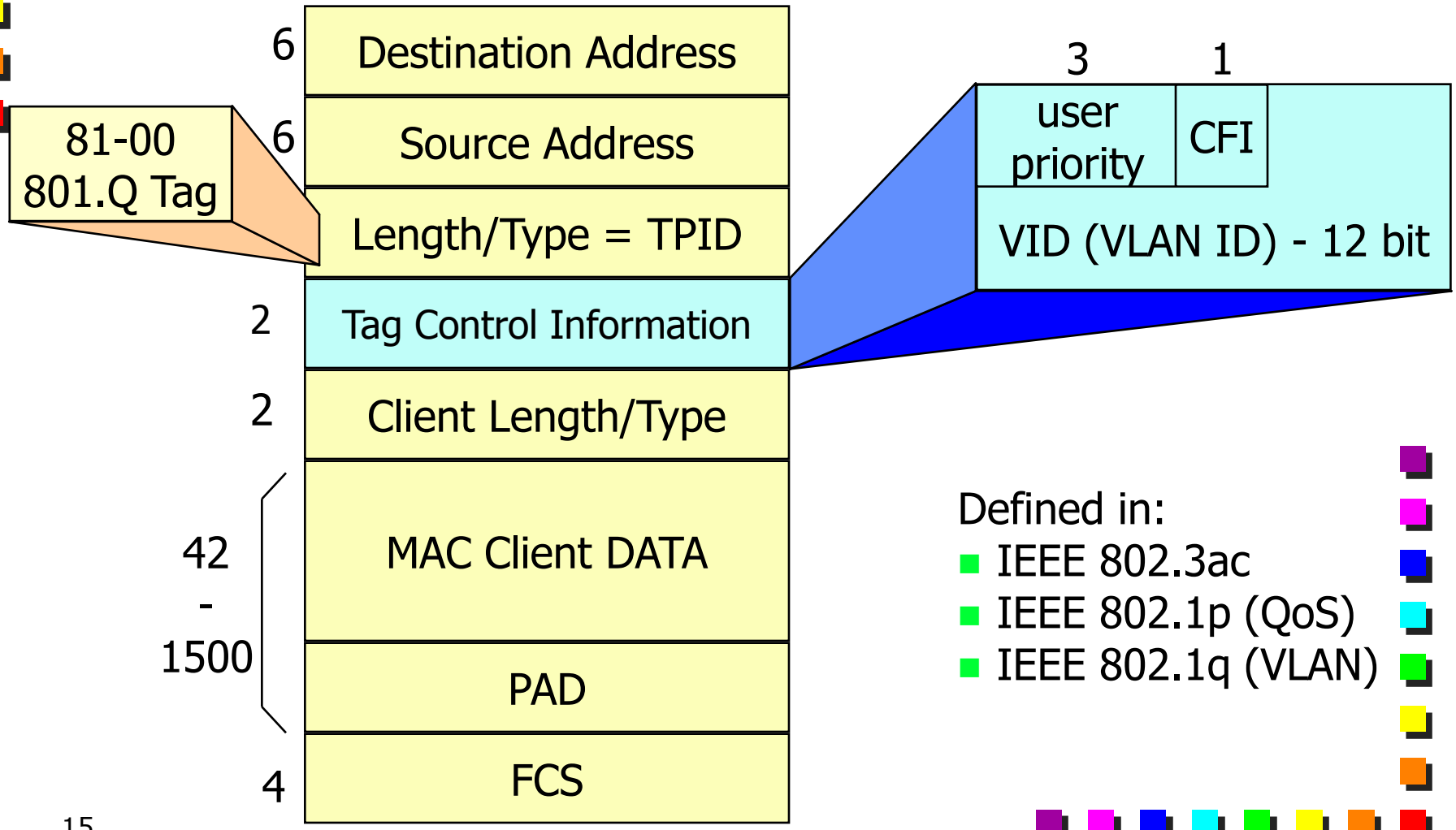
- Scheduling-based solution in 802.1p
  - Priority control
  - Valid on all IEEE 802 technologies
- Congestion control in 802.3x
  - Flow control
  - Valid only on IEEE 802.3

IEEE 802.1p

Characteristics of the 802.1p standard

- Very simple solutions that…
  - Does not solve the problem
    - Limits the troubles in your network
  - Does not implement QoS, just Priority
- Characteristics
  - 8 priority levels
    - It does not imply any hierarchical relationship among them, even if the word *priority* is used
  - Priority level in the VLAN portion, encoded with 3 bits
  - Different (logical) queues for different services
    - At least 2, at most 8
    - Usually implemented in hardware

Tag coding: IEEE 802.1p e 802.1q



Defined in:

- IEEE 802.3ac
- IEEE 802.1p (QoS)
- IEEE 802.1q (VLAN)

Priority/traffic association proposed

User Priority	Description
0	Best Effort (BE)
1	Background (BA)
2	(not defined)
3	Excellent Effort (EE)
4	Controlled Load (CL)
5	Video, < 100ms latency and jitter (VI)
6	Voice, < 10ms latency and jitter (VO)
7	Network Control (NC)

IEEE 802.1p recommended aggregation

Code No.	Kind of traffic							
1	BE							
2	BE				VO			
3	BE				CL		VO	
4	BK		BE		CL		VO	
5	BK		BE		CL	VI	VO	
6	BK		BE	EE	CL	VI	VO	
7	BK		BE	EE	CL	VI	VO	NC
8	BK	----	BE	EE	CL	VI	VO	NC

Scheduling

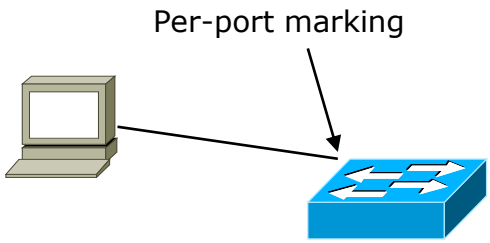
- IEEE 802.1p recommends fixed priority as traffic/queue association
- Possibility to use variable priority scheduling algorithms
  - Round robin, weighted round robin, weighted fair queuing
- Different range equipments can offer different algorithms
- Configuration commands let to
  - Assign priority value (user priority) to queue
  - Set the scheduling algorithm

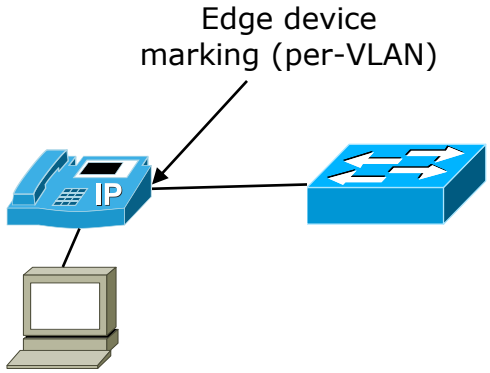
Who will set the Priority Level in the packet? (1)

- Network card of the host
  - Several problems
    - What about software configuration?
    - We need VLAN trunking on the Access side
- Access Switch
  - Several problems
    - We have L2; we should need L3-L4 for proper identification
    - No knowledge about application that generated the packet
    - MAC-based: who will keep the MAC database updated?
  - Usually done based on input port
    - Not very clever, though

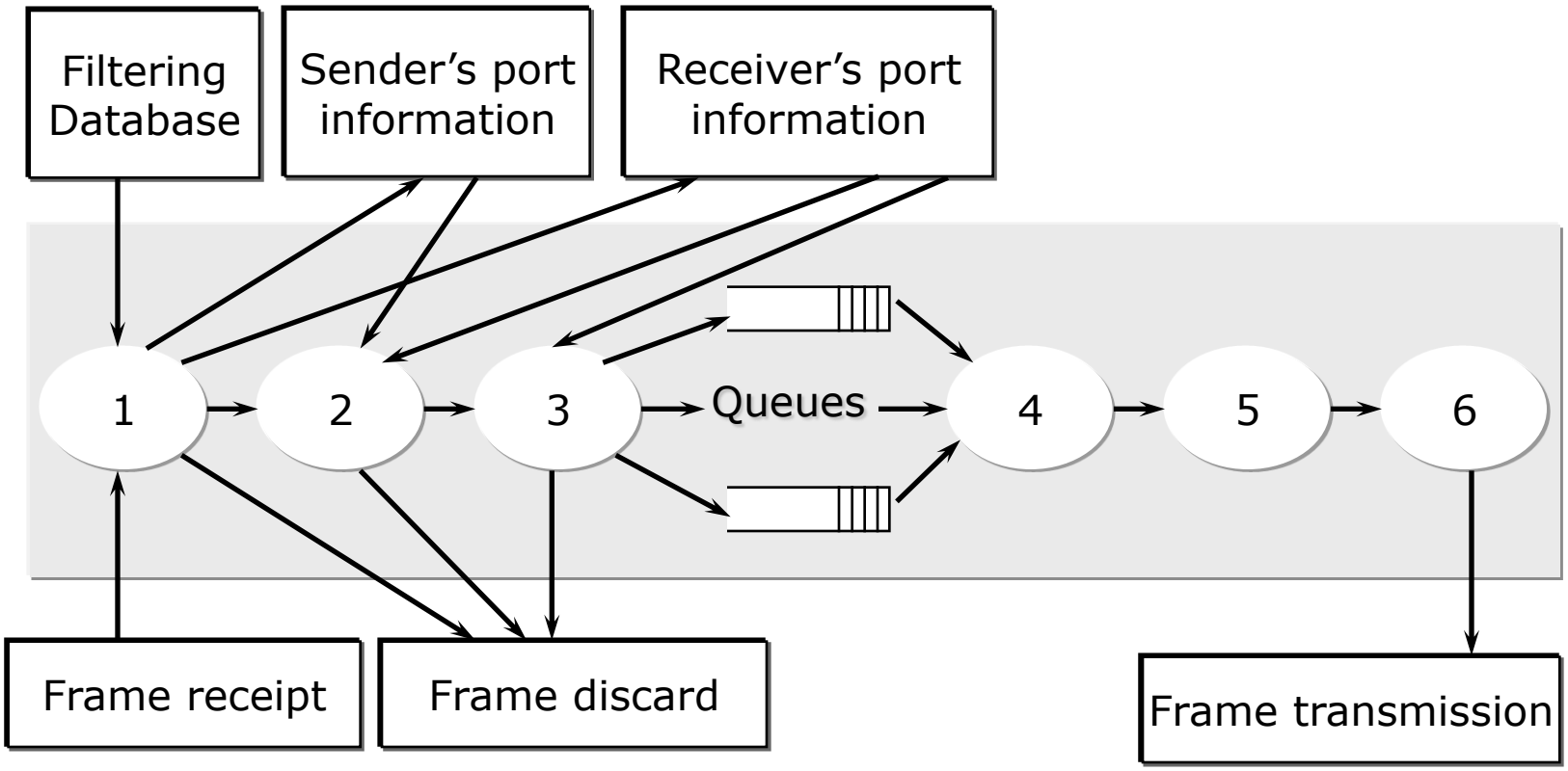
Who will set the Priority Level in the packet? (2)

- Some examples of deployment in nowadays networks
  - Switch-based marking (per port)
    - Problematic in case multiple users are connected through the same access link
  - Edge-device marking (per VLAN)
    - Possible when the edge device is trusted





IEEE 802.1p switch functional architecture



1 Filtering Frames
2 Enforcing topology restriction (STP)
3 Queueing Frames

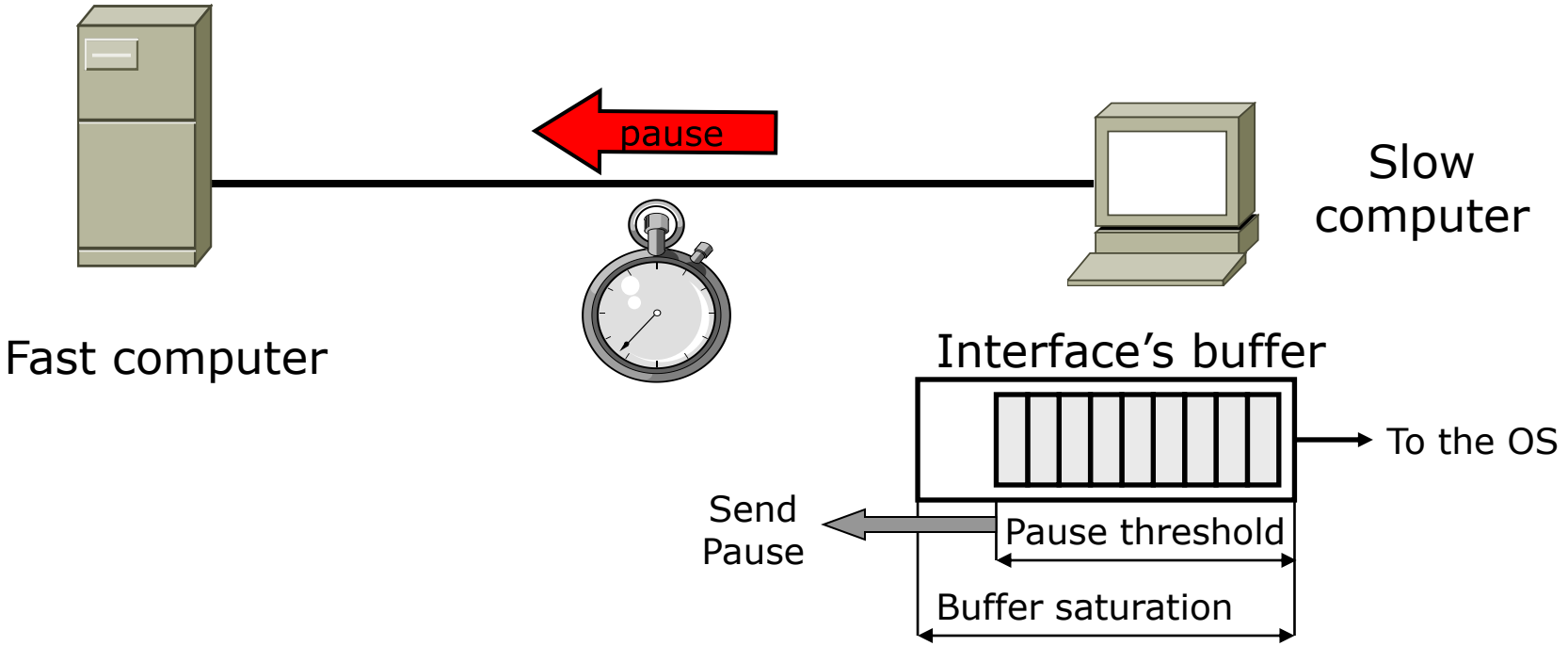
4 Selecting frames for transmission
5 Mapping priority
6 Recalculating FCS

IEEE 802.3x

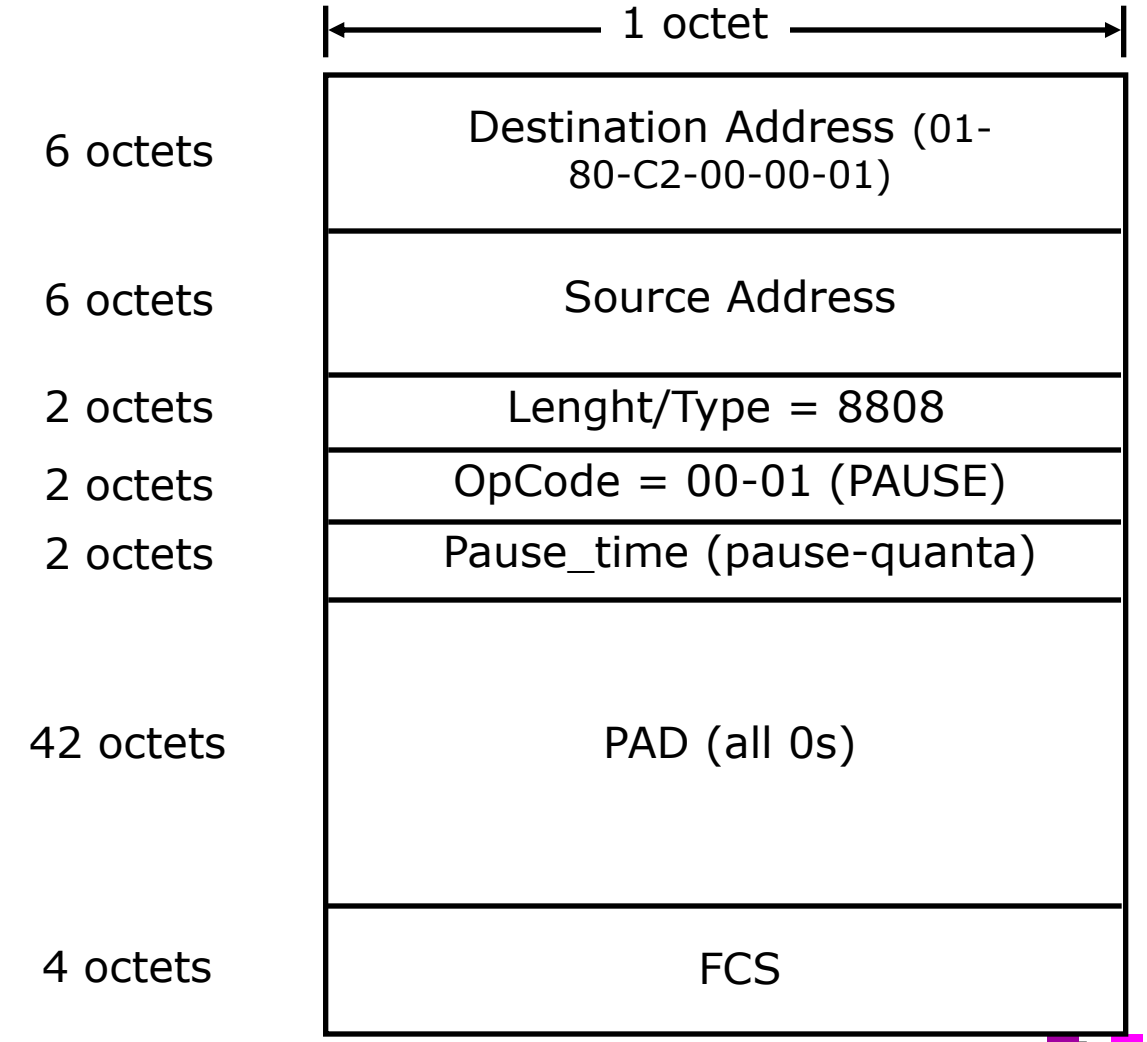
Characteristics of the 802.3x standard

- Implements a flow control at the Ethernet level
  - In addition to the Flow control existing TCP level
- Special PAUSE packet
  - Sent in multicast on a physical link
  - The receiver must stop data transmission on **that** link for a given amount of time
    - The amount of "idle" time is specified within the packet
    - This time can be updated (extended or reduced) by another PAUSE packet
    - In case of a switch, the other links still receive/send data
- 802.3x defines also the Full Duplex mode
  - Not presented in these slides

IEEE 802.3x Flow Control: example



PAUSE packet



Pause time

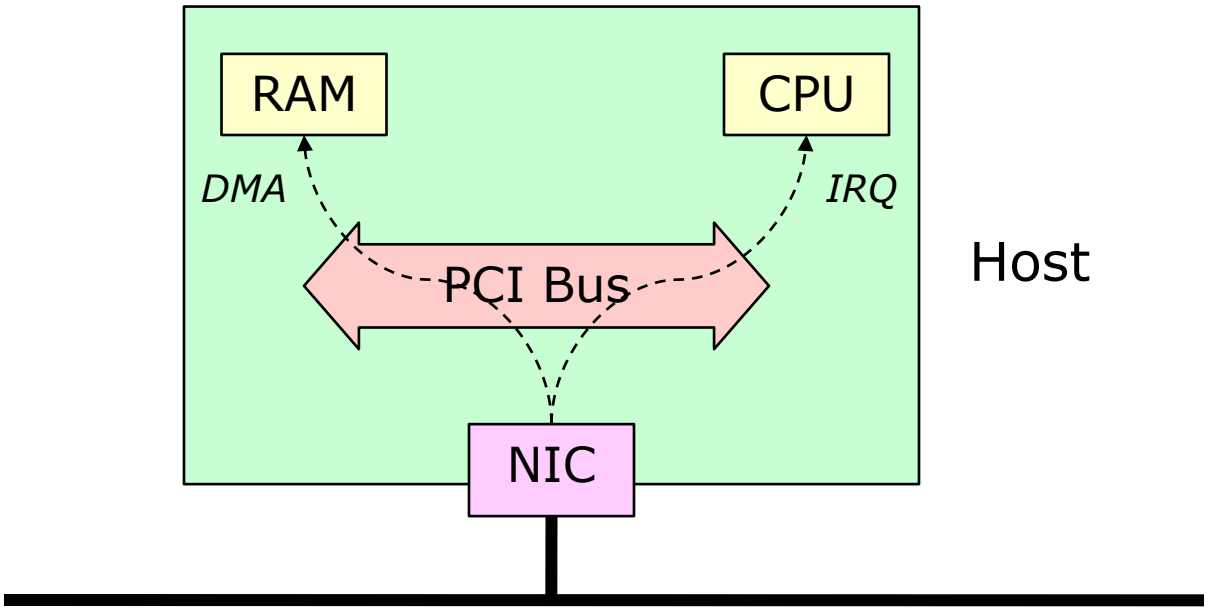
- Defined in *number of pause-quanta*
  - Values from 0 to 65535
  - A single *pause-quanta*: 512 bit time
  - Transported in the “pause_time” field
- Total pause time (in bit times)
  - <=100 Mb/s
    - T-Pause = pause-quanta * 512
  - >100 Mb/s
    - T-Pause = pause-quanta * 512 * 2

IEEE 802.3x: flow control modes

- Two flow control mechanisms:
  - asimmetric mode
    - only one equipment sends pause packet, the other just receives the packet and stop transmitting
  - simmetric mode
    - both equipments at link's edge can transmit and receive the pause packet
- Configurable at each edge device
  - Configuration should be coherent on both sides
  - An auto-negotiation phase will determine the actual configuration on each link

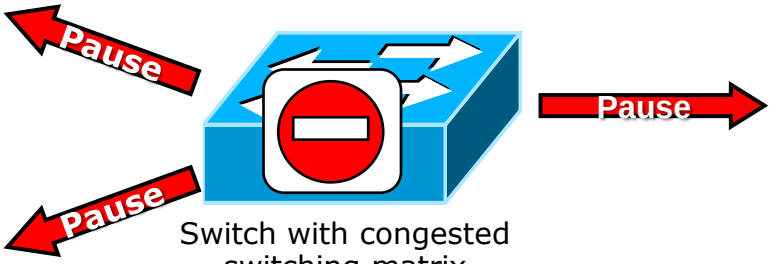
Sending the PAUSE packet (1)

- Simple for an host
  - OS-independent (the hardware does it all)
  - Although the NIC/OS interaction may prevent the generation of PAUSE packets
    - Livelock in the OS kernel



Sending the PAUSE packet (2)

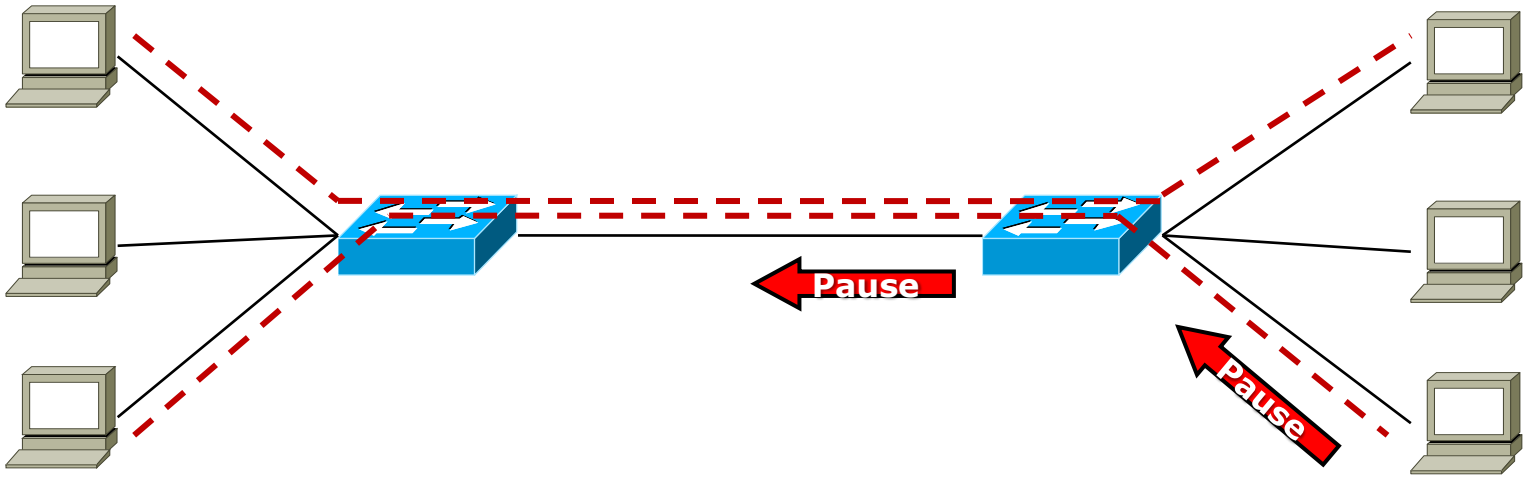
- May be problematic for a switch
  - Key question: who is the responsible for a queue that starts growing?
  - Input-buffered switch: simple
  - Output-buffered switch: all IN ports that have packets in the output queue?
  - Switching matrix congested: all IN ports?
- Several commercial switches accept a PAUSE packet (and block transmission on that port), but cannot send it
  - Cannot be modified by configuration



Switch with congested switching matrix

Effects of Flow Control in the network

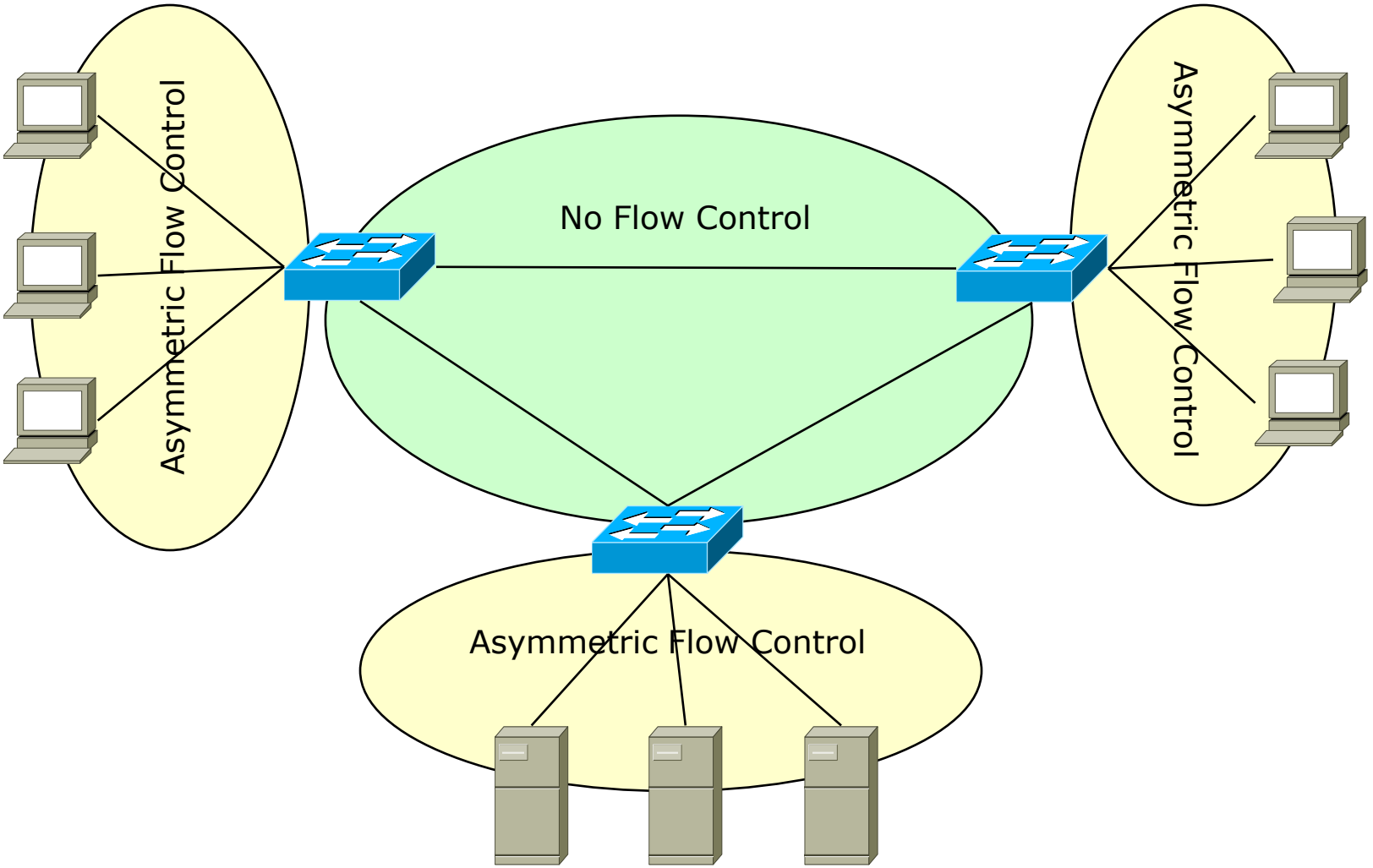
- The PAUSE packet is *per link*
- May be fine in end-hosts, may be problematic in backbone networks
  - The network may experiment complete blocks for some time
  - The effect of a bunch of slow stations in the network



Realistic approach in 802.3x deployment (1)

- Asymmetric flow control on the access network
  - Hosts can pause the network, but not viceversa
- No flow control in the backbone
- Characteristics
  - We can cope with temporary congestions on input buffers of station's network interfaces
  - Does not block the entire network in case of a bunch of slow stations
  - Does not solve entirely the problem, but it may be useful anyway

Realistic approach in 802.3x deployment (2)



Conclusions (1)

- QoS is potentially a nice feature
  - QoS issues are possible even in LAN
    - Small buffers, mismatch in link speed (access vs uplink), micro-congestions, server bandwidth
  - Good for Voice, Storage
- QoS is just one of the problems to consider
  - Resiliency

Conclusions (2)

- Problems
  - 802.1p
    - Interoperability between different vendors
    - Lack of flexibility for marking (mainly per-port)
  - 802.3x
    - Realistic approach: edge only
    - Full-features approach
      - Who is the responsible for the congestion
      - Blocks in the backbone network
- Do we really need QoS?
  - Or, can we just stay with some congestions in the network?