



Switch architecture

Mario Baldi

Politecnico di Torino

<http://staff.polito.it/mario.baldi>

Pietro Nicoletti

Studio Reti

<http://www.studioreti.it>

Based on chapter 8 of:

M. Baldi, P. Nicoletti, "Switched LAN", McGraw-Hill, 2002, ISBN 88-386-3426-2

Copyright Notice

This set of transparencies, hereinafter referred to as slides, is protected by copyright laws and provisions of International Treaties. The title and copyright regarding the slides (including, but not limited to, each and every image, photography, animation, video, audio, music and text) are property of the authors specified on page 1.

The slides may be reproduced and used freely by research institutes, schools and Universities for non-profit, institutional purposes. In such cases, no authorization is requested.

Any total or partial use or reproduction (including, but not limited to, reproduction on magnetic media, computer networks, and printed reproduction) is forbidden, unless explicitly authorized by the authors by means of written license.

Information included in these slides is deemed as accurate at the date of publication. Such information is supplied for merely educational purposes and may not be used in designing systems, products, networks, etc. In any case, these slides are subject to changes without any previous notice. The authors do not assume any responsibility for the contents of these slides (including, but not limited to, accuracy, completeness, enforceability, updated-ness of information hereinafter provided).

In any case, accordance with information hereinafter included must not be declared.

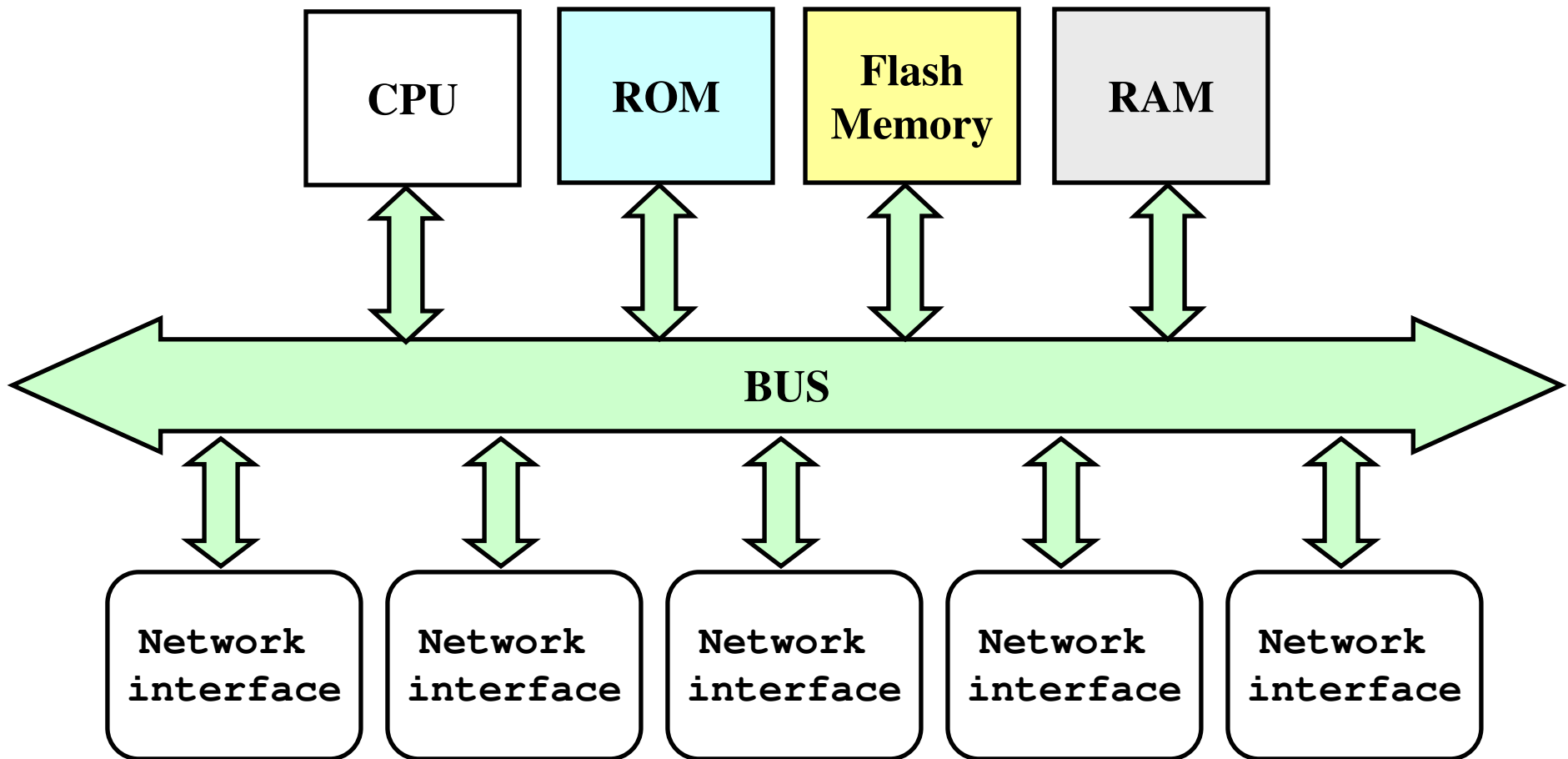
In any case, this copyright notice must never be removed and must be reported even in partial uses.

Bridge or Switch?

- This two terms are often used indifferently
 - The devices are *functionally* the same
- *Bridge*: Layer 2 internetworking device
 - Transfers MAC frames among separate LANs
- *Switch*: commercial term, introduced to emphasize the speed of a device
 - Same functionalities
 - Higher number of ports
 - Higher aggregate throughput
 - Usually hardware (ASIC)-based forwarding

Traditional bridge architecture

Low scalability → low number of ports



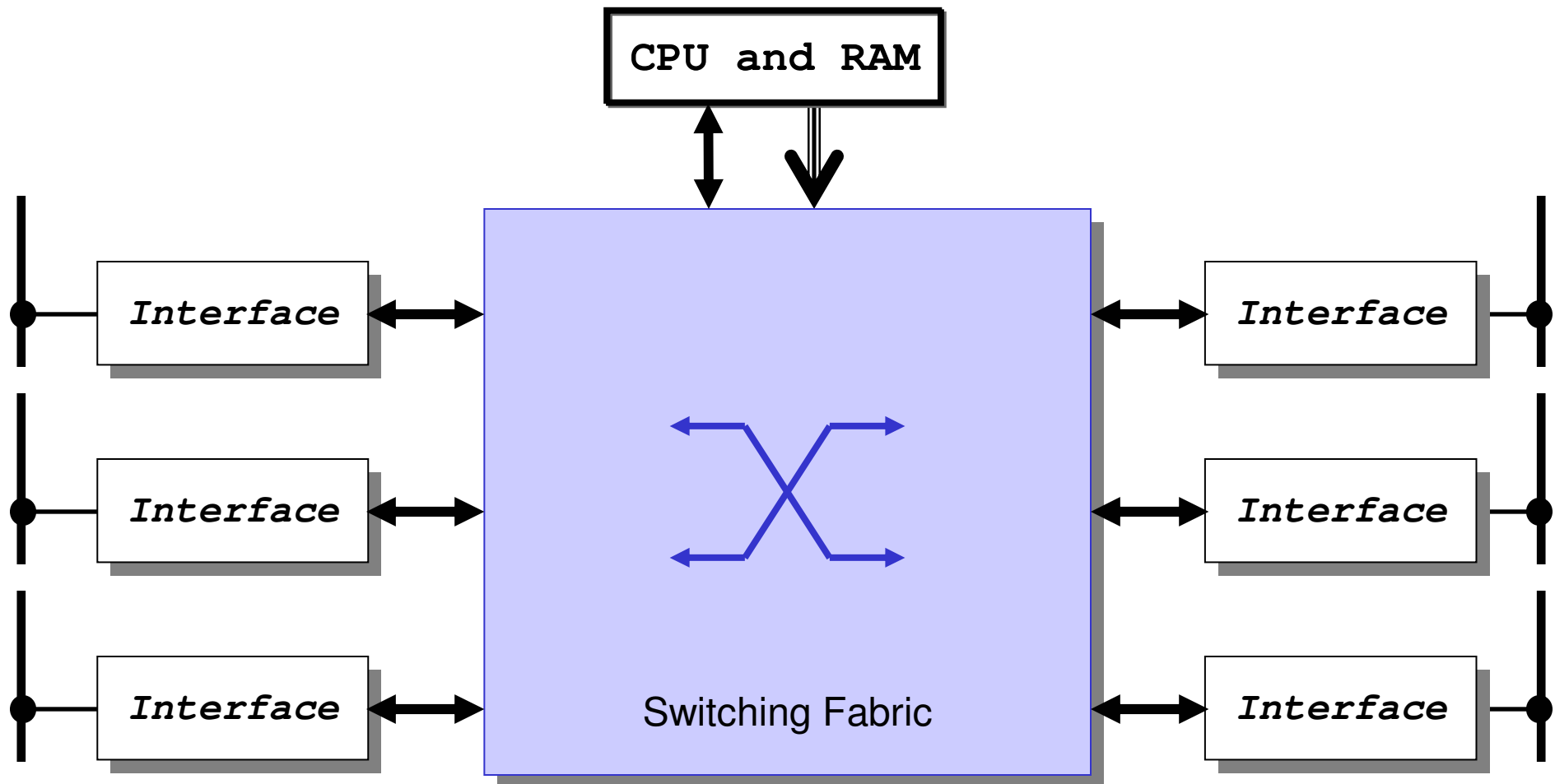
Scalability limitations

- An high number of ports is required to realize switched LANs
 - Possibly one port for each network station
- Bridge scalability limited by these bottlenecks:
 - Processor
 - Memory
 - Bus
- Increasing the number of ports (or their speed) by N times requires the same increase in
 - Elaboration capabilities of the processor
 - Access speed of the memory that keeps routing information
 - Bus transfer capacity

How to overcome these limitations

- Distribution of functionalities that are usually centralized
 - Elaboration
 - Presence of many processors
 - Switching
 - Switching fabric
 - Many simultaneous paths, between inputs and outputs
 - Space switching instead/above time switching (as in the bus)
- Use of a specialized hardware
 - “Ad hoc” design
 - Application Specific Integrated Circuit (ASIC)
 - Less flexible, but optimized (faster)

Architecture of a switch



Functionalities distribution

- Central processor: control
 - Spanning tree protocol execution
 - Switching fabric reconfiguration
 - Management
- Interface processors
 - Packet forwarding
 - Packet parsing
 - Routing decision
 - Possible modification of the packet
- Scalability: each interface processor elaborates only packages received on that interface
 - If the number of interfaces grows, the number of processors must be increased

Issues

- Information update and forwarding
 - The Filtering Database has to be accessed by the interface processors
 - Complex sharing and synchronization techniques
 - Centralized table
 - Local copies (cache)
- Coordination between interface processors and the main processor
 - Control policies of the switching fabric
- Complex (and non-standard) procedures developed by vendors during years
 - Different background

Switching fabric

- Bus
- Crossbar
- Multistage network

Non blocking Switching fabric

It can send frames received from any interface on the correct outgoing interface, as long as it is not busy with another transfer

Bus

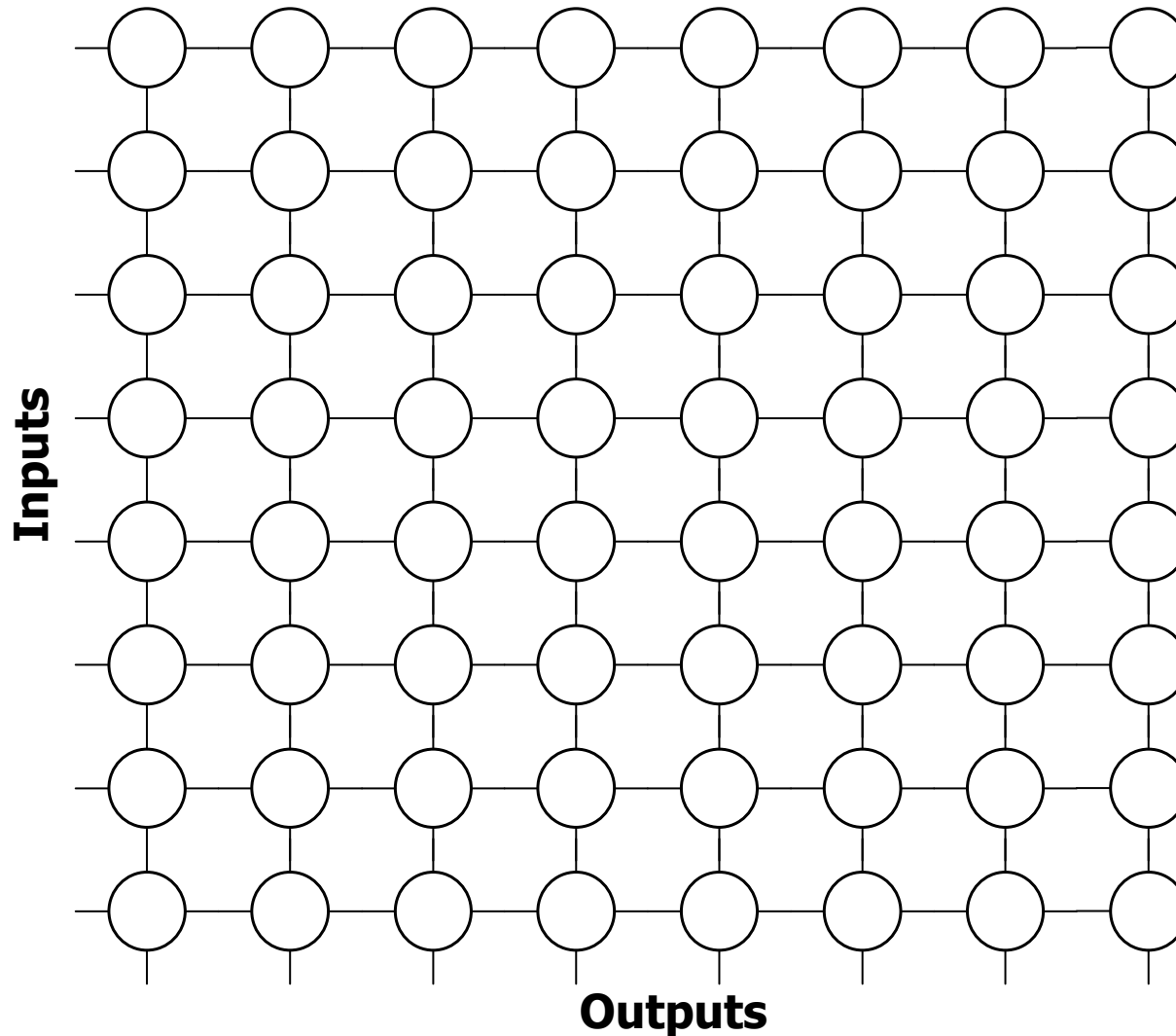
- Intrinsic blocking → *speedup*
 - Capacity equal to the aggregate capacities of interfaces
 - Example: 64 interfaces at 1 Gb/s → bus at 64 Gb/s
- Limited scalability: if the number of interfaces grows
 - Bus capacity must be increased
 - Bus length grows
 - Increased sensitivity to electromagnetic interferences
- Solution: increase parallelism
 - Speed transmission on each line stays limited
 - The complexity of connectors is increased
 - Interference issues among lines
 - Increases the granularity of transfers
 - Possible inefficiencies

Crossbar

- Non-blocking Switching fabric by definition
 - It can connect anytime each input to each free output
- It distributes packets that must be transferred on different paths
 - Space switching
 - Not necessary speedup
 - Transfer capability from input to output equal to the capacity of each interface
 - Aggregate transfer capacity equal to the aggregate capacity of interfaces
- Logical vision as an network based on elementary commutators



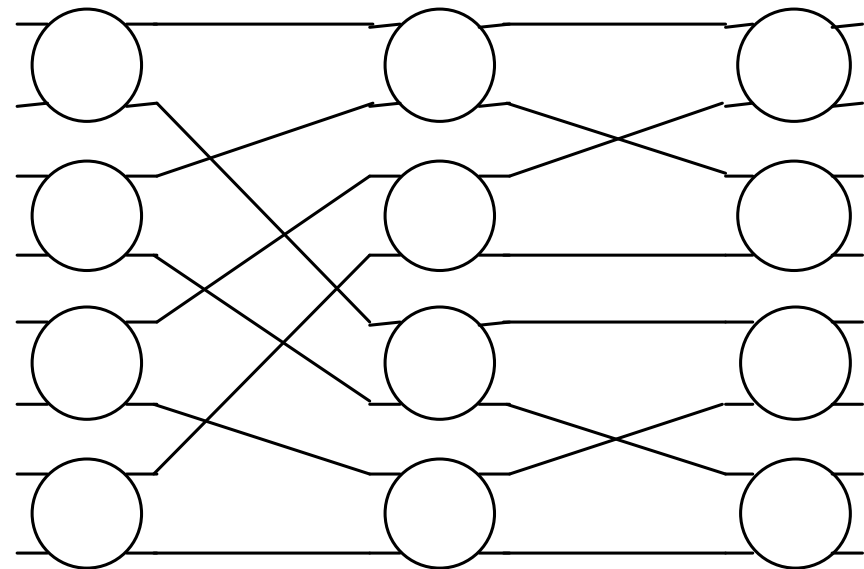
Crossbar: conceptual view



Quadratic complexity:
10 times more interfaces
→ 100 times more switches

Multistage networks

- Better scalability if compared to crossbar
- Clos
 - Non blocking
- Banyan
 - Minimum number of: $o(N \log N)$
 - Highest scalability
 - Blocking



Non-blocking switching fabrics : is that all?

- Frames received from different inputs cannot be moved simultaneously to the same output
- One is sent, the remaining ones are stored at the input stage

Q: How to avoid input frames memorization?

A: By increasing the transfer speed:

- Each frame can be moved from an input to an output during the time needed to receive a frame
- The cumulative transferring capability is equal to the aggregate capability of the interfaces

Scalability.?

Speedup

- Overall transfer capability is higher than the capabilities fo the interfaces

- In the worst case:

Non blocking switching fabric
 +
 Speedup equal to the number of inputs
 =
 no storage of incoming packets

- Ideally

Non blocking switching fabric
 +
 Speedup equal to 2
 =
 no inputs congestion

- Assumption on traffic distribution: are they realistic?
 - Complex algorithms to handle input queues (input queuing)
- Speedup has an impact on the interface circuitry
 - For example, the memory on the output board

The right (?) way is a trade-off

- High speedup
 - Memorization at the output stage (output buffering) → lower complexity
 - Lower switching fabric scalability
- Low speedup
 - Memorization at the input stage (input buffering) → greater complexity
 - Complex algorithms to handle queues (scheduling)
 - Higher switching fabric scalability

Trade-off: the solution

- Limited Speedup – often lower than 2
- Buffers both in the input and in the output stage (combined I/O buffering)
- Suboptimal handling of queues (but it can be implemented)

Does everybody follow this way?

Goals

- Minimize complexity
- Maximize scalability
- Ensure decent performances

Solution

- Low speedup (it can be even 1)
- Queues only at the output stage or very simple scheduling at the input stage
- Switching fabric blocking if it is necessary

Result

- Satisfying performances with real traffic profiles
 - Low contention probability for the same output
 - Low average load on interfaces

Non blocking switching fabric + speedup: is that all?

No, if the purpose is to ensure quality of service!

- The elimination of the contention for the output interface does not eliminate the contention for transmission
 - It is not possible to send more than a frame at a time
 - One frame is sent, the others are stored
- The resulting service
 - Depends on the number of competing frames
 - Depends on the *instantaneous* traffic profile
- To improve interface speed does not solve the global problem
 - To increase receiving speed!!!!

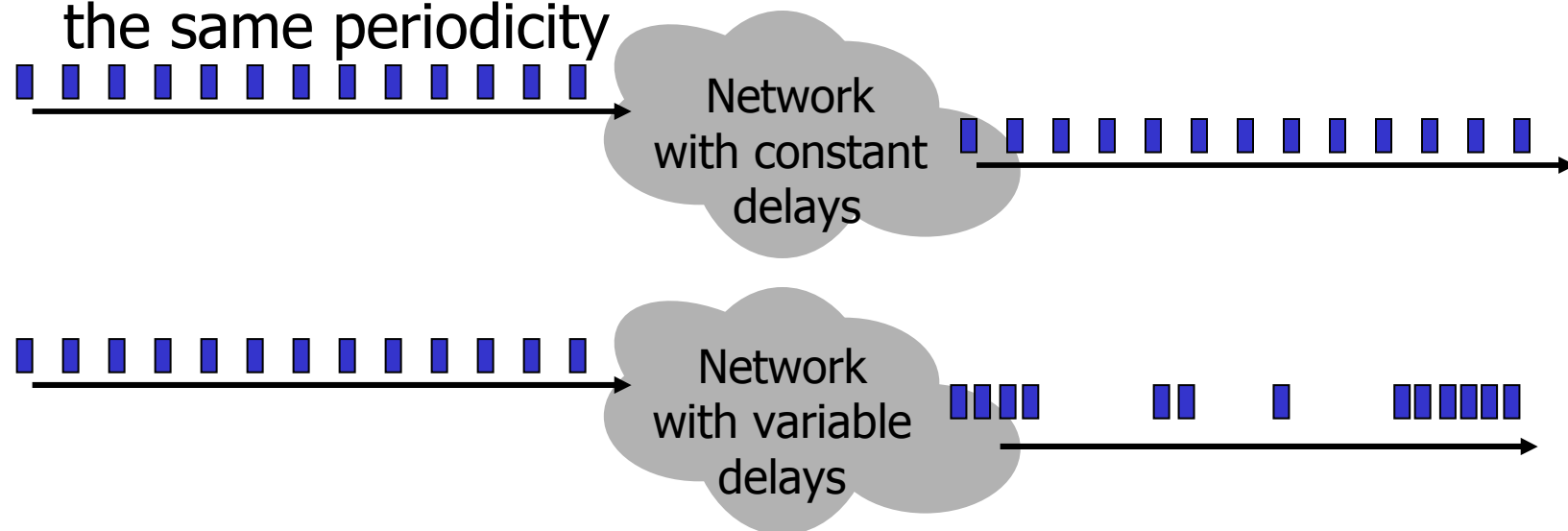
Consequences and remedies

- Discard frames
 - If the buffers are big enough they can mitigate the issue
- Variable delays
 - Differentiated queuing and scheduling algorithms
 - Choose the next packet to send from the buffer in an optimal way (?)
 - More complex algorithms improve delay control
 - Normally complex layer 2 switches are not desired
 - Limitation on the number of competing frames (admission control)
 - Usually it is not used by layer 2 switches

Real-time applications

Reception timings affect the behaviour

- Voice, telephony, music, video, videoconference
- Getting more and more used on local and non-local networks
- The original signal is sampled at regular intervals
- To get good quality, samples must be reproduced using the same periodicity



Delay control

- Replay buffer
 - To the destination:
 - Does not require modifications to network devices
 - Can be implemented by the application itself
 - Increases delays: not ideal for interactive applications
- Advanced queue handling
 - Solution at the root of the problem
 - Differentiated queues
 - Complex scheduling algorithms
 - Traffic control
 - Network engineering
 - Traffic engineering
 - Resources reservation (admission control)

IEEE 802.1p