

Real-Time Systems

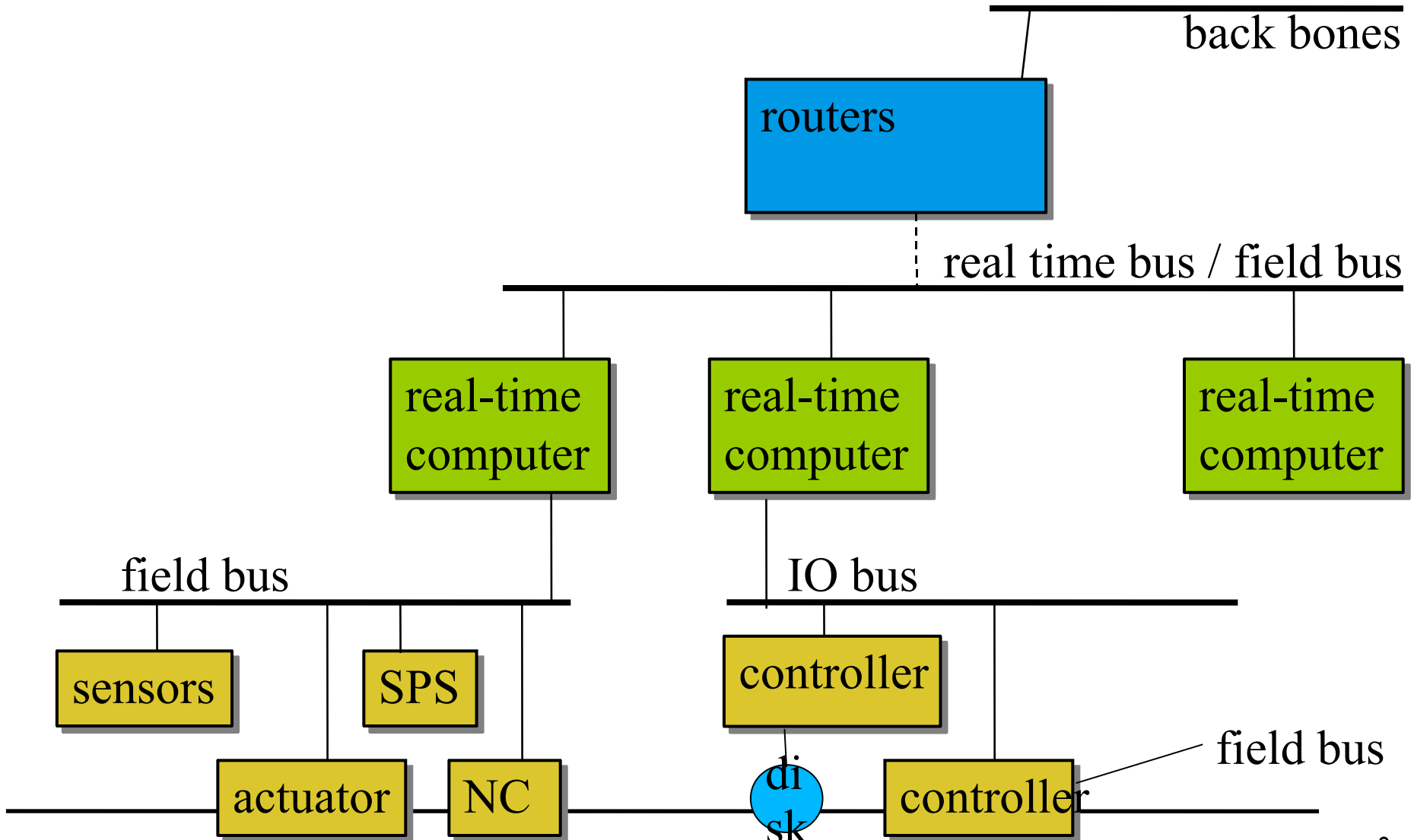
Real-Time Communication

Hermann Härtig
(following Kopetz, Liu, Schönberg, Löser)

Contents

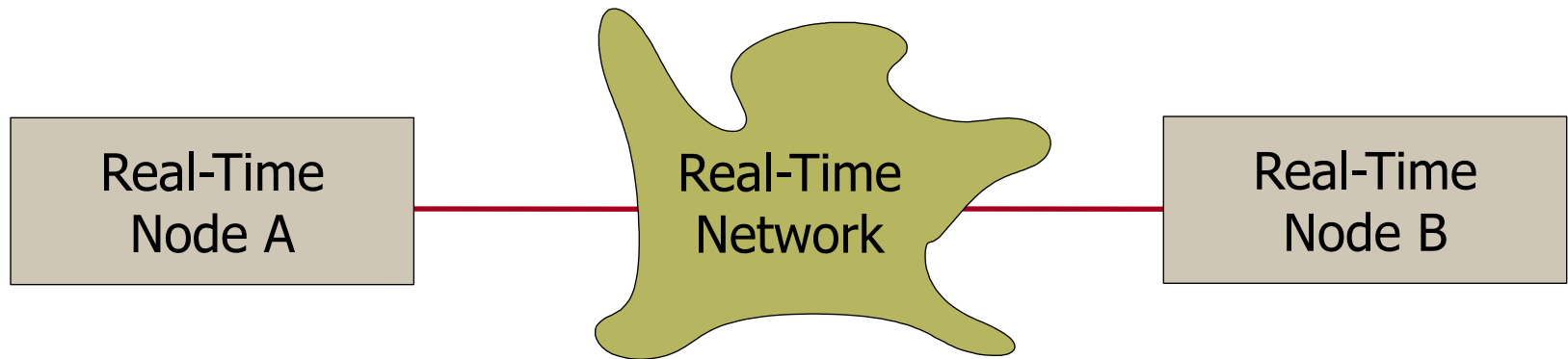
- Overview
- IO Busses: PCI
- Networks as schedulable resources:
Priority / Time-Driven / Weighted Round Robin
 - Priority – Based Field: CAN/Token Ring
 - Weighted Round Robin in Wide Area
 - Ethernet in RT:
 - Shaping
 - TT-Ethernet
- RT-Corba (very short)

General Scenario



Purposes of a Real-Time Network

- Deliver communication services to the requesting nodes reliably, securely, efficiently and **timely**
 - lower bound for bandwidth
 - upper bound for latency and jitter

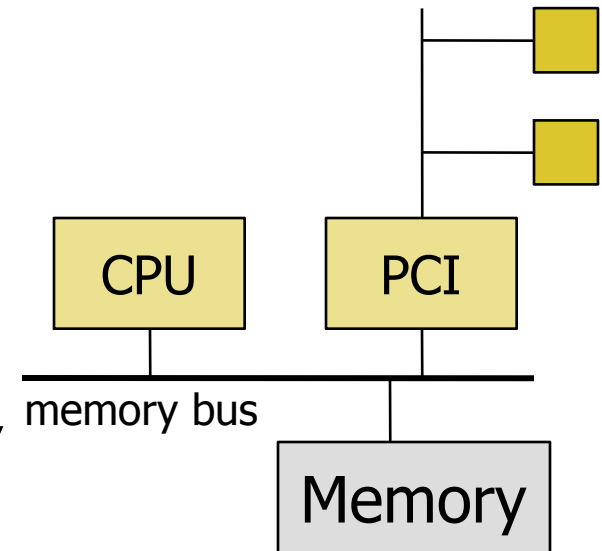


Typical requirements

- Short data in control applications
Large data in media applications
- Short periods (ms)
 - monitoring, feedback control
- Fast aperiodic (ms)
 - alarms
- Non real-time data
 - configuration, logs
- Multicast
- Predictability in the presence of faults

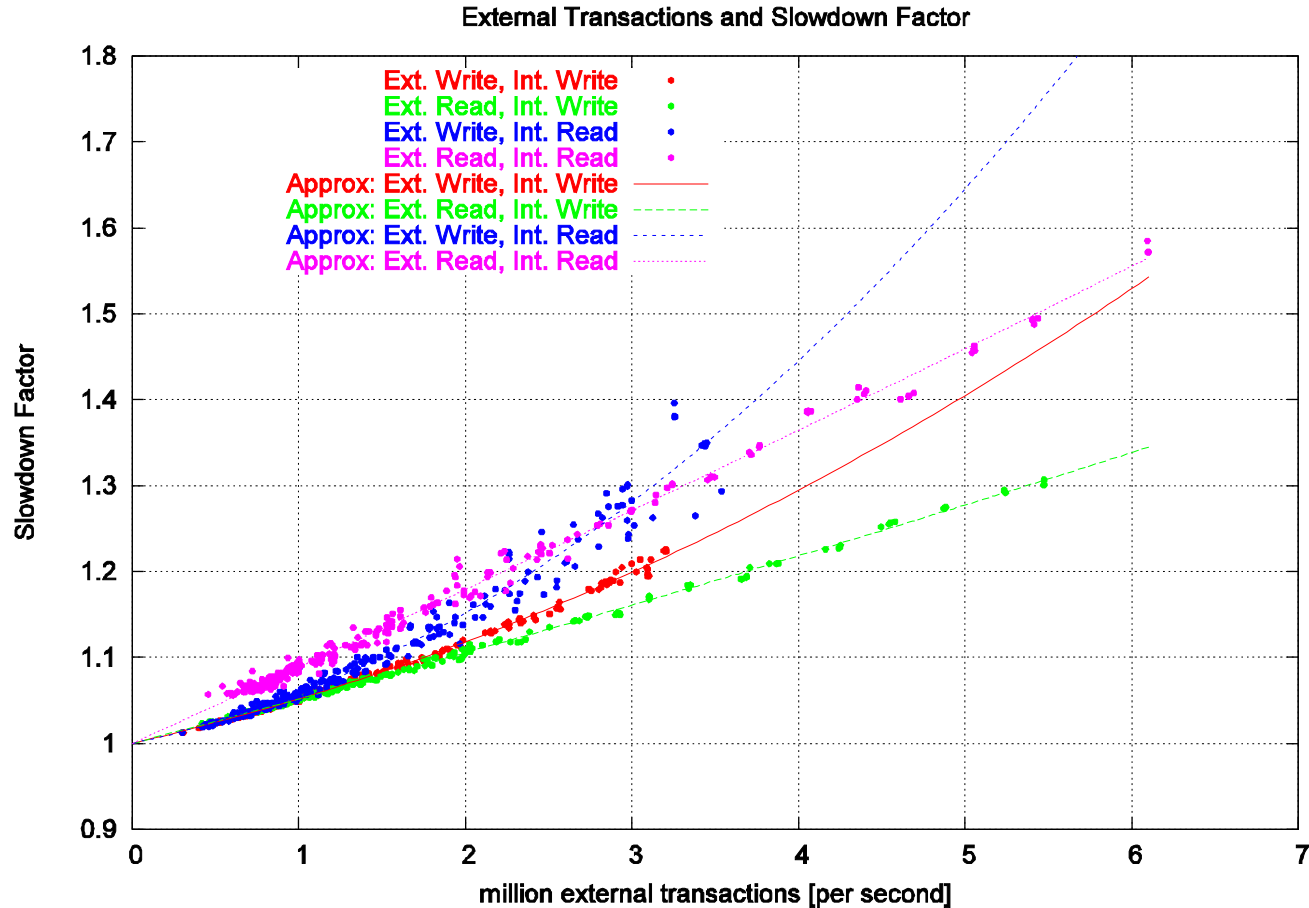
Intra-Node Communication: IO Bus (PCI)

- All data transfers share the memory bus
 - Memory bus conflicts
 - PCI devices start transfers when data is available
 - CPU-Memory-accesses are influenced by PCI devices
 - Applications are slowed down by external load



- Slowdown factor F : $(ET: \text{execution time})$
 $ET \text{ under external load} / ET \text{ without any external load}$
 - application-specific
depends on memory access pattern
 - obtained by measurements
 - ~~attempts to statically predict~~

Some Measurements



Source: Dissertation Sebastian Schönberg (see our Web-Pages)

Intra-Node Communication: PCI Bus

- PCI Bus specification does not prevent real-time
 - Arbitration responsible for granting bus to devices
 - Current implementations use round-robin arbitration
- Telecom Systems
 - PCI bus for control data
 - TDMA bus for real-time data
- Alternative: real-time capable PCI bus arbiter
 - Assign individual share of the PCI bus to devices
 - Enforce reservation
 - Give unused bandwidth to time-sharing devices
- Research topic
 - References: Schönberg, Philips

Networks as Schedulable Resources

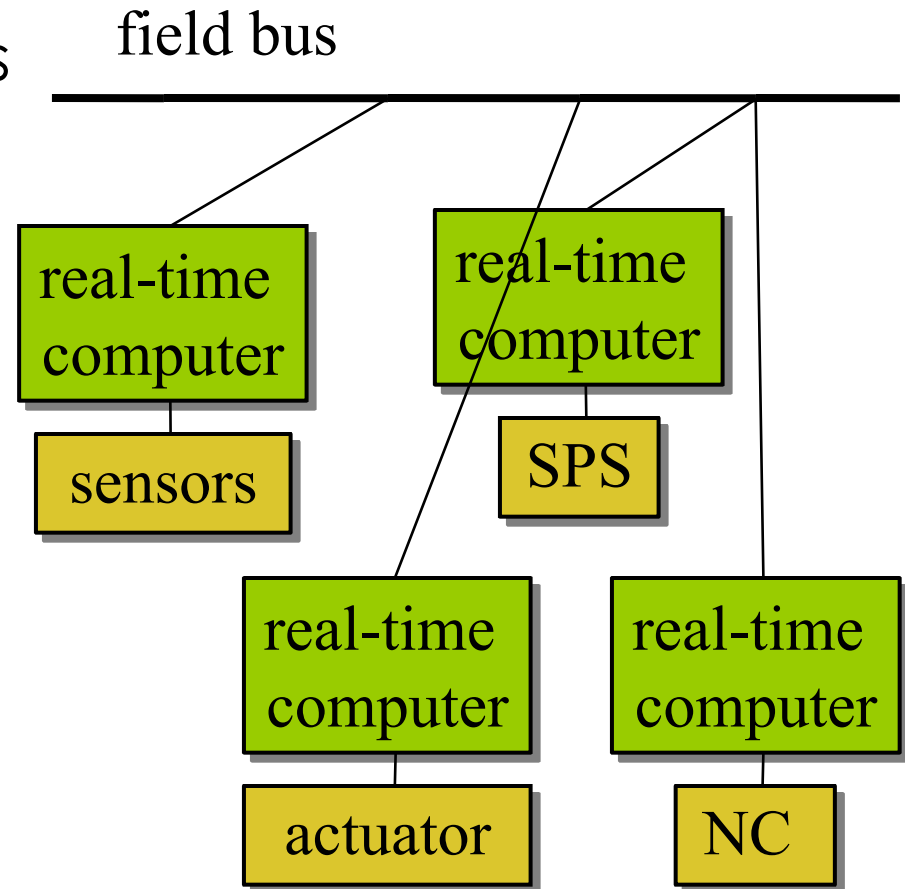
- analogous to CPU scheduling:
 - time driven
 - priority driven
 - (Weighted Round Robin)
- Analogy:
 - WCET: Amount of traffic / message transmission time
 - Periodic tasks: periodic messages
messages are usually non-preemptive !!
 - Deadline: max delay
 - Admission: connection establishment
 - Schedulers in switches or distributed on nodes

Examples

- Fieldbus:
 - CAN, Token Ring: priority based scheduler
 - TTP/Flexray/TT-Ethernet: time driven scheduling
- Switch:
 - Delay EDD
 - Weighted Round Robin (neither priority nor time driven)

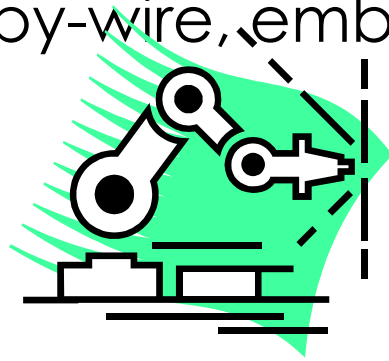
Intra-Node Communication: Field Busses

- Situation is moving
 - Smart sensors, actors
 - Wireless lans

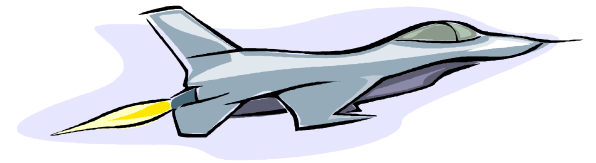


Fieldbus

- Networks for process control, factory automation, cars, avionics, X-by-wire, embedded applications



- Networks are typically called fieldbusses:
 - collapsed network stacks (application services access the data link directly)
 - real-time transport
 - short application messages (no need for fragmentation and reassembling)
 - often a bus, hence a single broadcast domain, no



Multiple Access to Shared Medium

- CSMA/CD shared ethernet
- CSMA/BA with priorities (CAN)
- Token Ring with priorities
- TDMA

CAN - Controller Area Network

- Bosch GmbH, Version 2.0 released in 1991
- ISO Standards 11519 (94) and 11898 (95)
- used in automotive industry, process control, manufacturing automation, embedded application domains
- cost-effective

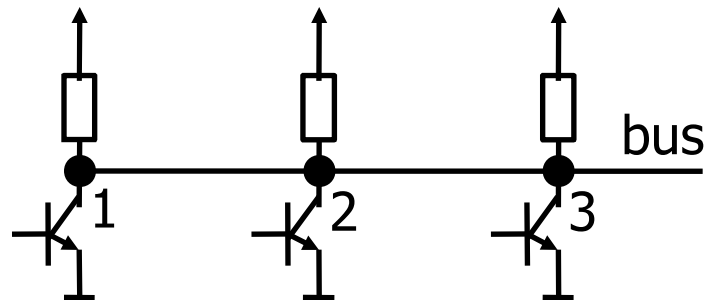
- CSMA/BA:
carrier sense multiple access/bitwise arbitration

Bus Arbitration with CAN (Priority Based)

- CSMA/BA (carrier sense multiple access/bitwise arbitration)
- Bit-wise collision resolution using message identifiers, sent from HSB to LSB
- Wired "and", implemented with bus drivers

0 - Dominant level

1 - Recessive level



- Node stops if a "0" is seen when sending a "1"

CAN Message



- Bit stuffing to allow synchronization at bit level
 - after 4 '0's or 4 '1's: dummy 1 or 0
- Transmission time of a message in bit-times:

$$C = 13 + 34 + \text{databits} + (34 + \text{databits} - 1) / 4$$

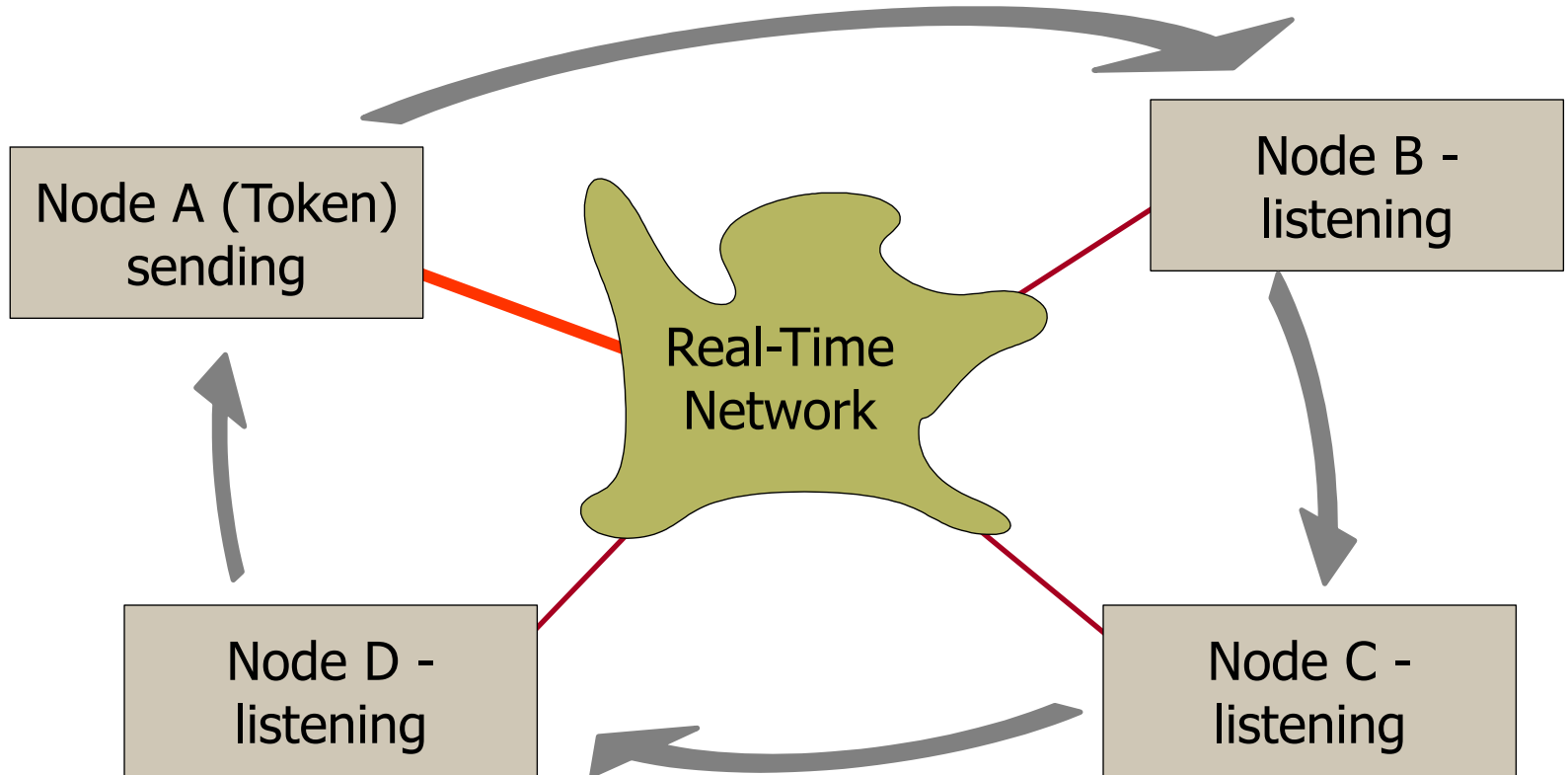
CAN - Performance

- Transmission rate from 5KBit/s to 1MBit/s
- Bit-synchronized bus access
- Bus length depends on transmission rate
 - 40m with 1MBit/s, 1000 with 50KBit/s
- transmission rate increases with shorter busses
... or with increased speed of light

- 2048 priorities (11 Bit) in version A,
>500 Mio priorities (29 Bit) in version B
- Short messages (0-8 Byte)
- Efficiency depends on msg length: 0%..47%

Medium Access Control using a Token Ring

- Messages follow a logical ring
- Token required as permission to send (round robin scheduling)



Token Ring with Priorities

- Ring token has 2 priority fields:
Token Priority, Reservation Priority
- Token can be taken only, if **priority field** in token is lower than priority field in outgoing packet
- **Reservation Priority** in packet field is set to highest priority of waiting messages that are passed by packet
- Upon arrival at sender, the **token priority** is set to the **packet priority reservation field**
- **A station that raises the priority must lower it when it sees a free token**
- **Good example in**
http://www.wildpackets.com/support/compendium/token_ring/token_priority

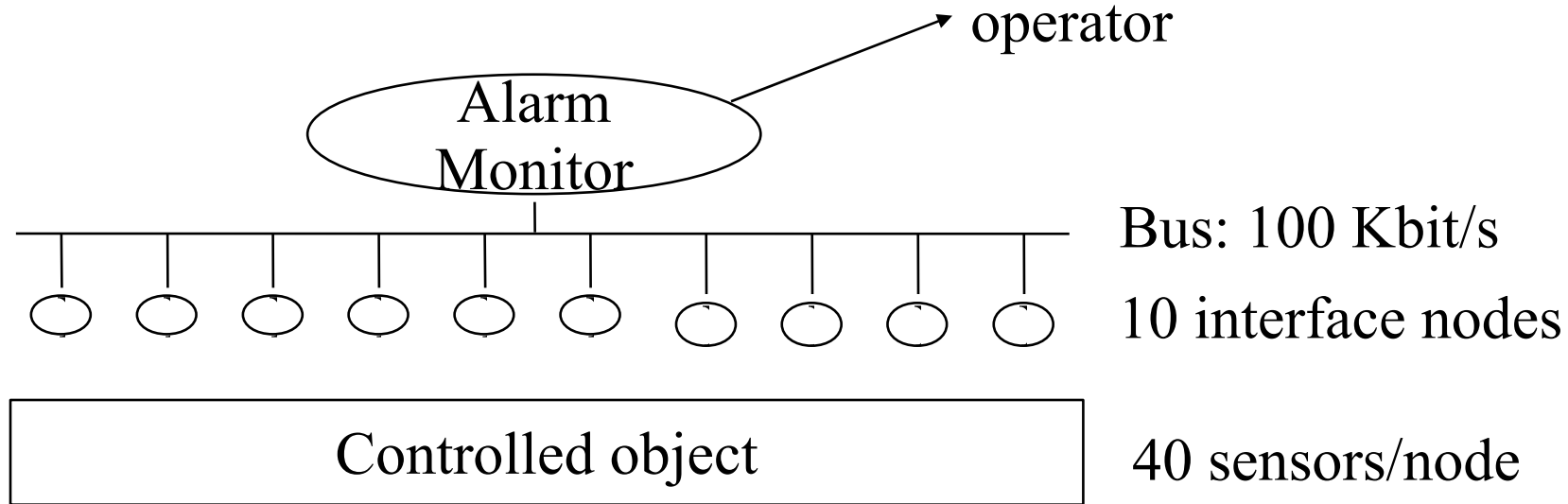
More on Tokens

- General medium access protocol
 - Not necessarily a strict ring
 - Master/Slave
 - Table-based
- Functional Parameters
 - Maximum token holding time at each node
 - Rings: resulting maximum token rotation time
- Problems:
 - lost tokens
 - duplicated tokens

Time Division Multiple Access (TDMA)

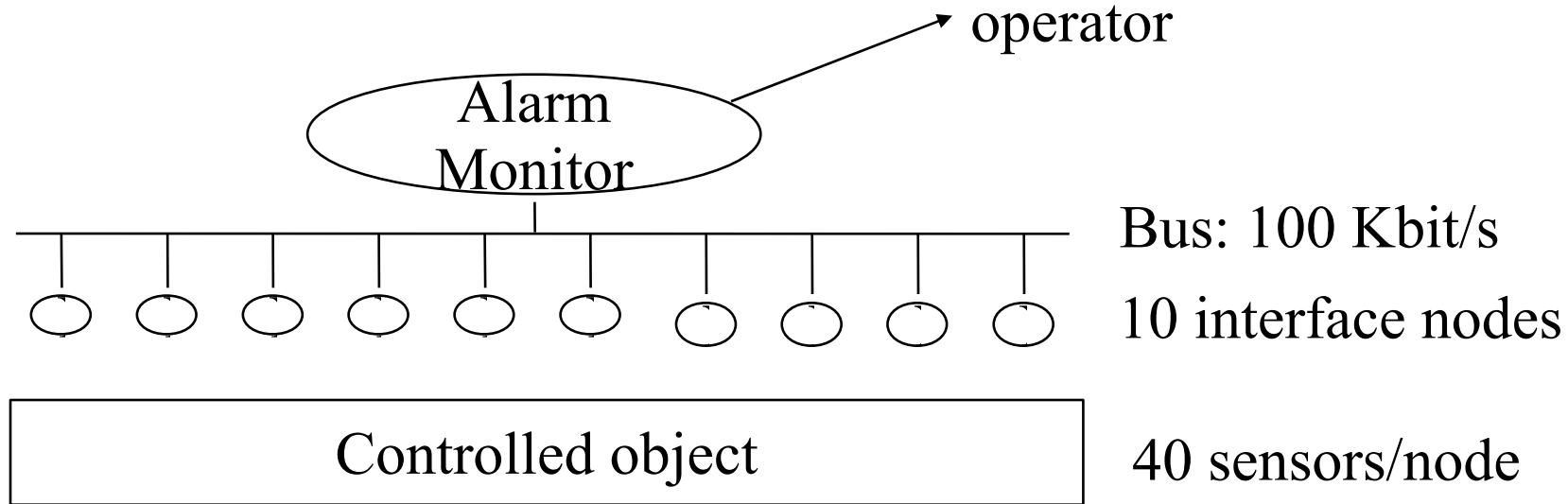
- Divide Time periodically into Slots
- Allocate Slots
- hope/enforce usage of Slots

Kopetz: Event vs. Time Triggered



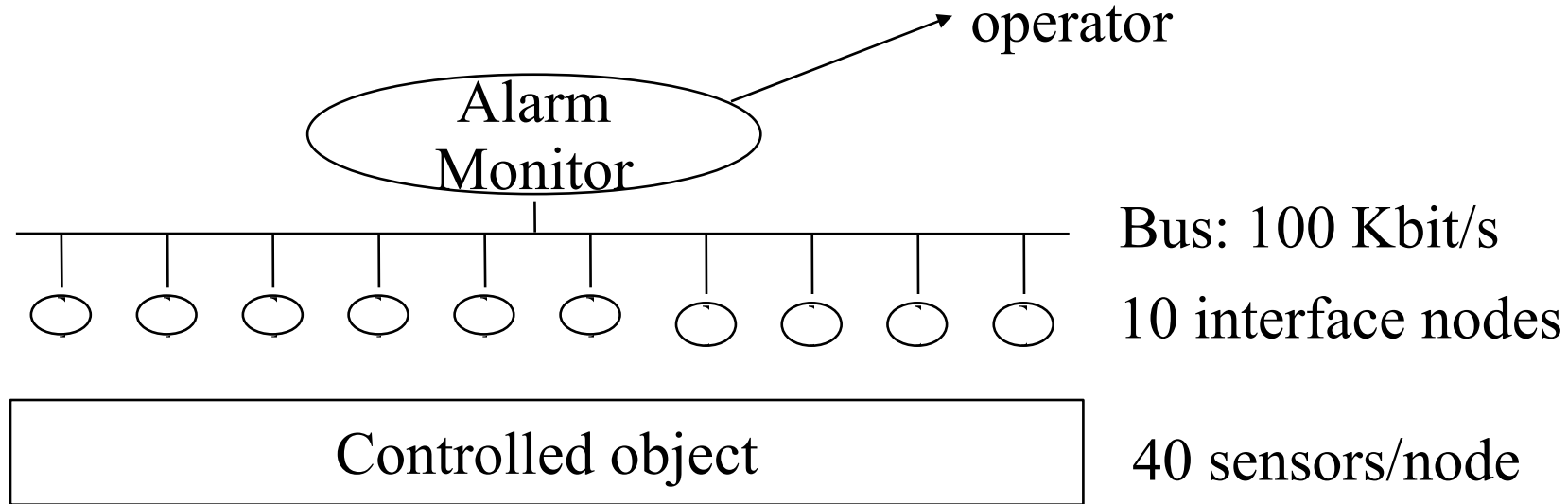
- Each node supervises 40 alarm conditions and sends messages (as soon as possible)
- Deadline: 100 ms to notify operator
- Communication bandwidth: 100 kbits/second

Kopetz: Event Triggered (ex. CAN)



- Send message as soon as alarm condition is seen
 - alarm name 8 Bit, CAN overhead 44+4 Bit \Rightarrow 56 Bit
- allows 180 messages within deadline of 100ms
- worst case requirement: 400 messages ...

Kopetz: Time Triggered (TDMA bus)



- Each node sends state message every 100ms
 - Data field: 40 Bit + overhead 44 + gap 4 \Rightarrow 88 Bit
- allows 110 messages within 100ms period
- but, just 10 required!
 - i.e. 10 % of bandwidth

Real-Time Communication in WANs

principles

- “find” connection (reservations)
- determine local properties at each switch of connection
depending on switch-local scheduler
- add up delays
- take care for jitter (provide buffer, traffic shapers)

Schedulers in Switches

- EDF: e.g., Delay-EDD “Service Discipline”
- static priorities
- ...
- see text books

- weighted round robin

Scheduling Example: Weighted Round Robin

Principle:

- Message passes multiple switches from source to dest.
- Scheduler at each switch looks at input buffers in round robin fashion
processes *weighted* number of units of messages per round

Scheduling Example: Weighted Round Robin

Messages:

- e_i max „length“ of message in “units”
- p_i Period: minimum inter-arrival time
- number of hops
switches passed by message from
source to destination

guaranteed end-to-end delay ??

Scheduling Example: Weighted Round Robin

Switches:

w_i each connection is given w_i slots per round.
 RL round length (switch design parameter)
 e_i max „length“ of message in units processed per slot if $w_i=1$

□ $w_i \leq RL$

Roundlengths ?

Weights?

Delay? Per Node, end to end?

Scheduling Example: Weighted Round Robin

Some Calculations:

Delay per Switch: $RL * (\lceil e_i / wt_i \rceil)$

RL is bound by delay guarantees of switch
for example, to guarantee that messages are
transmitted within a period:

$$RL \leq \min p_i$$

$$wt_i \geq \lceil e_i / \lfloor p_i / RL \rfloor \rceil$$

end to end delay (pipelined “execution”):

$$(\lceil e_i / wt_i \rceil + \lceil n - 1 \rceil) * RL$$

RT CORBA (1.0)

- Extensions, not modifications
- Fixed Priority Systems
- End to end predictability for fixed priority CORBA systems

RT CORBA

- Extend CORBA by:
 - Thread priorities
 - Bound duration of thread priority inversion
 - Bound latency of operation invocation

- In each RT-CORBA system involved:
 - RT OS with scheduling mechanism
 - RT object request broker
 - RT communication
 - Application

Priorities

- Assign platform independent RT-CORBA priority to invocations
- ...
- Map to priorities of underlying OS consistently between nodes with different native priority schemes

```
//IDL
module RTCORBA {
typedef short priority;
const priority minPriority = 0;
const priority maxPriority = 32767;}
```

Some Extensions to CORBA

- Thread interface: Scheduling Policy
- Two server priority models
 - Client priority propagation
 - Priorities assigned at server side
- Thread pools
 - Preallocation (reduce priority inversion)
 - Partitions threads to isolate applications

Priority Banded Communication

- Multiple Connections to reduce priority inversion



- Banding
 - Each connection may represent a range of priorities
 - Different range in each band (including 1)

Summary

- Field busses
 - Scheduling analysis analog to processors
 - Use time driven / static priority / table based scheduling
- Switched LANs
- WANs
 - Dynamic
 - Use local scheduling disciplines in switches/routes to enforce system wide behaviour
 - Admission for connections required

References

- J.-Y. Le Boudec and P. Thiran: **Network Calculus**. Springer Verlag, LNCS volume 2050, 2001.
- H. Kopetz: **Real-Time Systems**. Kluwer Academic Publishers, 1997.
- Othmar Kyas: **ATM-Netzwerke**. DATACOM-Buchverlag, 1996.
- J. Loeser, H. Härtig: **Low-latency Hard Real-Time Communication over Switched Ethernet**. In Proceedings of the 16th Euromicro Conference on Real-Time Systems (ECRTS 2004), June 2004, Catania, Italy.
- Malcolm, Zhao: **Hard Real-Time Communication in Multiple-Access Networks**. In Real Time Systems Journal, Kluwer Academic Publishers, 1995.
- S. Schönberg: **Impact of PCI-Bus Load on Applications in a PC Architecture**. In Proceedings of the 24th IEEE Real-Time Systems Symposium (RTSS 2003), December 2003, Cancun, Mexico.
- www.profibus.com

Switched Ethernet in Real-Time

Message loss properties:

- No collisions !
- But buffer overflows

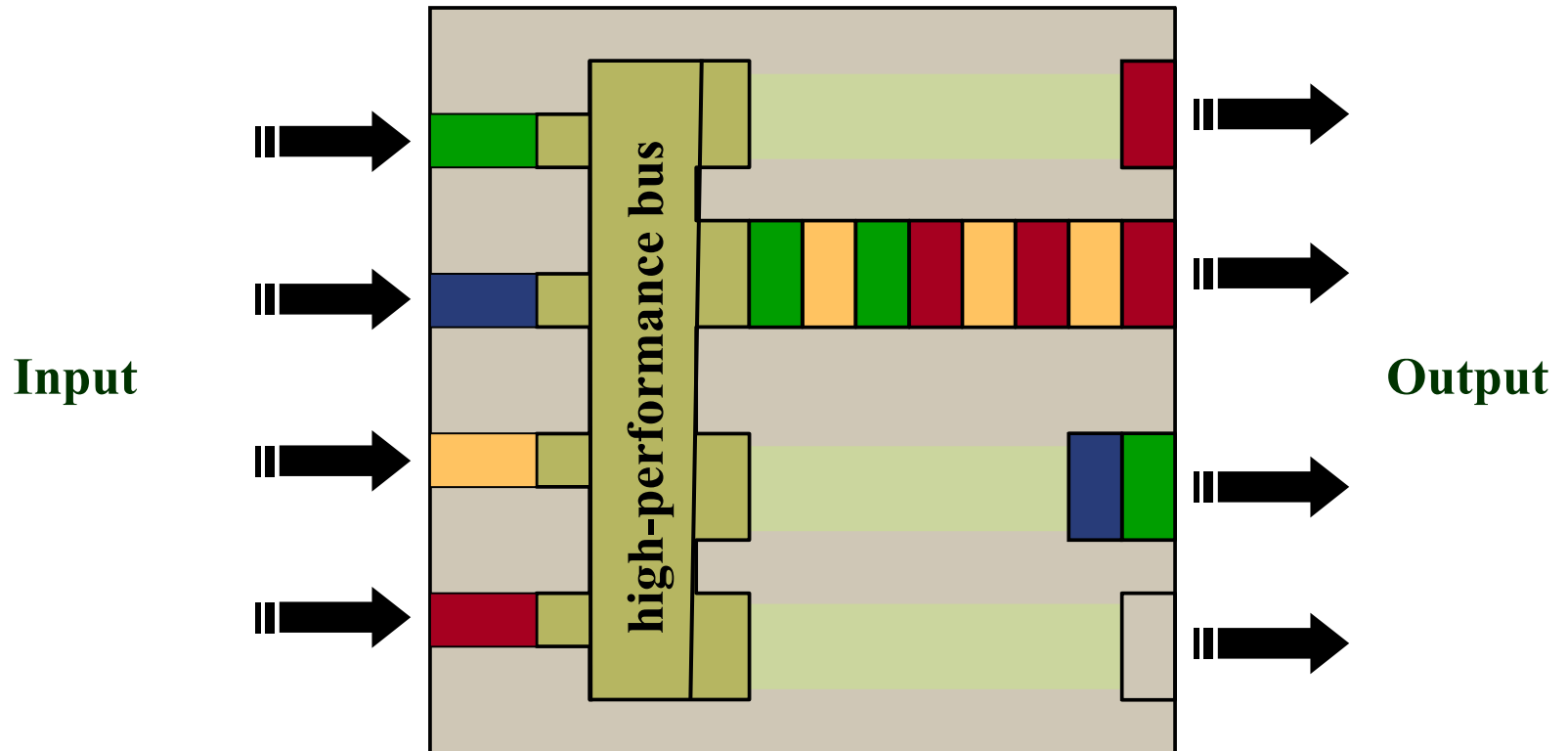
Timing properties:

- Switching fabric
- Buffer draining

Principles:

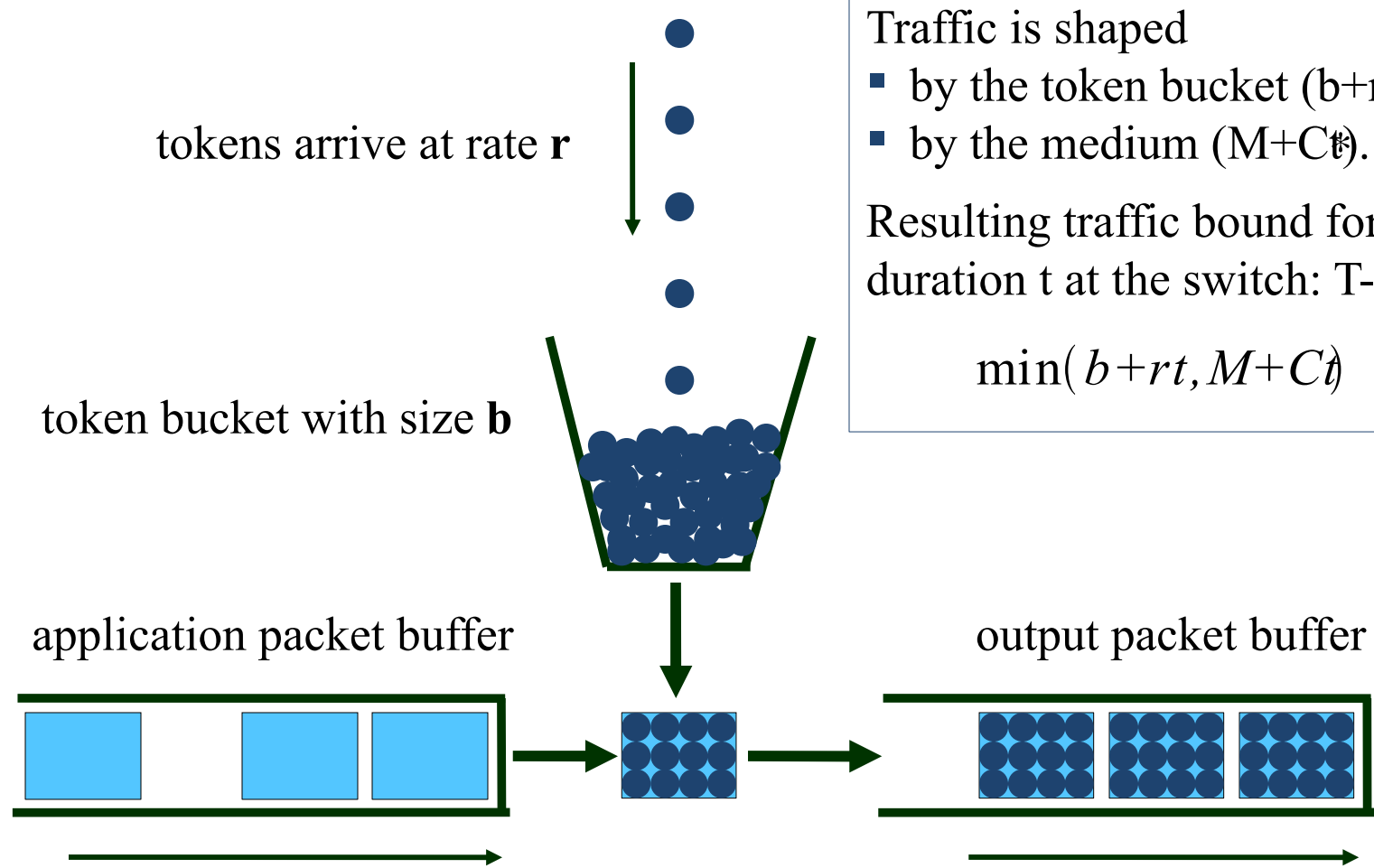
- Participants limit flow, for example enforced by OS
- Time-driven operation

Switched Ethernet (Löser)



- Delay bound depends on traffic to the output port
- If queues get too long, frames are dropped
- ~~Traffic on input ports must be bound~~

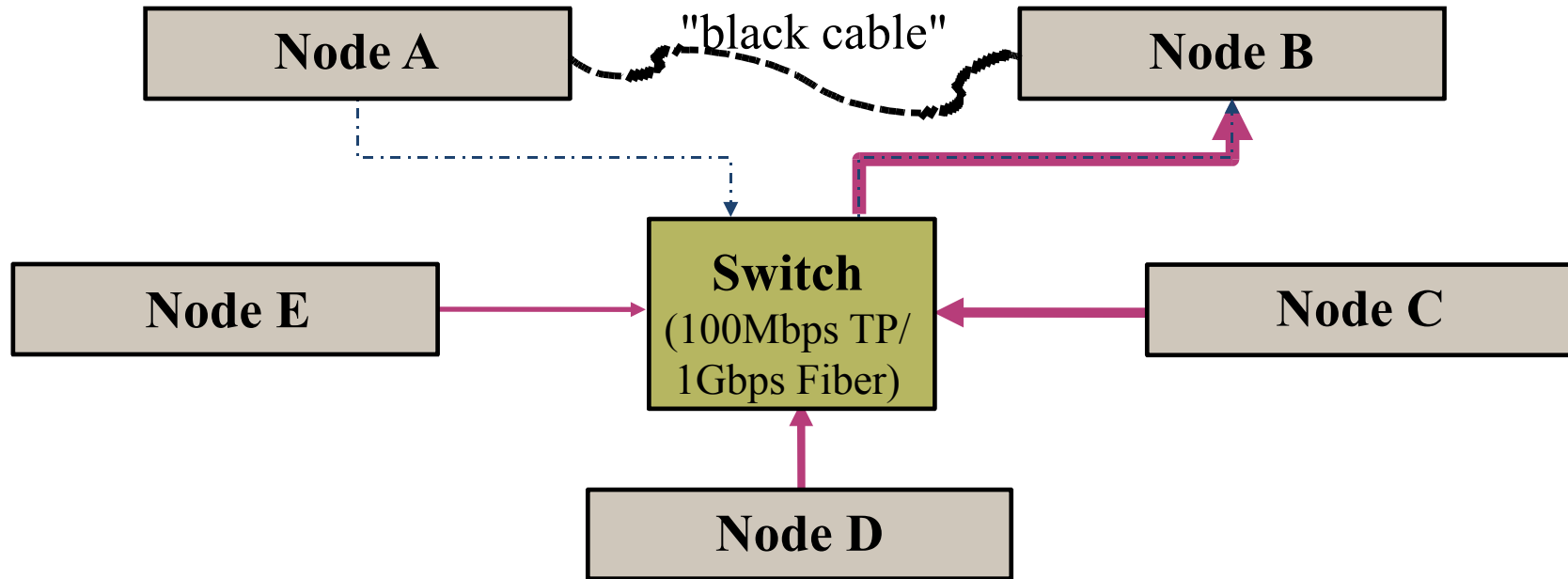
Token Bucket Shaper at Sender (Löser)



Shaper Implementation by OS Trade Off (Löser)

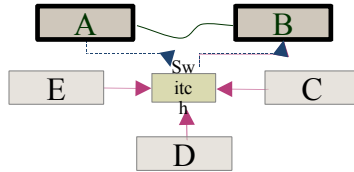
- Short delay guarantees
- Requires: smaller message jitter
- Requires: shorter shaper-process periods
- Requires: more context switches + shorter process periods

Traffic Shaping: Measurements(Löser)



- Node A: 64 byte frames to Node B, every 1ms
 - Nodes C, D, E: test traffic
- A: Celeron 900MHz, B: Celeron 1.7GHz, C: Duron 800Mhz,
D: Celeron 1.7GHz, E: Duron 900MHz

Traffic Shaping: 100MBit/s, Delays A→B(Löser)



	100μs	1 ms	10 ms
Bandwidth of Nodes C, D, E	C: 40Mbps, D: 32Mbps, E: 20Mbps		
Network utilization	93 %		
Queueing delay bound	502 μs	1.3 ms	9.28 ms
Max. observed app-to-app delay A→B	483 μs	1.3 ms	8.76 ms