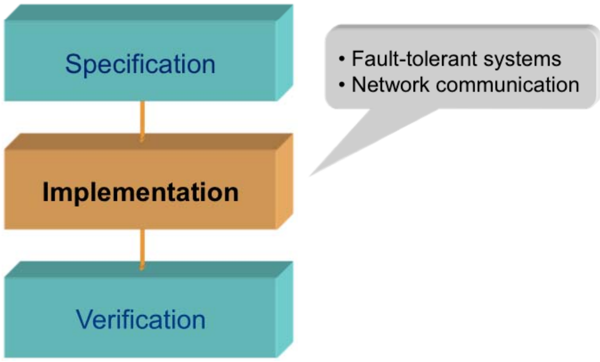
CHALMERS

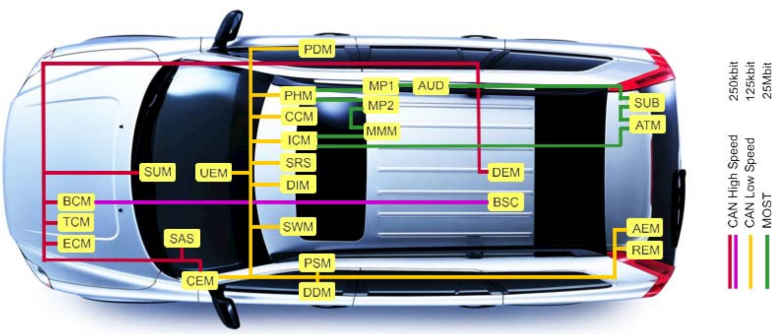
Real-Time Systems



CHALMERS

Dependable Distributed Real-Time Systems

Aircraft/automotive embedded systems require support for fault-tolerance as well as network communication



CHALMERS

Fault-tolerant systems

What are the effects if the hardware or software is not fault-free in a real-time system?

CHALMERS

Fault-tolerant systems

What causes component faults?

- Specification or design faults:
  - Incomplete or erroneous models
  - Lack of techniques for formal checking
- Component defects:
  - Manufacturing effects (in hardware or software)
  - Wear and tear due to component use
- Environmental effects:
  - High stress (temperature, G-forces, vibrations)
  - Electromagnetic or elementary-particle radiation

CHALMERS

Fault-tolerant systems

What types of faults are there?

- Permanent faults:
  - Total failure of a component
  - Caused by, e.g., short-circuits or corrupted data structures
  - Remains until component is repaired or replaced
- Transient faults:
  - Temporary malfunctions of a component
  - Caused by, e.g., ion radiation or power fluctuation
- Intermittent faults:
  - Repeated occurrences of transient faults

CHALMERS

Fault-tolerant systems

How are faults handled at run-time?

- Error detection:
  - Erroneous data or program behavior is detected
    - Watchdog mechanism, comparisons, diagnostic tests
- Error correction:
  - The originally-intended data/behavior is restored
    - Intelligent codes used for restoring corrupt data
    - Check-pointing used for restoring corrupt program flow
- Fault masking:
  - Effects of erroneous data or program behavior are ”hidden”
    - Time (re-execute code) or space (replicated hardware) redundancy
    - Voting mechanism (e.g., majority voting) or N-modular redundancy (i.e., $2m+1$ units to mask m faults)

CHALMERS

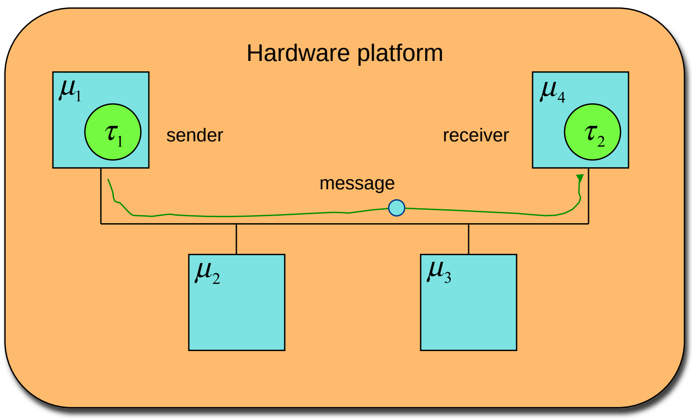
Fault-tolerant systems

To extend real-time computing towards fault-tolerance, the following issues must be considered:

1. What is the fault model used?
   - What type of fault is assumed?
   - How and when are faults detected?
2. How should fault-tolerance be implemented?
   - Using time redundancy (re-execution)?
   - Using space redundancy (replicated tasks/CPUs/networks)?
3. What scheduling policy should be used?
   - Extend existing policies?
   - Suggest new policies?

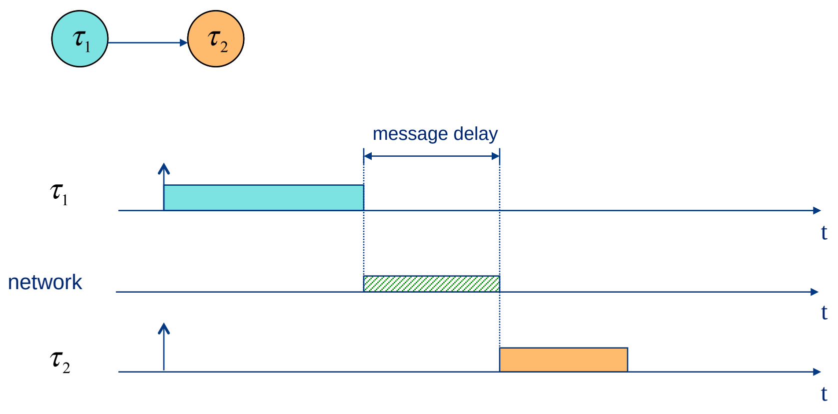
CHALMERS

Network communication



CHALMERS

Network communication



CHALMERS

Network communication

Message delay:

- Message delays are caused by the following overheads:
  - Formatting (packetizing) the message
  - Queuing the message, while waiting for access to medium
  - Transmitting the message on the medium
  - Notifying the receiver of message arrival
  - Deformatting (depacketizing) the message

Formatting/deformatting overheads are typically included in the execution time of the sending/receiving task.

CHALMERS

Network communication

Queuing delay:

- The cause of the queuing delay for a message depends on the actual network used. For example:
  - Waiting for a corresponding time slot (TDMA)
  - Waiting for a transmission token (Token Ring)
  - Waiting for a contention-free transmission (Ethernet)
  - Waiting for network priority negotiation (CAN)

CHALMERS

Network communication

Transmission delay:

- The delay for transmitting the message is a function of:
  - Message length (bits)
  - Data rate (bits/s)

$$t_{\text{frame}} = \frac{N_{\text{frame}}}{R}$$

and

  - Communication distance (m)
  - Signal propagation velocity (m/s)

$$t_{\text{prop}} = \frac{L}{v}$$

CHALMERS

Network communication

How is the message transferred onto the medium?

- Contention-free communication:
  - Senders need not contend for medium access at run-time
  - Time-division, multiple-access (TDMA)
- Token-based communication:
  - Each sender using the medium gets one chance to send its messages, based on a predetermined order
- Collision-based communication:
  - Senders may have to contend for the medium at run-time
  - Ethernet, CAN

CHALMERS

Network communication

TDMA-based protocols:

- One or more dedicated time slots for each processor:
  - Example: medium access is divided into minor communication cycles (CC) and major system cycles (SC)
  - Message queuing delay is bounded (can be made negligible with appropriate scheduling)
- Examples:
  - TTP/C (Time-Triggered Protocol)
  - FlexRay

CHALMERS

Network communication

Token-based protocols:

- Utilize a token for the arbitration of message transmissions on a shared medium
  - The sender is only allowed to transmit its messages when it possesses the token
  - Message queuing delay is bounded
- Examples:
  - Token Bus (IEEE 802.4)
  - Token Ring (IEEE 802.5)
  - FDDI

CHALMERS

Network communication

Ethernet-based protocols:

- Senders attempt to send a complete message
  - Collision-detect mechanism is used to determine if there is a need for re-transmission
  - Message queuing delay can in general not be bounded!

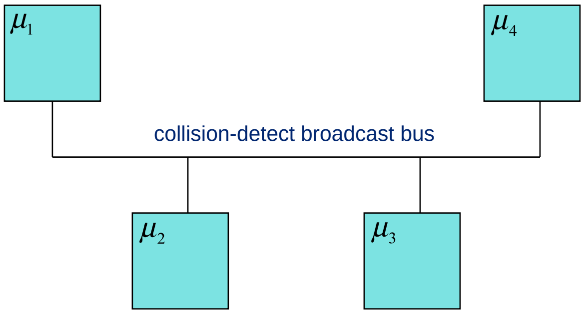
CAN protocol:

- Senders transmit a message header (with an identifier)
  - Collision-detect mechanism is used to determine who will be allowed to send the entire message
  - Message queuing delay can be bounded with appropriate identifier assignment

CHALMERS

The CAN protocol

Controller Area Network (CAN): (Bosch 1991, SAE 1993)



CHALMERS

The CAN protocol

CAN message frame format: (short format)

SOF	11-bit identifier	control	0 - 8 bytes of message data	error control	Ack	EOF

11-bit identifier is used for two purposes:

- assign a priority to the message (low number ⇒ high priority)
- enable receiver to filter messages

Wired-AND:

Each node monitors the bus while transmitting.

If multiple nodes are transmitting simultaneously and one node transmits a '0', then all nodes will see a '0'. If all nodes transmit a '1', then all nodes will see a '1'.

CHALMERS

The CAN protocol

CAN protocol: (binary countdown)

1. Each node with a pending message waits until bus is idle.
2. The node begins transmitting the highest-priority message pending on the node. Identifier is transmitted first, in the order of most-significant bit to least-significant bit.
3. If a node transmits a recessive bit ('1') but sees a dominant bit ('0') on the bus, then it stops transmitting since it is not transmitting the highest-priority message in the system.
4. The node that transmits the last bit of its identifier without detecting a bus inconsistency has the highest priority and can start transmitting the body of the message.

CHALMERS

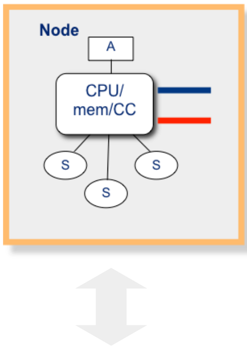
Dependable distributed networks

Contemporary communication networks suitable for dependable distributed real-time systems

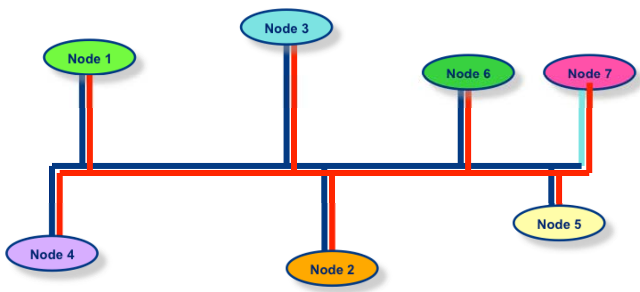
- TTCAN:
  - Widely used in today's automotive electronic systems
- TTP/C:
  - Operational in civil aircrafts
- FlexRay:
  - Anticipated in next generation automotive electronic systems

CHALMERS

The TTCAN protocol



- Based on the CAN protocol
- Bus topology
- Media: twisted pair
- 1Mbit/s

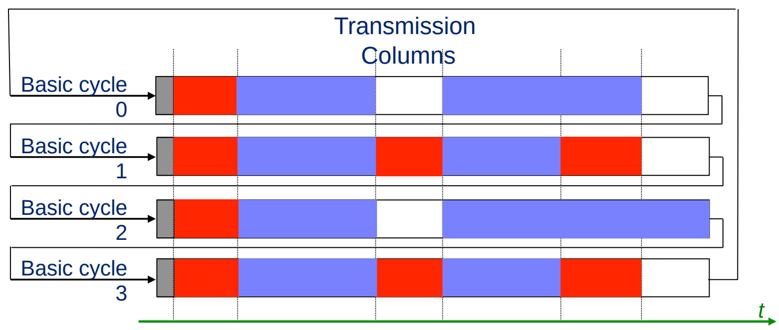


A second controller is required to implement the redundant bus

CHALMERS

The TTCAN protocol

- ”Exclusive” – guaranteed service
- ”Arbitration” – guaranteed service (high ID), best effort (low ID)
- ”Reserved” – for future expansion…

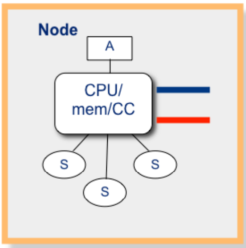


Time is global and measured in *network time units* (NTU's)

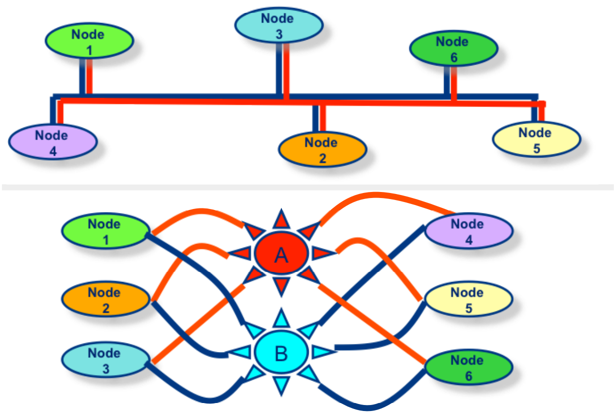
CHALMERS

The TTP/C protocol

- Double channels (one redundant). Bus topology or "star" (optical)
- Media: twisted pair, fibre
- 10 Mbit/s for each channel



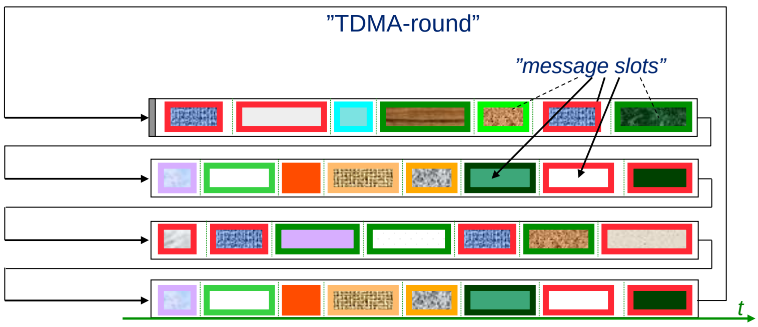
A network is built on either twin buses or twin stars.



CHALMERS

The TTP/C protocol

All communication is statically scheduled
Guaranteed service

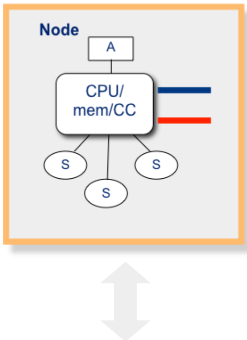


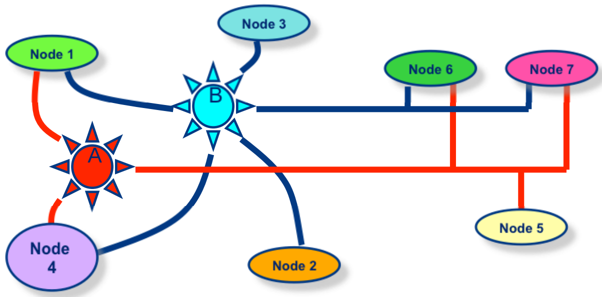
Non-periodic messages have to be fitted into static slots by the application

CHALMERS

The FlexRay protocol

- Double channels, bus or star (even mixed).
- Media: twisted pair, fibre
- 10 Mbit/s for each channel





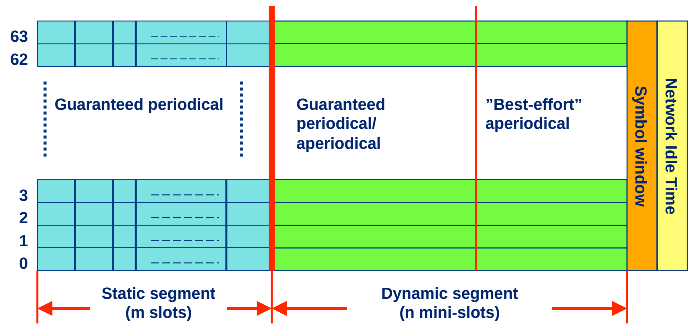
Redundant channel can be used for an alternative schedule

CHALMERS

The FlexRay protocol

"Static segment" (compare w/ TTCAN "Exclusive")
– guaranteed service

"Dynamic segment" (compare w/ TTCAN "Arbitration")
– guaranteed service (high ID), "best effort" (low ID)



Max 64 nodes on a Flexray network.

CHALMERS

Real-Time Systems

Facing the written exam

Tuesday 14:00 – 18:00, March 15, 2011
in the "V" building

Note: in case you need to take a re-exam, you must remember to register in the Student Portal

CHALMERS

Facing the exam

Reading guidelines:

- Lecture notes ("PowerPoint hand-outs")
  - All material are **very** relevant
  - No exam questions regarding the guest lectures!
- Course book: "Real-Time Systems …", Burns & Wellings
  - Overview reading (chapters given on course web page)
- Compendium: "Real-Time Systems …", Tindell
  - Overview reading (chapters given on course web page)
- Compendium of examples
  - Good experience in solving theoretical analysis problems

CHALMERS

Facing the exam

Permitted to use during the exam:

- Chalmers-approved calculator
  - Important aid for feasibility analysis problems
- ”Ada Distilled” (Richard Riehle) + ”Ada vs Java” (Quick Ref.)
  - Important aid for understanding basic principles of Ada
- Ada95 Reference Manual
  - Important aid for understanding how to:
    - write parallel programs in Ada95
    - implement low-level operations in Ada95
    - express real-time properties in Ada95

CHALMERS

Facing the exam

Important knowledge areas:

- Design principles for real-time systems
  - Real-time systems: typical properties, misconceptions
  - Real-time constraints: origin, interpretation (soft/hard)
  - Design phases: specification, implementation, verification
  - Verification: methods, difficulties, pitfalls
- Real-time kernels
  - Task management: data structures, task states, task switches
  - Services: actions taken for different types of system calls
  - Memory management: fundamental principles
  - Fault tolerance and data communication

CHALMERS

Facing the exam

Important knowledge areas (cont'd):

- Principles of parallel programming
  - Parallelization: pros & cons
  - Mutual exclusion: definition, implementation
  - Deadlock: definition, management
  - Starvation: definition, management
- Language constructs for parallel programming in Ada95
  - Tasks: creation, synchronization
  - Shared objects: protected objects, semaphores, monitors
  - Real-time: concept of time, delays, priorities
  - Low-level: I/O-addressing, bit manipulation, interrupt handling

CHALMERS

Facing the exam

Important knowledge areas (cont'd):

- Scheduling theory
  - Task model: WCET, deadline, period, offset
  - Scheduling: definitions, priorities, preemption
  - Feasibility test: purpose, exactness (sufficient/necessary)
- Static scheduling
  - Properties: time table, pros & cons
  - Scheduling: generation of time tables, run-time behavior
- Dynamic scheduling (RM, DM, EDF):
  - Properties: priority assignment, optimality, pros & cons
  - Scheduling: run-time behavior, construct timing diagram
  - Feasibility test: theory, assumptions, exactness, complexity

CHALMERS

Facing the exam

What type of exam problems will there be?

- Real-time computing concepts
  – Will probe your general knowledge in real-time computing
- Programming concepts
  – Will probe your general knowledge in how to design parallel real-time programs
  – No exam problems where you will write lots of program code!
- Scheduling concepts and theory
  – Will probe your knowledge in WCET analysis, scheduling and feasibility analysis

Let yourself be inspired, but not controlled, by the contents of old exams!

CHALMERS

Real-time systems

… and then ...

… what to do if you are curious and want to know more?

CHALMERS

Design of real-time systems

What additional issues are there?

- How are tasks assigned to processors?
  - New possibilities and difficulties arise with multiple processors
- How is system overload handled?
  - What tasks to execute is not always an easy choice
- How are aperiodic tasks handled?
  - Design of server-based / server-less aperiodic task handling
- How is inter-processor communication scheduled?
- How is fault tolerance obtained in the system?

These issues (and more) are addressed in the advanced course in ”Parallel and Distributed Real-Time Systems” (EDA421/DIT171, quarter 2)