Traffic Class Assignment for Mixed-Criticality Frames in TTEthernet

Voica Gavriluț Paul Pop

Technical University of Denmark
Kongens Lyngby, Denmark

Outline

- Motivation
- Architecture and application models
- The TTEthernet protocol
- Problem formulation and motivational example
- Optimization strategy: Tabu Search and cost function
- Experimental results
- Summary, message and future work

Motivation

- Trend: From "federated" to "integrated" architectures, where distributed applications of different criticality share the same platform
- *Mixed-criticality systems*: integrate safety-critical, mission-critical and non-critical applications
  - Our focus is on the mixed time-criticality:
    hard real-time, **soft** real-time and **non critical** (non real-time)
    - *Hard real-time*: missing a deadline leads to failure
    - *Soft real-time*: missing a deadline degrades the service
- Safety-critical communication protocols:
  - Specialized protocols in each area (e.g., CAN, FlexRay, SAFEBus, ProfiNet)
  - Trend: extending Ethernet
    - Ethernet: low cost, high speed, but unsuitable for real-time & safety-critical systems
    - Extensions: AFDX/ARINC 664p7, EtherCAT, FTT-Ethernet, **TTEthernet** (our focus)

TTEthernet Traffic Classes

TTEthernet

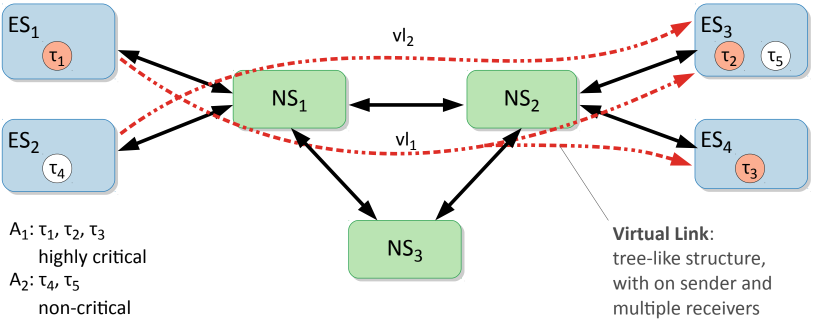
- Standardized as SAE 6802
- ARINC 664p7 compliant
- Developed and marketed by TTTech Computertechnik AG
- Used in several application areas: automotive, aerospace, industrial

Multiple traffic classes support mixed-criticality requirements

- Time-Triggered (TT)
  - Very low latency and jitter
  - The frames are sent based on schedule tables; highest priority
- Rate-Constrained (RC)
  - Compatible with ARINC 664p7; lower priority than TT
  - Guaranteed bandwidth via a "Bandwidth Allocation Gap" (BAG)
  - Bounded worst-case end-to-end latency
- Best-Effort (BE)
  - Standard Ethernet frame
  - No timing guarantees; lowest priority

Our problem: how to assign the traffic classes to mixed-criticality messages

Architecture Model



- Virtual Links (VL)
  - Emulate point-to-point connections and provide the separation required for messages of mixed-criticality
  - Each message has a VL, and we assume that VL routing is given

Application Model

- Mixed-criticality messages

  HRT: periodic hard real-time messages with a hard deadline
  SRT: periodic or sporadic soft real-time messages with a *utility* function
  NC: aperiodic non-critical messages

Message	Source	Destination(s)	Size	Period	Deadline/ Utility
$m_1 \in \mathcal{M}^{HRT}$	ES_1	$\{ES_3, ES_4\}$	80 B	750 μs	200 μs
$m_2 \in \mathcal{M}^{SRT}$	ES_3	$\{ES_2\}$	300 B	2 ms	1.4 ms/ see utility fig.
$m_3 \in \mathcal{M}^{NC}$	ES_2	$\{ES_1, ES_3\}$	1200 B	-	-

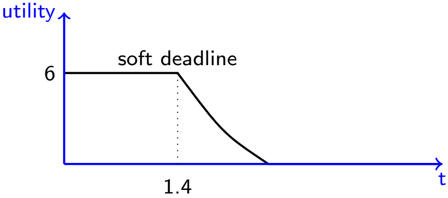


Figure: Example $utility(t)$ function for SRT messages

TTEthernet: TT and RC traffic

- TT Traffic
  - TT frames are sent based on schedule tables and have the highest priority
  - The schedules contain the time when TT frames are sent and received on the links
- RC Traffic
  - RC frames are queued up at the outgoing ports, and have to wait for TT frames and other RC frames
  - A “Traffic Regulator” assures that there is at most one frame sent during a BAG interval L_{max} is the maximum size of a RC frame
- Traffic integration policies:

  Preemtion: The transmission of lower priority message is interrupted and resumed after the integral transmission of the higher priority message

  *timely block: The lower priority message transmission is postponed if it would interfere with the transmission of a scheduled higher priority message

  Shuffling: The transmission of higher priority message is postponed until the lower priority messages sending is finished

Problem Formulation

Given

- The architecture model; the TTEthernet cluster
- The application model $\mathcal{M} = \mathcal{M}^{HRT} \cup \mathcal{M}^{SRT}$ including all the message properties
- Note: for each message we know its VL and the VL routing

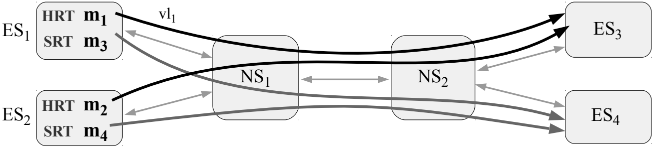
Determine

- The traffic class $\mathcal{TC}(m_i)$ for each message m_i
- The BAG and L_{max} for each RC message
- The sending schedule tables $\mathcal{S}_S$ for each TT message

Such that

- The HRT messages are schedulable and
- The total utility for SRT messages is maximized.

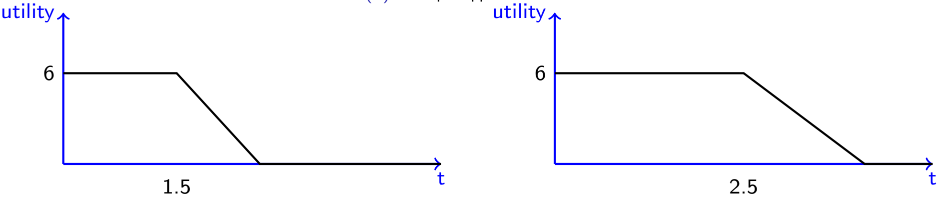
Motivational Example: Introduction



(a) Example architecture model

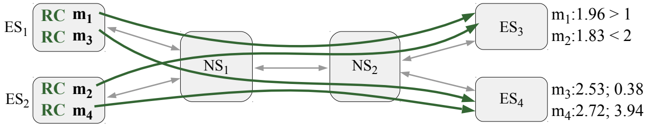
Msg.	Size	Period	Deadline / (Utility)
$m_1 \in \mathcal{M}^{HRT}$	50 B	2 ms	1 ms
$m_2 \in \mathcal{M}^{HRT}$	62.5 B	3 ms	2 ms
$m_3 \in \mathcal{M}^{SRT}$	500 B	4 ms	1.5 ms / (max. 6; 0 at 2.6 ms)
$m_4 \in \mathcal{M}^{SRT}$	750 B	4 ms	2.5 ms / (max. 6; 0 at 4.1 ms)

(b) Example application model

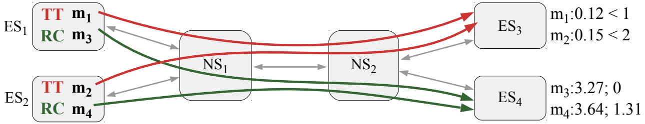


(c) Example Utility Functions

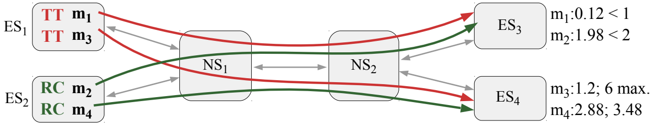
Motivational Example



(a) All messages are RC; m_1 is **not** schedulable; total achieved utility is only 36% out of 12.



(b) HRT messages are TT and SRT are RC. m_1 and m_2 are schedulable, but the total utility is only of 11%.



(c) HRT m_2 is RC, SRT m_3 is TT. HRT are schedulable, and the total utility is increased to 79%. m_3 has a maximum utility.

Optimization Strategy: Tabu Search

Tabu Search meta-heuristic

- Search heuristic
  - Explores the search space using **Design Transformations**
  - Maximizes the **Cost Function**
  - Avoids revisiting recent solutions by labeling them as "tabu"

Cost Function

- $Cost(\Psi) = wp_{HRT} \cdot \delta_{HRT} + \sum_{m_i \in \mathcal{M}^{SRT}} m_i.utility(WCD(m_i))$

  Degree of schedulability:
  $\delta_{HRT} = \sum_{m_i \in \mathcal{M}^{HRT}} \min(0, m_i.deadline - WCD(m_i))$
  WCD is the worst-case end-to-end delay

  TT: given by the schedule table
  RC: determined using a trajectory approach-based analysis method

Design Transformations

- Switch Traffic Class: switches the traffic class of a message
- Modify Schedule: advances or postpones a TT frame
- Modify VL: increases or decreases the BAG and L_{max}

Experimental results

Name	No. HRT msgs.	No. SRT msgs.	SFS		TCA		
			%HRT sched.	%SRT utility	Running time (h:min)	%HRT sched.	%SRT utility
tc1	9	11	44.44%	90.27%	00:50	100%	100%
tc2	11	23	54.54%	85.07%	2:30	100%	99.63%
tc3	17	28	47.06%	64.10%	3:45	100%	95.77%
SAE	40	39	70.00%	81.72%	5:00	100%	94.61%
orion	99	87	45.45%	78.80%	12:30	94.94%	98.68%

- Evaluated algorithms:
  - Traffic Class Assignment (TCA): Our proposed Tabu Search optimization
  - Straightforward Solution (SFS): all messages are RC, and *BAG* and L_{max} are optimized
- 3 synthetic cases and 2 real-life
  - The synthetic test cases have the same topology with an increasing number of messages
  - SAE is the "SAE automotive communication" benchmark
  - Orion is the "Orion Crew Exploration Vehicle" case study
- Implementation and hardware:
  - Java programming language (JDK1.8)
  - Intel Xeon E5-2665 at 2.4 GHz

Summary and message

Summary

- Addressed mixed-criticality applications implemented over TTEthernet networks
- Problem: decide the traffic class of each message
- Solution: Tabu Search-based optimization strategy

Message

- For mixed-criticality message it is not obvious what is the best traffic class
- We need tools to decide the assignment of traffic classes

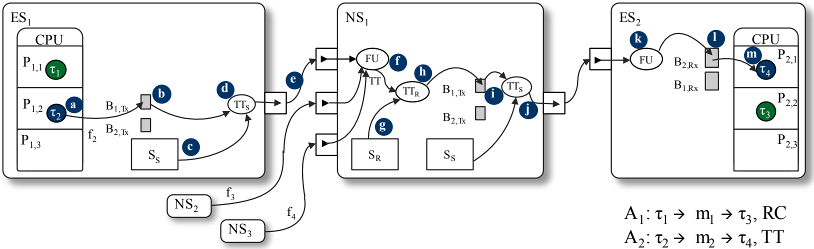
Future work

- Handle the fragmenting and packing of TT frames
- Consider that the traffic class is assigned per dataflow link and not per message
- Ongoing: comparing against an SMT-based solution

Discussion

	Advantages	Disadvantages
TT	▶ Can provide low latency and jitter ▶ There is a SMT-based schedules synthesizer that can handle large systems ▶ Has the most predictable behaviour due to the scheduled traffic	▶ Schedules are not flexible (difficult to add new messages) ▶ The SMT-based approach cannot take into account the RC traffic ▶ RC traffic is still used for legacy reasons ▶ Uses more bandwidth due to the integration policy
RC	▶ There are methods to compute the WCD, so the latency can be bounded ▶ Uses less bandwidth ▶ Better suited for sporadic traffic; no wasted resources ▶ More flexible (easier to add new messages)	▶ Larger latency and jitter ▶ Requires complex analysis and optimization methods for bounded latency and resources utilization

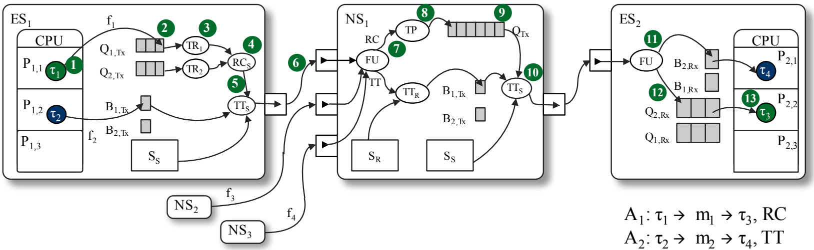
Backup: TTEthernet: TT Example



$A_1: \tau_1 \rightarrow m_1 \rightarrow \tau_3$, RC
$A_2: \tau_2 \rightarrow m_2 \rightarrow \tau_4$, TT

- **a** Packing message m_2 into frame f_2
- **b** Place f_2 in buffer $B_{1,Tx}$ for transmission
- **c** Send time specified in send schedule S_S
- **d** TT_S sends f_2 to NS_1
- **e** f_2 is sent on the dataflow link to NS_1
- **f** The Filtering Unit (FU) checks the frame f_2
- **g** Expected receive time specified in receive schedule S_R
- **h** TT_R checks if f_2 arrives according to schedule
- **i** Place f_2 in buffer $B_{1,Tx}$ for transmission
- **j** Send time specified in send schedule S_S
- **k** FU checks f_2
- **l** Store the frame into receive buffer $B_{2,Rx}$
- **m** Task τ_4 reads f_2 from buffer

Backup: TTEthernet: RC Example

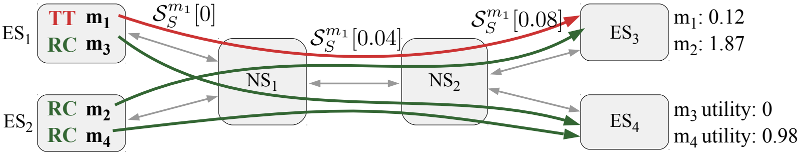


1. Packing message m_1 into frame f_1
2. Insert it in queue $Q_{1,Tx}$
3. Traffic Regulator (TR) ensures bandwidth for each VL
4. RC scheduler RC multiplexes frames coming from TRs
5. TT_S transmits f_1 when there is no TT traffic
6. f_1 is sent on the dataflow link to NS_1
7. FU checks the validity of the frame
8. Traffic Policing (TP) checks that f2 arrives according to the BAG
9. Copy f_1 to outgoing queue Q_{Tx}
10. Send f_1 when there is no TT traffic
11. FU checks f_1
12. Copy to receiving $Q_{2,Rx}$
13. Task τ_3 reads f_1 from the queue

Backup: Optimization Strategy: Design Transformations

- Switch Traffic Class $STC(m_i)$; switches the traffic class of a message:
  - From RC to TT, uses an initial schedules generator
  - From TT to RC, uses $m_i.period$ and $m_i.size$ to determine the vl_i parameters
- Modify Schedules $MS(m_i, postpone)$; affects only TT messages and postpones (when $postpone = TRUE$) or advances the schedules of a message, on all links, keeping the transmission sequence valid
- Modify VL BAG and L_{max} $MVL(m_i, increase)$; affects only RC messages and doubles (when $increase = TRUE$) or halved the $vl_i.BAG$ and $vl_i.L_{max}$

Backup: Optimization Strategy 1

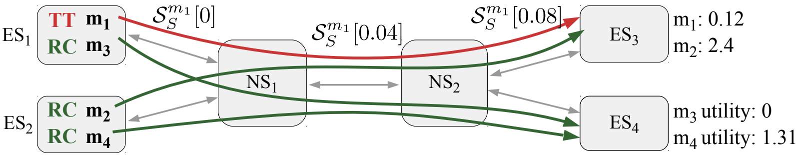


(a) The *current* solution; Cost=0.98

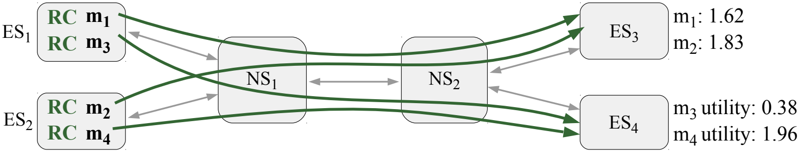
Message	$\mathcal{TC}$	link	$\mathcal{S}_S/(BAG, L_{max})$	iterations
m_1	TT	$NS_1 - NS_2$	[0.09]	14
m_2	RC	—	(4, 125)	5
m_3	TT	$ES_1 - NS_1$	[1]	0
m_3	TT	$NS_1 - NS_2$	[1.3]	7

(b) Tabu list

Backup: Optimization Strategy 2

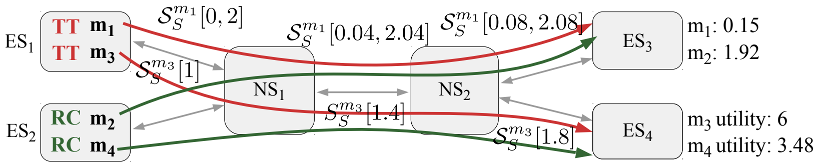


(c) Modify RC VL: *BAG* and L_{max} are doubled; *Cost* $= -1.89$; tabu

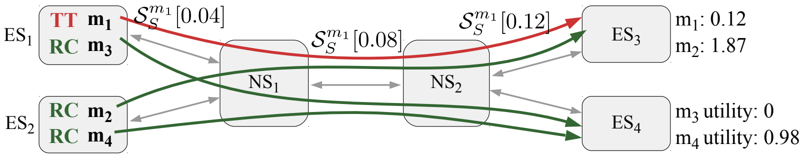


(d) Switch Traffic Class of m_1 from TT to RC; *Cost* $= -2.62$; non-tabu

Backup: Optimization Strategy 3



(e) Switch Traffic Class of m_3 from RC to TT; $Cost = 9.48$; non-tabu



(f) Modify Schedule of m_1 on $ES_1 - NS_1$ by postponing it with 0.04 ms; $Cost = 0.98$; non-tabu