A unifying view of Loosely Time-Triggered Architectures

Albert Benveniste (Inria-Rennes), Paul Caspi (Verimag)
Anne Bouillard (ENS)
Alberto Sangiovanni-Vincentelli
Stavros Tripakis, and Claudio Pinello

EMSOFT 2010 — revisited 2012
Motivation

Loosely Timed-Triggered Architecture

Back Pressure LTFA

Time-based LTFA

Performances and comparison

Conclusion
Motivation

Moving from Federated to Integrated Architectures:

- aeronautics: IMA
- automobile: AUTOSAR
Motivation

Moving from Federated to Integrated Architectures:
- aeronautics: IMA
- automobile: AUTOSAR

Model-based design processes:
- Platform-based Design [A. Sangiovanni-Vincentelli]
- COC tool at PSA; and more
Motivation

Moving from Federated to Integrated Architectures:
- aeronautics: IMA
- automobile: AUTOSAR

Model-based design processes:
- Platform-based Design [A. Sangiovanni-Vincentelli]
- COC tool at PSA; and more

Models of Computation and Communication (MoCC)
- for the functions (synchronous, Kahn Networks, . . .)
  with corresponding formalisms
- for the architectures
**Motivation**

TTA (Time-Triggered Architecture) [Kopetz]

A comprehensive MoCC-based architecture:

- strong synchrony
- global discrete notion of time
- time-based fault-tolerance
- time-based scheduling (TDMA)
- time-based interfaces

Resistances to TTA:

- cost of synchronization
- rigidity of TDMA
- cost of re-design (adaptations & upgrades)
Motivation

TTA (Time-Triggered Architecture) [Kopetz]
A comprehensive MoCC-based architecture:
- strong synchrony
- global discrete notion of time
- time-based fault-tolerance
- time-based scheduling (TDMA)
- time-based interfaces

Resistances to TTA:
- cost of synchronization
- rigidity of TDMA
- cost of re-design (adaptations & upgrades)
Motivation

Computers on trains for speed control

Computers on tracks for collision avoidance and to avoid losing a train (ghost train!!)

MBPC

Wired communications for fixed computers

For computers on trains: use wheels or wireless

Communication by Sampling (LTTA)
Motivation

**AFDX technology – Addressing: MAC, IP, UDP**

- Avionics communications are based on multicast:
  - one transmitter
  - one or several receivers
- Asynchrony of individual clocks
- No reconfiguration capability in the AFDX network

Alt = 10 000 ft

Benveniste et al. ( ) A unifying view of Loosely Time-Trigged Architectures
Motivation

⇒ Relax TTA to LT TA

(Loosely Time-Triggered Architecture)
Loosely Timed-Triggered Architecture

1 Motivation

2 Loosely Timed-Triggered Architecture

3 Back Pressure LTTA

4 Time-based LTTA

5 Performances and comparison

6 Conclusion
Loosely Time-Triggered Architecture

Communication by Sampling (CbS)

1. Communication medium \( \sim \) set of shared memories, 1 per variable
2. Each computer periodically samples its external world
   And so does the communication medium itself
Loosely Time-Triggered Architecture

Communication by Sampling (CbS)

1. Communication medium \( \sim \) set of shared memories, 1 per variable
2. Each computer periodically samples its external world
   And so does the communication medium itself

Advantages:

- communication medium off-the-shelf;
- autonomy, no deadlock, no livelock;

Results, however, in losses and duplications.
Loosely Time-Triggered Architecture

Problems when writing/sensing with non synchronized clocks:

- duplications
- losses
Loosely Time-Triggered Architecture

Problems when writing/sensing with non synchronized clocks:

no harm so far for continuous feedback control

RT-Builder [Geensys→DS]
JitterBug/TrueTime [Arzen]
Problems when writing/sensing multiple discrete signals:

Cases 1 and 2 correspond to two different outcomes for the local clock of $A_1$. 

Benveniste et al. () A unifying view of Loosely Time-Triggered Architectures
Preservation of the semantics: setting the problem

Flow equivalence ensured by a special *LTTA protocol*.

Two approaches:
- building on *back-pressure* and *elastic circuits*
- building on *time*
1 Motivation

2 Loosely Timed-Triggered Architecture

3 Back Pressure LTTA

4 Time-based LTTA

5 Performances and comparison

6 Conclusion
Back-Pressure LT TA

**top:** Node as a Net
reads and writes alternate

**bottom:** Link as a Net
1-buffer on each link

$\mathcal{N}_i$: directed link $j \rightarrow i$
dashed: back-pressure

$\mathcal{N}_{ji} = \left( \prod_i \tilde{\mathcal{N}}_i \right) \times \left( \prod_{j \rightarrow i} \mathcal{N}_{ji} \right)$

elastic circuit (BP-EC)
Back-Pressure LT TA

Skipping mechanism at node $i$ triggered by the clock of node $i$

$\tilde{N}_i$: directed link $j \rightarrow i$
dashed: back-pressure

$$\prod_i \tilde{N}_i \times \prod_{j \rightarrow i} N_{ji}$$

elastic circuit (BP-EC)

$$\left( \prod_i N_i \right) \times \left( \prod_{j \rightarrow i} N_{ji} \right)$$

BP-LTTA
1 Motivation

2 Loosely Timed-Triggered Architecture

3 Back Pressure LTTA

4 Time-based LTTA

5 Performances and comparison

6 Conclusion
**Time-based LTTA: difficulty**

\[ M_{ji} = \]

\[ N_{ji} = \]

Time-based, pure CbS link (top), is compared to BP-link (bottom)

Observe the read arc and lack of synchronization that results

To recover synchronization:

- slow-down by synchronizing on *local* physical time
- accelerate using a token-based *publication* mechanism
Time-based LT TA: general principle

**Aim:** ensure a clean alternation of writing and reading phases throughout the entire architecture.

To recover synchronization:
- slow-down by synchronizing on local physical time
- accelerate using a token-based *publication* mechanism
Time-based LT TA: distributed protocol (2 nodes)

wait 2 between writing and reading

\[ r_i \rightarrow w_i^1 \rightarrow w_i^2 \]

writing and reading

wait 2 between

Publications by the other node

\[ w_j^1 \rightarrow \cdots \rightarrow j \rightarrow w_j^q \]
Time-based LTDA: distributed protocol (2 nodes)

wait 2 between writing and reading

wait before writing and publishing

Benveniste et al. () A unifying view of Loosely Time-Triggered Architectures
Time-based LT TA: distributed protocol (2 nodes)

wait 2 between writing and reading

\[ r_i, \ w^1_i, \ w^2_i \]

synchronize writings

\[ w^1_i, \ w^2_i, \ r_i \]

wait before writing and publishing

\[ w^1_i, \ w^2_i \]

on publications

publications by other node

\[ w^q_j, \ r_j, \ \Pi_1, \ \Pi_2 \]
Time-based LTTA: distributed protocol (general)

\[ p - 1 \text{ transitions} \]

\[ q - 1 \text{ transitions} \]
Motivation

Loosely Timed-Triggered Architecture

Back Pressure LTTA

Time-based LTTA

Performances and comparison

Conclusion
Performances and comparison

**Back-pressure LTTA performance**

- Inter-tick time of local clocks: $T_{\text{min}} \leq \kappa_{i,k+1}^i - \kappa_{i,k}^i \leq T_{\text{max}}$
- Communication delays: $\tau \leq \tau_{\text{max}}$.

**Back-Pressure LTTA**

Elastic circuit: $\lambda_{\tilde{N}} = \frac{1}{2(\tilde{T}_{\text{max}} + \tilde{\tau}_{\text{max}})}$
Back-pressure LTTA performance

- Inter-tick time of local clocks: \( T_{\text{min}} \leq \kappa_{k+1}^i - \kappa_k^i \leq T_{\text{max}} \)
- Communication delays: \( \tau \leq \tau_{\text{max}} \).

**Back-Pressure LTTA**

**Elastic circuit:** \( \lambda_{\tilde{N}} = \frac{1}{2(\bar{T}_{\text{max}} + \bar{\tau}_{\text{max}})} \)

With skipping mechanism:

- \( T_{\text{max}} = \bar{T}_{\text{max}} \)
- \( \tau_{\text{max}} = \tau_{\text{max}} + T_{\text{max}} \)

\[ \lambda_N = \frac{1}{4 \bar{T}_{\text{max}} + 2\bar{\tau}_{\text{max}}} \]
Performances and comparison

**Time-based LTTE performance**

**Tuning \( p \) and \( q \)**

**Rd** The \( k \)-th firing of transition \( r_i \) occurs only after all places \( \Pi_j \) have been written \( k - 1 \) times.

**Wr** The \( k \)th firing of one of the (conflicting) transitions \( w_i^1 \ldots w_i^q \) occurs only after transitions \( r_j, j = 1 \ldots n \) have all been fired \( k \) times.
Performances and comparison

Time-based LT-TTA performance

Theorem

These conditions on $p$, $q$ ensure $\hat{L}_M = L_{\tilde{N}}$:

\[
\begin{align*}
 p & \geq \frac{\tau_{\text{max}}}{T_{\text{min}}} + \frac{T_{\text{max}}}{T_{\text{min}}} \\
 q & \geq \frac{\tau_{\text{max}}}{T_{\text{min}}} + \frac{T_{\text{max}}}{T_{\text{min}}} + p \left( \frac{T_{\text{max}}}{T_{\text{min}}} - 1 \right)
\end{align*}
\]

Theorem

Worst case throughput ($p_\star$ and $q_\star$ optimal):

\[
\lambda_M = \frac{1}{(p_\star + q_\star) T_{\text{max}}}
\]
Performances and comparison

**Time-based LTTA performance**

**Theorem**

These conditions on $p, q$ ensure $\hat{L}_M = L_{\tilde{N}}$:

$$p \geq \frac{\tau_{\text{max}}}{T_{\text{min}}} + \frac{T_{\text{max}}}{T_{\text{min}}}$$

$$q \geq \frac{\tau_{\text{max}}}{T_{\text{min}}} + \frac{T_{\text{max}}}{T_{\text{min}}} + p \left( \frac{T_{\text{max}}}{T_{\text{min}}} - 1 \right)$$

**Theorem**

Worst case throughput ($p_\star$ and $q_\star$ optimal):

$$\lambda_M = \frac{1}{(p_\star + q_\star)T_{\text{max}}}$$

$t$: date of the first $(k-1)$th writing

$tw \leq \min(t + qT_{\text{max}}, t + \tau_{\text{max}} + T_{\text{max}})$

$t_r \geq t + pT_{\text{min}}$

$pT_{\text{min}} \geq \tau_{\text{max}} + T_{\text{max}}$

$\min(qT_{\text{max}}, \tau_{\text{max}} + T_{\text{max}}) + pT_{\text{max}} \leq \tau_{\text{max}} + (p + 1)T_{\text{max}}$
Comparison regarding throughput

- **BP-LTTA**: lower bound for throughput is

  \[ \frac{1}{\lambda_N} = 4T_{\text{max}} + 2\tau_{\text{max}} \]

- **Time-LTTA** when delay and jitter small relative to nominal period
  
  The lower bound for the throughput is

  \[ \frac{1}{\lambda_M} \approx 4T_{\text{max}} \]

- **Time-LTTA** for distant communications, i.e.,

  \[ \frac{T_{\text{max}}}{T_{\text{min}}} \approx 1 \quad \text{and} \quad \frac{\tau_{\text{max}}}{T_{\text{min}}} \gg 1 \]

  The lower bound for the throughput is

  \[ \frac{1}{\lambda_M} \approx 2 \left( \frac{T_{\text{max}}}{T_{\text{min}}} \right)^2 \tau_{\text{max}} \approx \frac{1}{\lambda_N} \]
Conclusion

1 Motivation

2 Loosely Timed-Triggered Architecture

3 Back Pressure LTTA

4 Time-based LTTA

5 Performances and comparison

6 Conclusion
Conclusion

- Relaxing TTA to LT TA while preserving MoCC
- Back-Pressure LT TA & Time-Based LT TA
- Similar performances w.r.t. throughput
- BP-LT TA more flexible & TB-LT TA more robust

- Blending the two (see paper)

- Services must be studied:
  - fault tolerance
  - scheduling
  - interfacing