A protocol for Mixed-Criticality management in Switched Ethernet networks

O.Cros, X.Li, L.George

December 1, 2015

What's the matter ?

Mixed-Criticality (MC) in network context

A protocol to manage MC

Delay computation

Simulation results

Plan and objectives

MC in network context

A protocol to manage MC

Delay computation

Simulation results

Context

Domains

Public transport (CAN) (Volvo, Renault, ...)
Avionics (AFDX) (Airbus)
Home automation
Defense

Context

Domains

Public transport (CAN) (Volvo, Renault, ...)
Avionics (AFDX) (Airbus)
Home automation
Defense

Principles

Message routing and scheduling
Classifying messages by importance

Context

Domains

Public transport (CAN) (Volvo, Renault, ...)
Avionics (AFDX) (Airbus)
Home automation
Defense

Principles

Message routing and scheduling
Classifying messages by importance

MC

Privileging messages in critical situation

- Critical for the vehicle, for the mission, for the users
- Overload context
- Assuring critical messages transmission

Why Mixed-Criticality ?

Today

1 network per group of functions (mechanical, comfort, gps tracking, ...)
Increasing of financial costs, weight, fuel and energy consumption
Example : 3/4 different antennas per public bus

Mixed-criticality

Mixing all the functions in the same network
Each function associated to a criticality level
MC management protocol to guarantee critical messages transmission

Mixed-Criticality

Related work

Mono/Multicore context
2-levels of criticality
QoS

Synchronization protocol

Ethernet IEEE-1588
PTPv2

Mixed-criticality

Problems

How to manage messages scheduling inside a embedded network ?
How to assure critical messages transmission ?

Mixed-criticality

Problems

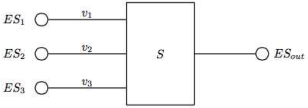
How to manage messages scheduling inside a embedded network ?
How to assure critical messages transmission ?

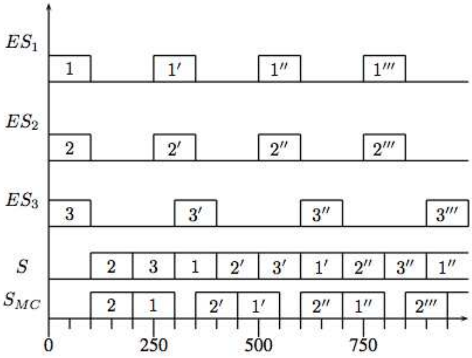
Solution

Providing MC management in embedded networks
Period-oriented or WCTT-oriented
Static-defined path

Example

Flow	T_i^{LO} (μs)	T_i^{HI} (μs)	C_i (μs)	u_i^{LO}	u_i^{HI}
v_1	500	250	100	0.2	0.4
v_2	500	250	100	0.2	0.4
v_3	300	-	100	0.33	-



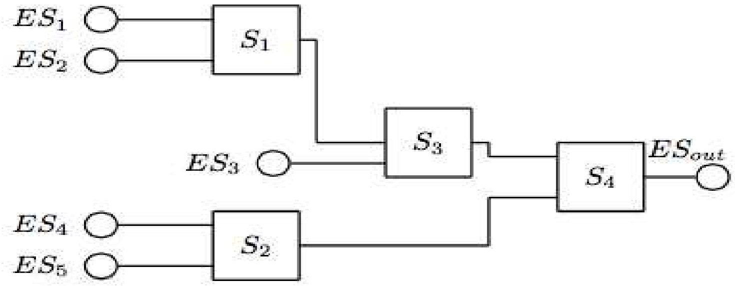


Topology

Centralized topology

Automotive Ethernet, AFDX targets
One central node to store criticality information

Example



Plan and objectives

MC in network context

A protocol to manage MC

Delay computation

Simulation results

A two-phase protocol

First phase

Switch-criticality call
Transmitting switch order to a central node
Centralized topology

Second phase

Multicast the switch criticality order (reliable multicast)
Sending new criticality info to all nodes
Reliable (deterministic) multicast

A two-phase protocol : The call phase

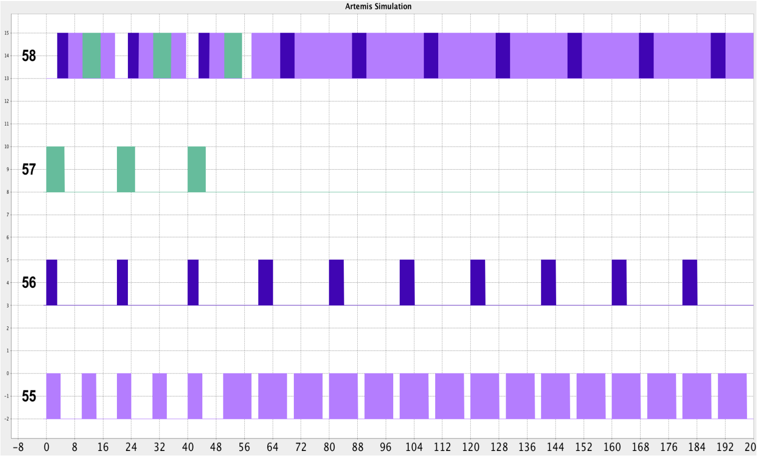
Principle

Triggering a criticality switch when :
Longer period detected Or shorter inter-arrival time detected
Fixed WCTT C_c (static known size)
Ethernet protocol

Transmission

Dedicated VLAN, highest priority
PTP messages : clock accuracy (PTP-ETE)

MC management protocol



A two-phase protocol : the multicast phase

Multicast

Each node can get the switch criticality order at a different instant
Clock precision and clock synchronization (PTP - IEEE1588)
Total order : coherency in the network
At each instant, all nodes have the same criticality information (precision ϵ)

Reliability

Each single physical link is bounded
Clock accuracy ϵ
Worst-case delay computation

Switch-criticality order

All nodes switch at the same time
Last reception instant : $\max_{n \in \mathcal{N}}(d_n * (C_c + sl) + \epsilon_n)$

Plan and objectives

MC in network context

A protocol to manage MC

Delay computation

Simulation results

The trajectory approach

Flows

Representing the network as a set of flows

Each flow v_i emits messages

$v_i = \{\mathcal{P}_i, C_i, \vec{T}_i\}$

Criticality management

One period per criticality level

$\vec{T}_i = \{T_i^{LO}, T_i^{HI}\}$

Call phase delay

Principle

Sending criticality switch call to central node
Emitting a call when a message exceeds its period or LO-WCTT
Highest priority VLAN

Delay

$$I^{n}_{delay} = F_{PTP} * \sum_{\substack{j \in hp_c \\ \mathcal{P}_c \cap \mathcal{P}_j \neq \emptyset}} \left(S^{first_{c,j}}_{max_c} - M^{first_{c,j}}_{c} + A_{c,j} \right) + \sum_{h \in \mathcal{P}_c} \delta^{h}_{c} + (|\mathcal{P}_n| - 1) * (sl + 2 * C_c)$$

Multicast phase delay

Principle

Sending the criticality level information to all nodes
Depending on the size of the network

Delay

$$M^{n}_{delay} = \quad d_n * (C_c + sl) + \epsilon_n$$

Total delay

Phases delay

$$S_{delay} = \max_{n \in \mathcal{N}} (I^n_{delay} + M^n_{delay})$$

Final expression

$$\begin{aligned} S_{delay} &= F_{PTP} * \sum_{\substack{j \in hp_c \\ \mathcal{P}_c \cap \mathcal{P}_j \neq \emptyset}} \left(S^{first_{c,j}}_{max_c} - M^{first_{c,j}}_{c} + A_{c,j} \right) \\ &+ \sum_{h \in \mathcal{P}_c} \delta^h_c \\ &+ (2 * \max_{n \in \mathcal{N}}(d_n) - 1) * (C_c + sl) + C_c(\max_{n \in \mathcal{N}}(d_n) - 1) + \epsilon \end{aligned}$$

Plan and objectives

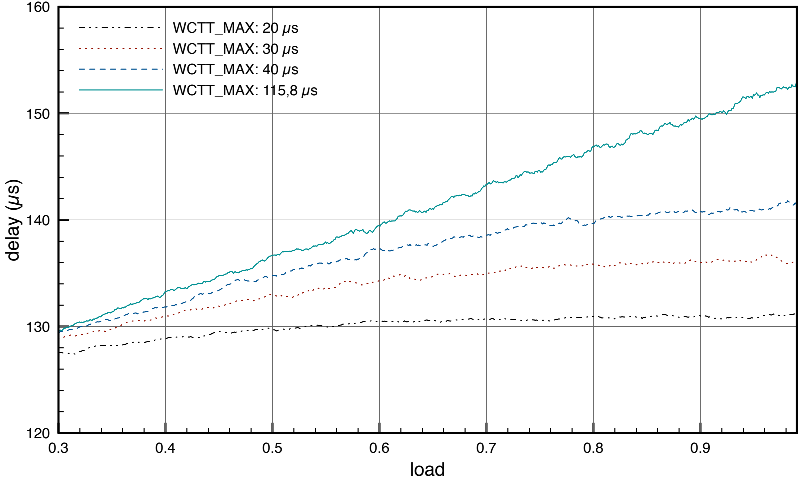
MC in network context

A protocol to manage MC

Delay computation

Simulation results

Criticality switch delay



Criticality switch delay

- Highest priority (except PTP)
- Non-preemptive effect
- Switch criticality delay stays constant
- Criticality messages transmission is guaranteed in a bounded time

Conclusion

- MC management protocol
- Reliable multicast
- Independent from the load

Perspectives

Delay computation on switch-criticality delay
Delay computation to return to low-criticality mode
Uncentralized MC management

Conclusion

Thanks

Thanks to all the authors of this presentation and publication
Thanks to our respective labs
Thank you for your attention !

?

Questions ? Feel free to ask !