Temporal Mechanisms for Communications in Real-Time Networks

Pascal LORENZ
University of Haute Alsace
34, rue du Grillenbreit 68008 Colmar - France
e-mail: lorenz@ieee.org

***Abstract*—Some processes should respect real-time constraints to be valid. Temporal mechanisms should indicate whether time constraints are met or not, and when they are not, these mechanisms should identify the causes of their non-compliance. We present some mechanisms enabling to indicate whether the time constraints were met in real-time and time-critical systems.**

Keywords-Temporal constraints, Time Windows, Real-Time, Modelization, Communication.

I. INTRODUCTION

In real-time environments, the processes consistencies depend of the obtained results but also of the respect of temporal constraints. This imposes that the start and/or the end of execution respect the timing constraints. Systems which must meet strict time constraints to prevent disasters are called time-critical systems [1, 4, 5].

Thus, for example, we should be able to know, why in an industrial plant, when exceeding a critical threshold temperature detected by a sensor at the time t1, an actuator has not triggered the closure of a valve at time t1 + □t as it was intended in the specification. Is this due to the fact that the sensor has not transmitted information within the time limit or this is due to network congestion or this is due to actuator that has not taken into account the information received on time ?

This paper is structured as follows: in section II, we describe the specification of time-windows constraints. Section III gives a detailed description of validity of a variable in a time-window. Section IV presents a modelization of the temporal mechanisms. Conclusion and future work are given in section V.

II. SPECIFICATION OF TIME-WINDOWS CONSTRAINTS

Communication delays in the networks are generally not deterministic, and therefore the data arriving at a consumer may be outdated, since a time-critical data have a lifetime (or validity) determined by the application [6, 7]. The data lifetime for the producer (respectively consumer) is the time interval during which the value produced (respectively consumed) remains valid and is valid for the application. Thus, data generated are valid and make sense for a certain period of time after their production.

We have chosen to represent the time involved during the communication steps in terms of time constraints. We have introduced mechanisms to know if during the communication, some data has been produced, transmitted and consumed in a timely and otherwise, they allow locating the causes of non-compliance with time constraints.

Different types of operators allow taking into account time constraints. Time constraints at the earliest date and latest date can be represented in terms of time intervals. Time window mechanisms take into account time constraints at the earliest date and latest date, and time constraints are expressed as the maximum time for a communication between entities [3].

The various possible combinations between two times intervals are described in the diagram below:

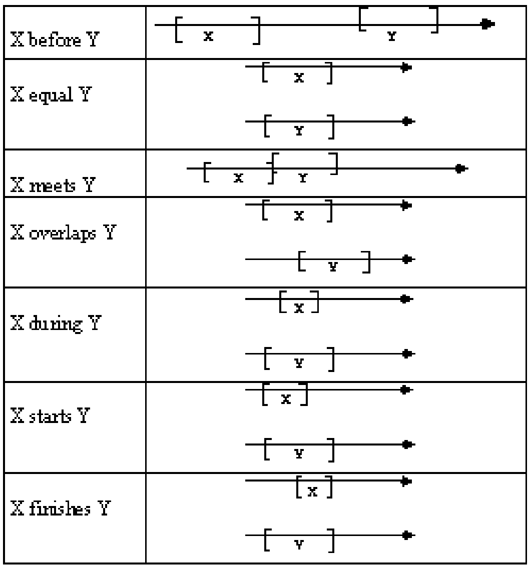

Fig. 1: Relationships between time intervals

We always assume that local clocks entities of producing and consuming entities are synchronized and that the time difference between all the entities producing and consuming is negligible compared to the nature of the application. Thus, beginnings and ends of time windows will be fixed by using physical clocks synchronized [2, 8, 9].

A time window, denoted TW thereafter, has three parameters:
- start time,
- ending time and
- duration.

These three parameters can be preset or not.

↑ represents the start time, ↓ the end and Δ the length (or duration) of the time window. Let x be a time window, if all parameters are known, then one can verify that ↑x + Δx = ↓x.

We will use the parameters i and v to represent the i^{th} time window of activity used for the transmission of a variable v.

For each new production, transmission, reception or consumption of a given variable v, a new time window is used by incrementing the index i of the TW.

A time window can be described as follows:

x y (v, i) with x = (↑, ↓ or Δ) and
y = (prod, send, rec or cons).

We will now present a set of rules that must always be taken into account to allow a consistent sequencing of the various TW.

The fact that the start of the TW is always prior to the end of the same TW (production, transmission, reception or consumption) can be represented by the following rule R1:

R1: ↑y (v, i) < ↓y (v, i)

If we apply R1 to the different transmission sequences, we obtain the following rules R2 and R3:

R2: ↑prod (v, i) < ↑send (v, i) < ↑rec (v, i) < ↑cons (v, i)

R3: ↓prod (v, i) < ↓send (v, i) < ↓rec (v, i) < ↓cons (v, i)

If we have only earliest constraints date, there is no need to chain the various TW, since it is only enough to start on time.

The sequencing of the various TW can be done by using either fixed or slippery TW.

Fixed TW allows managing earlier and latest constraints dates. On the other side, slippery TW can only manage latest constraints dates.

These can be defined as follows:

- Fixed TW: early and latest time intervals are fixed once and for all. Thus, for a production entity, if a production is done more or less early in the time slot allocated, it does not affect the start date for sending the variable. Similarly, in the consumer entity, the fact that a variable is received earlier or later has no effect for the start of the consumption of this received variable.

- Slippery TW: there is no time constraint on the start of different time intervals involved in the communication. Thus, as soon as data is produced within the constraints of time, it offers the possibility of the sending this later. Similarly, as soon as a data is received within the time constraints, the consumer can consume this later.

Fixed TW can be described as follows:

[↑prod (v, i), ↓prod (v, i)] before [↑send (v, i), ↓send (v, i)]
[↑send (v, i), ↓send (v, i)] overlaps [↑rec (v, i), ↓ rec (v, i)]
[↑rec (v, i), ↓rec (v, i)] before [↑cons (v, i), ↓cons (v, i)]

The slippery TW can be described as follows:

[↑prod (v, i), ↓prod (v, i)] overlaps [↑send (v, i), ↓send (v, i)]
[↑send (v, i), ↓send (v, i)] overlaps [↑rec (v, i), ↓rec (v, i)]
[↑rec (v, i), ↓rec (v, i)] overlaps [↑cons (v, i), ↓cons (v, i)]

Thus for fixed TW, the following rules R4 and R5 indicate the links between the communication steps:

R4: ↓prod (v, i) <↑send (v, i)
R5: ↓rec (v, i) <↑cons (v, i)

The different cases of sequence for fixed TW and sliding TW can be described as follows:

- Sequence of sliding windows = R1 + R2 + R3
- Sequence of fixed windows = R1 + R2 + R3 + R4 + R5
- Sequence of fixed windows at the producer and the sliding window at the consumer = R1 + R2 + R3 + R4
- A series of sliding windows at the producer and fixed windows at the consumer = R1 + R2 + R3 + R5

↑y (v, i +1) must be always higher than the beginning of the previous TW, represented by the rule following R6:

R6: ↑y (v, i +1) > ↑y (v, i)

III. TIME-WINDOW FOR THE VALIDITY OF A VARIABLE

A variable produced at a given moment has duration of life that varies depending on the consumers. Thus in the case of an industrial application, for an application process that makes statistics, some data can be considered valid with an infinite duration. On the other side, for an application process associated with an actuator operating in a real time environment, this same data will be, for example, valid for only 20 ms. Therefore, the notion of data lifetime depends on the use made of it.

To represent the data's lifetime, we will again use the time window mechanism. We will now try to formalize the concept of lifetime of a variable in the same way as we did for the operations of production, transmission, reception and consumption. A time window for the lifetime (or the validity) of a variable will be denoted VW.

We will use the same notation ↑, ↓ and Δ used by the TW. The parameters v and i will be used to represent, respectively, the i^{th} VW instantiated for the production of a variable v in a TW.

VW is always associated with a given TW, since the beginning of a VW takes always place in a production TW. Moreover, it is important to note that it is possible to have multiple productions of a given variable even within a single production TW.

When a variable is produced, it has a lifetime, denoted Δ, and it has a lifetime Δ' (with Δ'<Δ) when it is received by the consumers.

A time window on the validity of a variable is described as follows:

x val (v, i) with x = (↑, ↓ or Δ)

↑val (v, i) represents the moment when the variable v is produced and ↓val (v, i) represents the end of the validity of the variable v.

The various parameters of VW should generally be defined by the user. As for the TW, the beginning of a VW is always prior to the end of this VW. This translates into the following rule R7:

R7: ↑val (v, i) < ↓val (v, i)

As the production of a variable is always at the beginning of VW, it must necessarily be in the production TW since the purpose of a production TW is to verify that the data has been produced on time. This case is represented as follows:

R8: ↑prod (v, i) <↑ val(v, i) <↓prod (v, i)

For the end of the VW, the two main cases can appear:

- the end of the VW appears before the end of the TW consumption
- the end of the VW appears after the end of the TW consumption.

When the end of the VW occurs before the end of the consumption TW is represented as follows:

[↑val (v, i), ↓val (v, i)] finishes [↑prod (v, i), ↓cons (v, i)]

which is a special case of:

[↑val (v, i), ↓val (v, i)] during [↑prod (v, i), ↓cons (v, i)]

When the end of the VW occurs after the end of the consumption TW is represented as follows:

[↑prod (v, i), ↓cons (v, i)] overlaps [↑val (v, i), ↓val (v, i)]

Depending on the variable's lifetime, it is possible to transmit this information sooner or later, while issuing a valid data to the consumer entity.

Similarly in the consumer entity, depending on the variable's lifetime, it is possible to consume this data sooner or later while using valid data.

As the lifetime of a variable should allow to the production and consumption entities to know at any time if the value of a given variable is valid or not, it is necessary to introduce two cases:

- in the first case, the lifetime of a variable is known only locally by each consumption entities. Then only ↓val (v, i) enable to know when a certain variable is no longer valid.

- in a second case, the lifetime of a variable is set globally once and for all, then the consumption entities should calculate the remaining validity when it receives the variable.

We realize the importance of the clocks synchronization. If the local clocks of the producer and the consumer are not synchronized, then the notion of validity cannot be taken into account by the producing and consuming entities.

In the case where a single variable is used by several consuming entities, if the different local clocks are not synchronized, then the validity of this variable will expire at different times in different consuming entities.

IV. MODELISATION OF TEMPORAL MECHANISMS

At a given time, it is important to know if the current state is located inside or outside a TW and a VW; and to know if VW (v, i-1) exists or not.

The different possible situations for a TW and a VW are:

- located out of a TW, out of a VW and it does not exist a previous VW for a given variable v,
- located in a TW, out of a VW and it does not exist a previous VW for a variable v,
- located in a TW and in a VW; this state is called True state,
- located in a TW, out of a VW and it exists a previous VW for a variable v,
- located out of a TW and in a VW
- located out of a TW, out of a VW and it exists a previous VW for a variable v.

Table 1 describes the different possibilities with the previous proposed notation:

In TW (v, i)	In VW (v, i)	∃ a VW (v, i-1)	Name of the state
0	0	0	False
0	0	1	Expiration2
0	1	-	TW-Expiration1
1	0	0	Wait
1	0	1	VW-Expiration1
1	1	-	True

Table 1. Description of the different states

The symbols used in table 1 are:

0 = the condition is not verified
1 = the condition is verified.

Then the six possible combinations between the TW and VW are:

- 1 = False state: located out of a [Start Function (v, i), End Function (v, i)], out of a [Start VW (v, i), End VW (v, i)] and it does not exist a Start VW (v, i-1)

- 2 = Wait state: inside the a [Start Function (v, i), End Function (v, i)], out of the [Start VW (v, i), End VW (v, i)] and it does not exist a Start VW (v, i-1)

- 3 = True state: inside the [Start Function (v, i), End Function (v, i)], inside the [Start VW (v, i), End VW (v, i)], then at least one variable v has been produced respecting the timing constraints

- 4 = VW-Expiration1 state: inside a [Start Function (v, i), End Function (v, i)], out of a [Start VW (v, i), End VW (v, i)] and it exists a Start VW (v, i-1)

- 5 = TW-Expiration1 state: out of a [Start Function (v, i), End Function (v, i)], but inside [Start VW (v, i), End VW (v, i)]

- 6 = TW+VW-Expiration / Expiration2 state: out of the [Start Function (v, i), End Function (v, i)], out of a [Start VW (v, i), End VW (v, i)] and it exists a Start VW (v, i-1).

For each communication step, we will have a start event (generating an event to start the TW and the VW) and an end event (generating a time-out to close the TW and the VW).

The protocol integrating the 6 different states of the TW and VW is described in the following figure 2. The symbols used in figure 2 are:

: a state : an event

: an initial state : a test

action(s)

When the application start, the initial state is the False state.

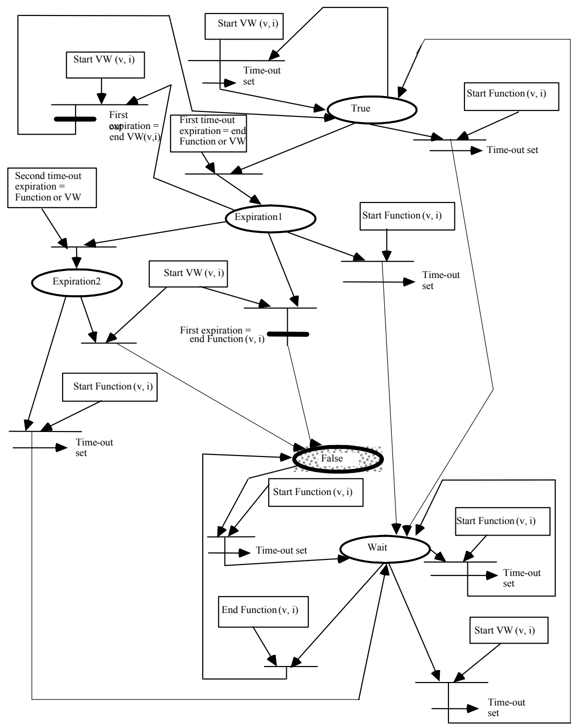

Fig. 2: Description of the protocol integrating Temporal Window and Variable validity time Window

V. Conclusion

We studied how time could be represented for real-time communication. We have introduced the concepts of time windows for production, transmission, reception and consumption, as well as concepts of time windows for the validity of variables in order to formalize the concepts of variables' lifetime.

We have proposed a set of rules to bind the different time windows and the time window for variables' validity.

In future work we will develop a prototype which supports the temporal mechanisms for real-time communication processes.

REFERENCES

[1] Brooke Shrader, Armen Babikyan, Nathaniel M. Jones, Thomas H. Shake, Andrew P. Worthen, "Rate Control for Network-Coded Multipath Relaying with Time-Varying Connectivity", IEEE Journal on Selected Areas in Communications, Vol. 29, no. 5, May 2011

[2] María José Pérez-Luque, Thomas D. C. Little, "A Temporal Reference Framework for Multimedia Synchronization", IEEE Journal on Selected Areas in Communications, Vol. 14, no. 1, January 1996, pp. 36-51

[3] Richard L. Schwartz, P. Michael Melliar-Smith, "From State Machines to Temporal Logic: Specification Methods for Protocol Standards", IEEE Transactions on Communications, Vol. 30, no. 12, December 1982, pp. 2486-2496

[4] Vijay Arya, N. G. Duffield, Darryl Veitch, "Temporal delay tomography", IEEE INFOCOM 2008 - 27th IEEE International Conference on Computer Communications, Vol. 27, no. 1, April 2008, pp. 495-503

[5] Chih-Jen Tseng, Chyou-Hwa Chen, "Exploiting the temporal dimension in QoS-aware multicast routing", ICC 2002 - IEEE International Conference on Communications, no. 1, April 2002, pp. 1284-1289

[6] Juan Jose Jaramillo, R. Srikant, Lei Ying, "Scheduling for Optimal Rate Allocation in Ad Hoc Networks With Heterogeneous Delay Constraints", IEEE Journal on Selected Areas in Communications, Vol. 29, no. 5, May 2011

[7] Ye Ge, Jennifer C. Hou, Hung-Ying Tyan, "A packet eligible time calculation mechanism for providing temporal QoS for multicast routing", ICC 1999 - IEEE International Conference on Communications, no. 1, June 1999, pp. 721-726

[8] Ahmet Feyzi Ates, Murat Bilgic, Senro Saito, Behcet Sarikaya, "Using Timed CSP for Specification Verification and Simulation of Multimedia Synchronization", IEEE Journal on Selected Areas in Communications, Vol. 14, no. 1, January 1996, pp. 126-137

[9] Ian F. Akyildiz, Wei Yen, "Multimedia Group Synchronization Protocols for Integrated Services Networks", IEEE Journal on Selected Areas in Communications, Vol. 14, no. 1, January 1996, pp. 162-173