Design and modeling techniques for real-time RTI time management
(11S-SIW-045)

Pierre Siron
DMIA Department,
Université de Toulouse, ISAE

Jean-Baptiste Chaudron
ONERA/DTIM/SER
ONERA, Centre de Toulouse

Eric Noulard
ONERA/DTIM/SER
ONERA, Centre de Toulouse

ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

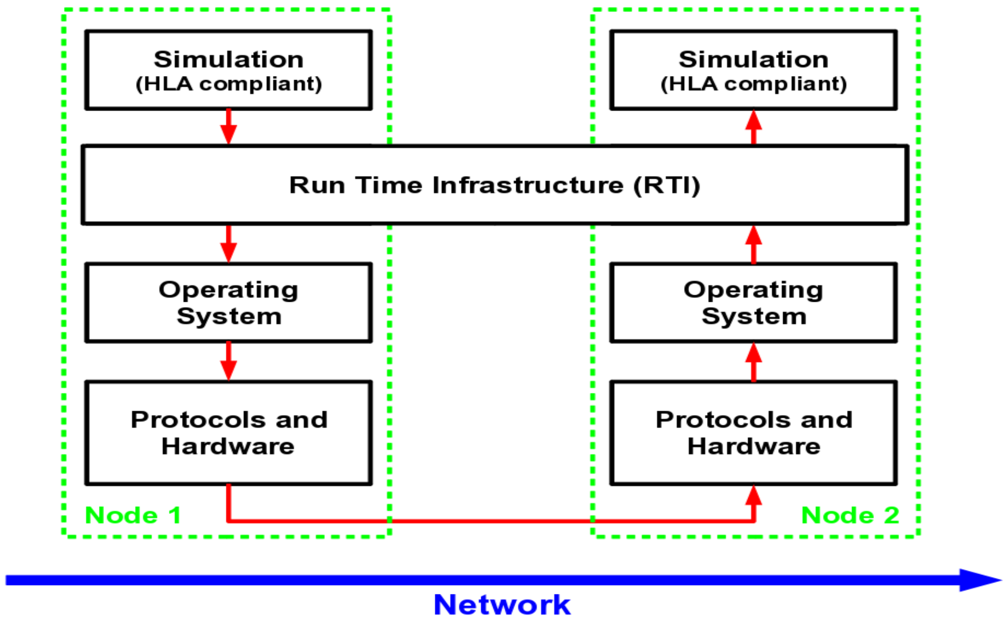
INTRODUCTION (1/5)

- Real Systems always respect *two principles* :
  - The *determinism principle* : the future of the system can be determined from its present state and its past:
    - At any time ***t***, there is an ***ε*** value for which the future behavior of the system at ***t + ε*** is exactly known.
  - The *causality principle* : the future never influences the past:
    - The system state at time ***t*** is independent of anything that may occur at a time ***t'*** greater than ***t***.
  - ***Any simulation of a real system have to ensure both principles***.

- Distributed Event Driven Simulation
  - A distributed simulation system consists of ***different autonomous computers*** that communicate through ***a global (or local) network***;
  - Simulators located on different computers interact with each other in order to achieve a *global common goal*:
    - Every simulator must determine the next instant, in the simulated time, which will produce a state change in the whole system.

INTRODUCTION (2/5)

- Middleware Level
  - Development of standards (CORBA, RPC,...) to consistently face problems involved by distribution (heterogeneous computers, network protocols):
    - *HLA standard* for distributed simulations (1.3 / IEEE 1516 / Evolved).
- ***Middleware*** in computing terms is used to describe a software agent acting as an ***intermediary*** between different distributed processes:
    - *Run Time Infrastructure (RTI)* is the HLA compliant middleware.



INTRODUCTION (3/5)

• CERTI Middleware

•*Open Source RTI* managed and maintained by Onera team (GPL, LGPL):

→ *ref:* ***09S-SIW-015***.

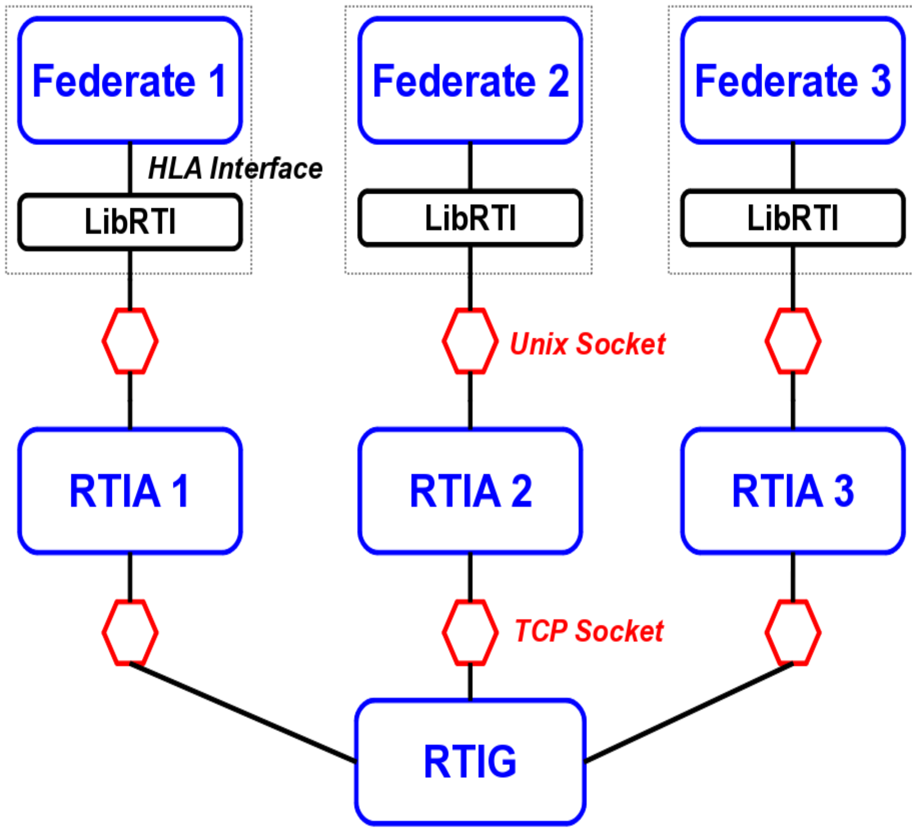
• Developed in ***C++***;
• Architecture of communicating processes;
• Implementation with ***TCP, UDP*** sockets;
• Available under ***Linux***, ***Unix*** and ***Windows*** operating systems.

• *Fully compliant* with 1.3 standard;
• *Not fully compliant* with IEEE 1516:

→ Work in progress.

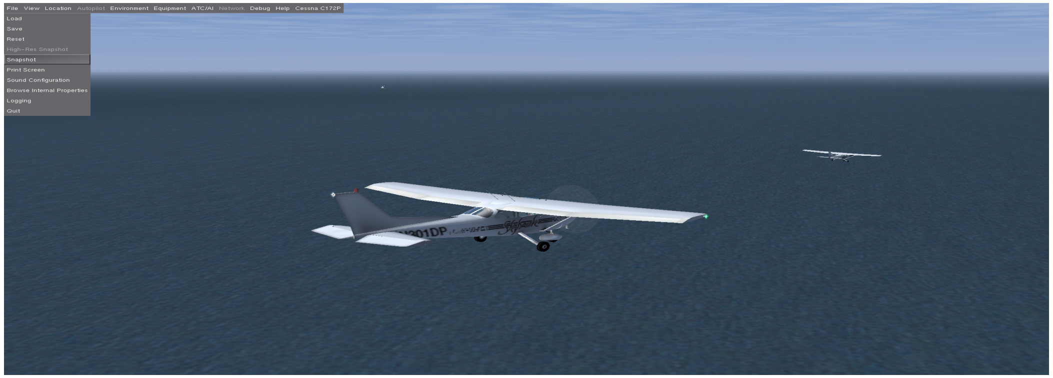
• Available at:

→ *http://pierre.siron.free.fr/certi.html*

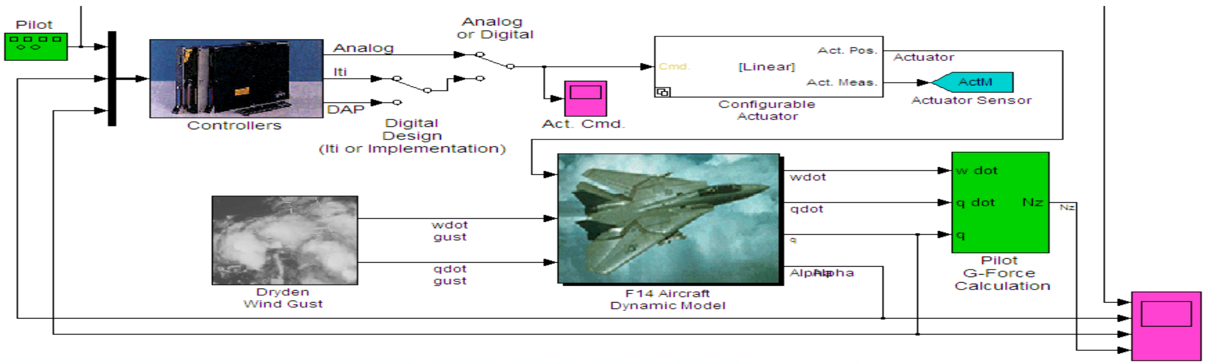


INTRODUCTION (4/5)

- Targeted Applications
  - Formation flying simulation (Xplane, Flight Gear, MS Flight Simulator,...)
    - ➔ Communication *between each simulator* with CERTI



  - Hardware-in-the-loop and embedded systems simulations
    - ➔ Connecting *sensors* and *actuators* with CERTI



INTRODUCTION (5/5)

- Our goal: *Using / Studying real-time properties with HLA standard*
  - To use HLA standard to allow communication between several distributed process w*ith timing constraints* (real time tasks);
  - To understand *weaknesses* and *strengths* of *time management techniques* for real time;
  - To propose solutions and techniques *to ensure determinism* of HLA time management.

- Plan

(1) ***Problematic & Backgroud***
- ***Global View***
- **Algorithms and Limitations**
- **HLA services concerned**

(2) ***TM for event driven RT federate***
- **NER, NERA and Time Creep**
- **A new Optimized Algorithm**
- **Illustration**

(3) ***TM for time driven RT federate***
- **Periodic Federates**
- **Metrics, Formulas**
- **Illustration**

- Time management mechanisms
  - One of the main benefits of this simulation standard HLA;
  - Allow a ***consistent global time*** throughout the simulation and ***to prevent causal anomalies*** effects;
  - Different kinds of approaches:
    - **Optimistic Strategy** (coherent-post):
      - *Virtual Time* (Jefferson).
    - **Conservative Strategy**:
      - Avoid the violation of the *local causality constraint* altogether;
      - Main interest of this work.

- Usefulness of Conservative Time Management for real time simulation ?
  - Ensure respect of deadlines;
  - Keep ***consistency*** between the different federates cycles during their execution.

- ***First Generation: NULL MESSAGE ALGORITHM [1979]***
  - Based on Chandy, Misra & Bryant original algorithm;
  - *Limitation for real-time:* Latency due to *null message exchange* between federates (depends on *lookahead* parameter).

- ***Second Generation: DISTRIBUTED SNAPSHOTS ALGORITHM [1993]***
  - Based on Mattern original algorithm;
  - *Limitation for real-time:* LBTS computation cannot generally be guaranteed to complete *within a bounded time* (Transient messages cause an LBTS computation to be aborted and retried).

- CERTI Implementation
  - Use ***NULL MESSAGE ALGORITHM*** algorithm;
  - Seems to have interesting behavior for real-time simulations;
  - Latency compensated by better synchronization.

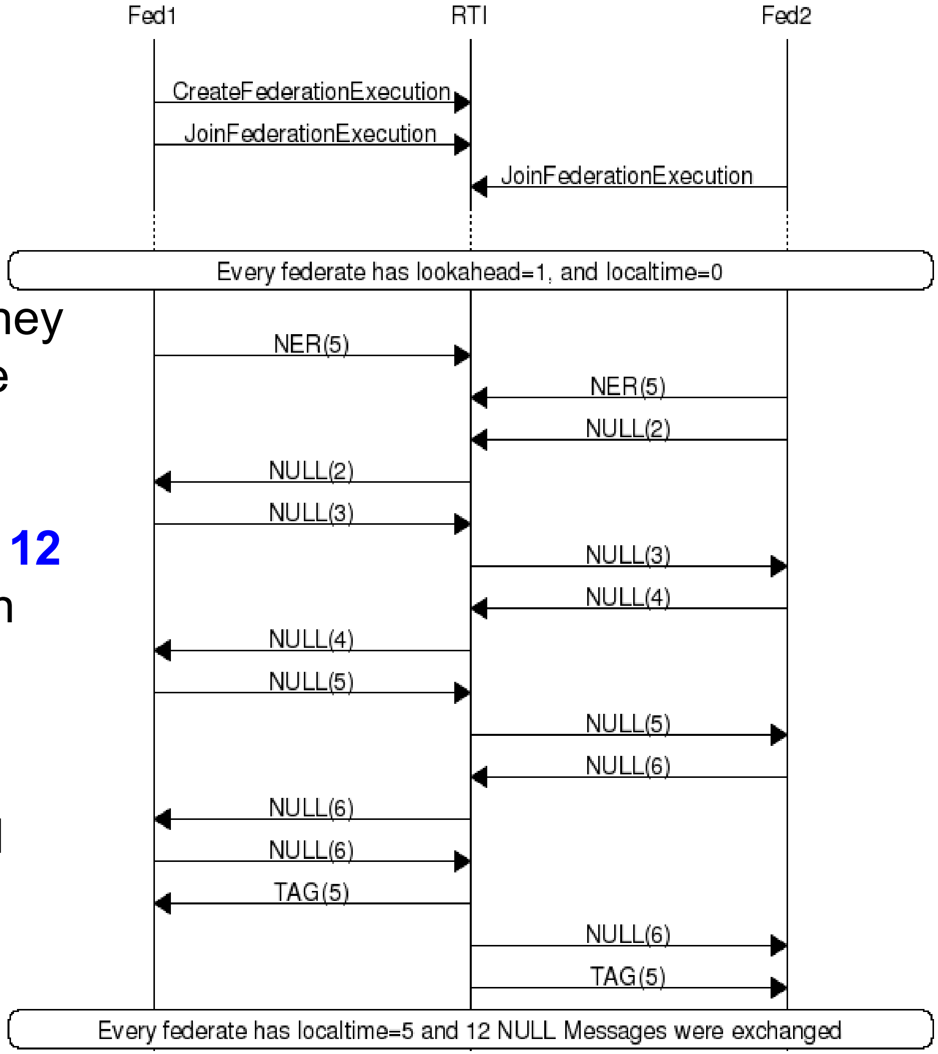
    → *ref:* ***08E-SIW-061***

- Time management HLA services concerned
  - Various services exist to allow the federate to express its requests for advancing its local logical time:
    - `timeAdvanceRequest()` (TAR);
    - `timeAdvanceRequestAvailable()` (TARA);
    - `nextEventRequest()` (NER);
    - `nextEventRequestAvailable()` (NERA);

- Type of federate concerned
  - TAR and TARA are devoted to federates which employ a *TIME-STEPPED mechanism*;
  - NER and NERA are devoted to federates which employ a *EVENT-DRIVEN mechanism*;
  - TARA and NERA are devoted to *zero-lookahead protocol*:
    - After **TAG(t)** messages with timestamp equal to **t** can still be delivered by the federate.

Time Creep Problem

- Two federates : Fed1 and Fed2 with lookahead=1 call the NER(5) service;
- They are alone in the federation so that they could theoretically advance their local time strait to instant t=5;
- Classical NULL message algorithm imply **12** null messages exchange for advance each federate;
- In several case, the number of Null Messages may become unacceptable and limits the performance of the simulation:
  - → *Lookahead Time Creep Problem.*



NULL MESSAGE PRIME ALGORITHM

- The idea of our NULL MESSAGE PRIME algorithm is to *take advantage of the RTIG* (CERTI CRC Central Run-Time Infrastructure Component);

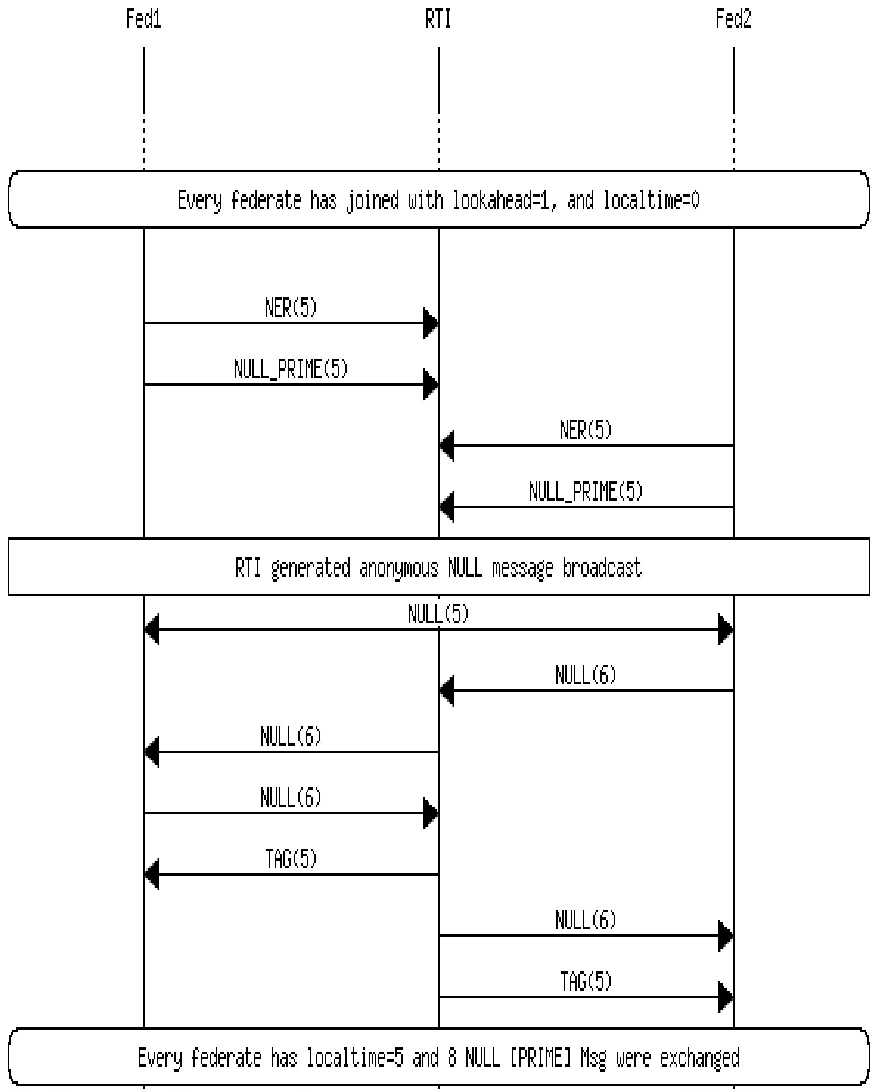
- In the classical NULL message algorithm : RTIG is only acting as a *pure gateway* and distributes the NULL messages to each concerned federate.

- **The new algorithm :**
  - When a federate is NERing it will send a NULL PRIME message to the RTIG;
  - RTIG computes an Federation-wide LBTS;
  - Whenever the RTI-LBTS strictly increases, the *RTIG* will *generate an anonymous NULL message* and *broadcast it to all time constrained federates*.

- The NULL PRIME Message algorithm co-exists with the classical NULL Message and the protocol is still valid when federate use TAR and NER services.

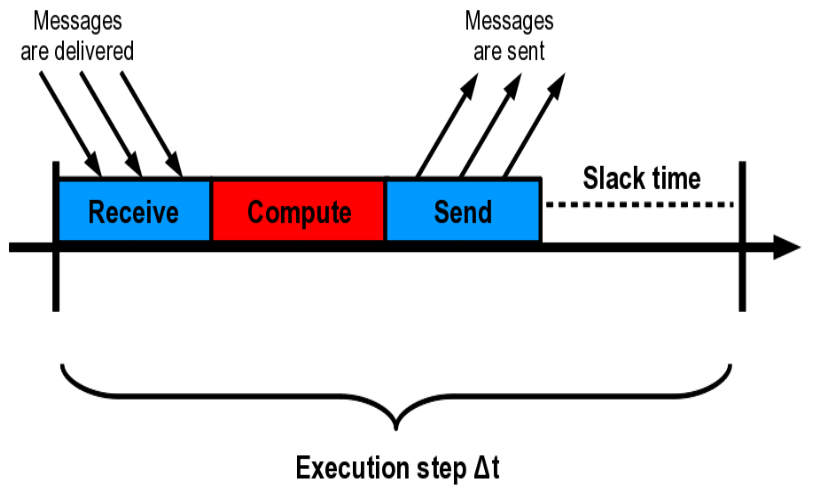
Illustration

- **In this case :**
  - the number of NULL message exchanged before TAG(5) is **8**;
  - In the original algorithm, it is **12**.
- The number of message generated by the algorithm is constant and independent from lookahead value (including zero lookahead).
- We think that the NULL PRIME Message algorithm is somehow equivalent to *global reduction* based algorithm like the one from *Mattern*.



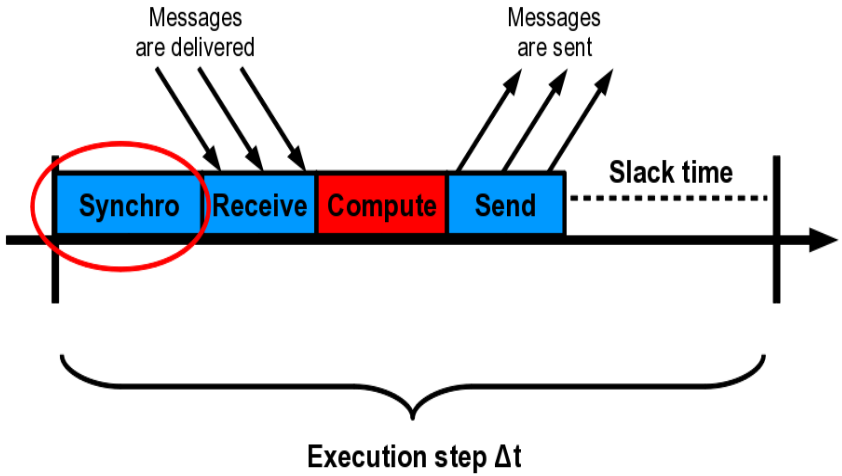
Repeatability within the simulations

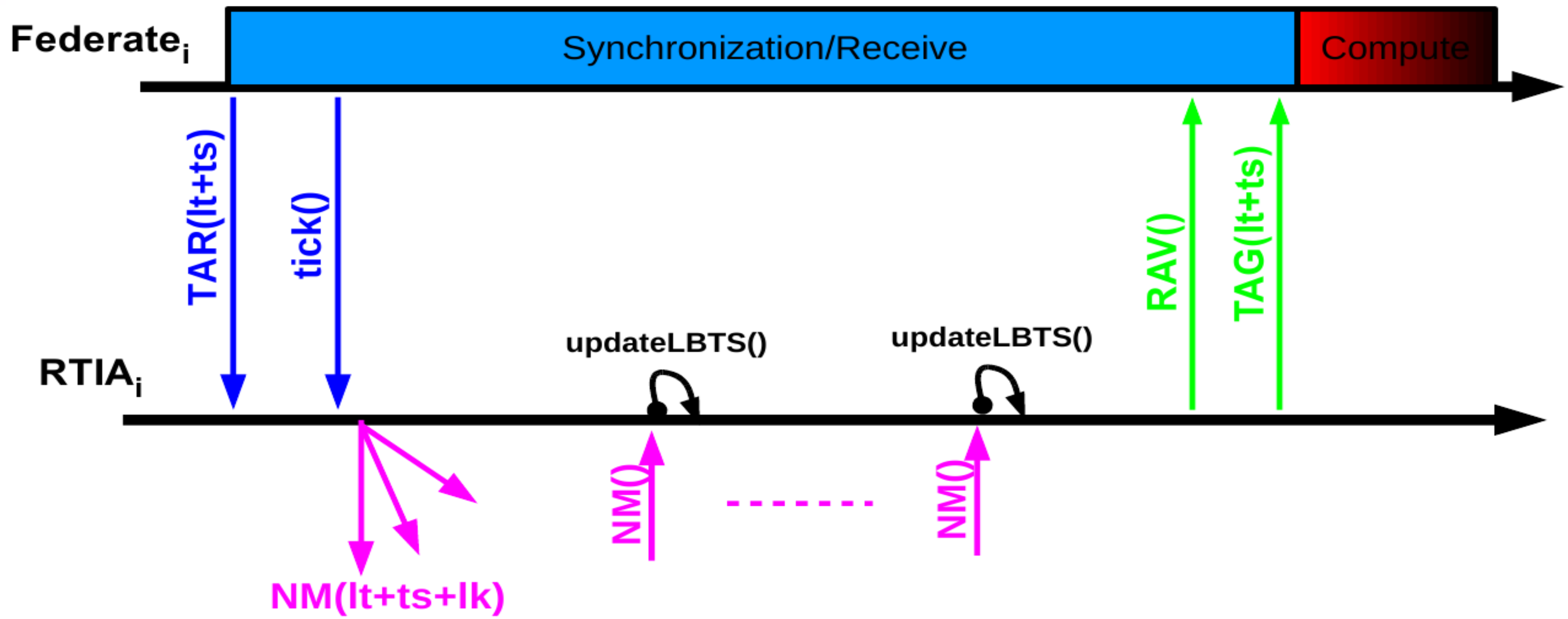
- Concept introduced by Fujimoto and McLean;
- Federates repeat the ***same pattern*** of execution periodically (time step noted *Δt*).



- Each step is the execution of 4 phases:
  - (1) a *reception* phase;
  - (2) a *computation* phase;
  - (3) a *transmission* phase;
  - (4) a *slack time* phase.

- Onera's studies show the necessity of adding a *synchronization* phase that could be done by 3 techniques:
  - (1) Consulting an hardware clock;
  - (2) Sending an interaction which rhythms the simulation;
  - (3) Using time management algorithms.





- Quantify NULL Message exchange
  - Allow a better evaluation of a *WCET* for a Real-time federate;
  - Add some deterministic mechanism;
  - Metrics available on an given simulated time interval;
  - Metrics available for a federate between its *TAR() service call* and *TAG() RTI callback*.

Basic Assumptions

• The global simulation (Federation) is composed by *N* periodic federates

• For a federate i noted *fed(i)*:

- *t(i)* its ***logical time***;
- *lk(i)* its ***lookahead***;
- *ts(i)* its ***time step*** (expression of its *computational periodicity* in simulated time);
- *gt(j)* is the global state vector of federate *j*; This vector is currently updated during simulation by NULL MESSAGE exchange;

• TS_{LCM} is the study interval usually equal to the least common multiple of all federate step.

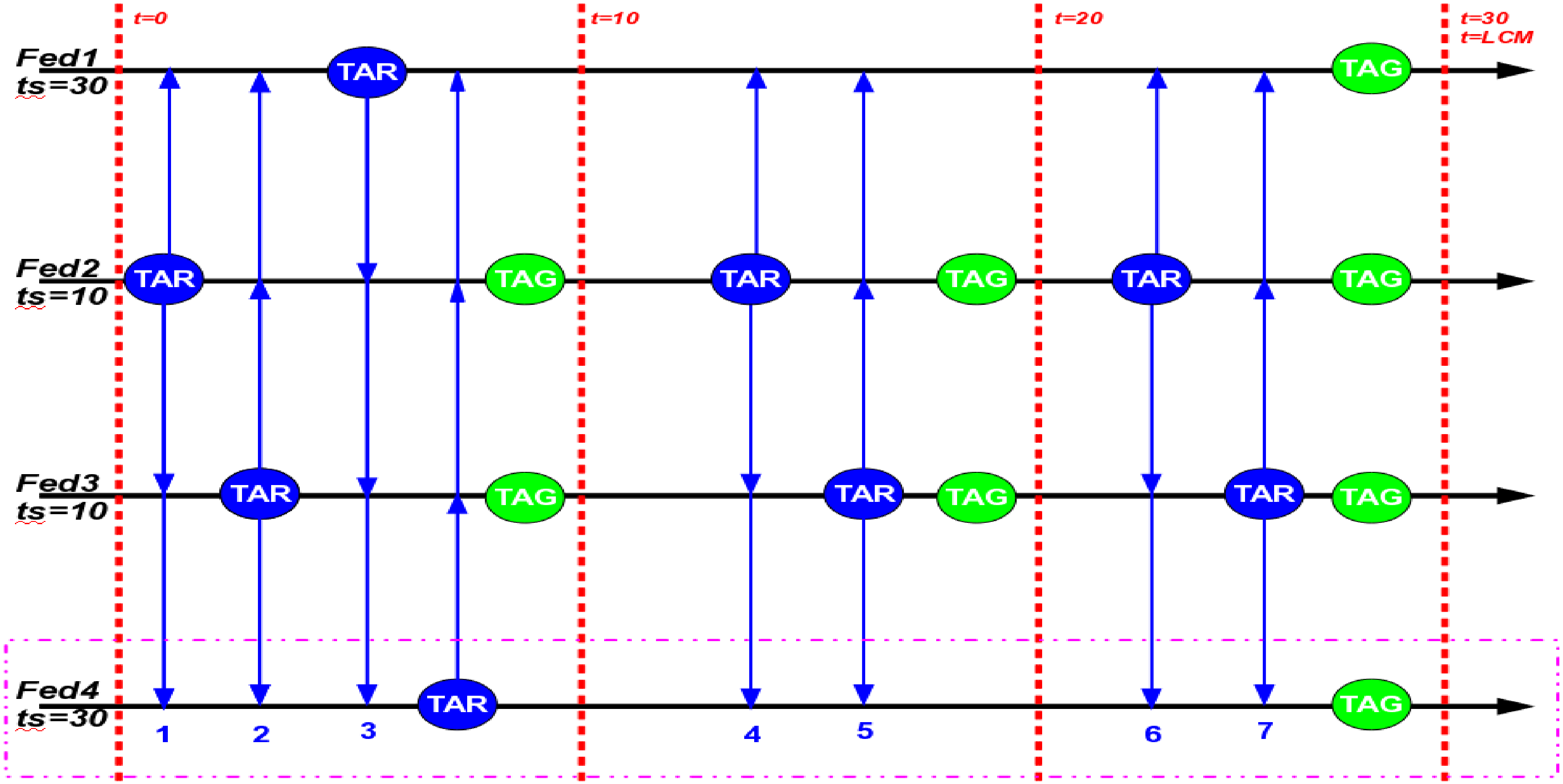
$$NM_S(i) = \frac{TS_{LCM}}{ts(i)}$$

$$NM_R(i) = \sum_j \left(\frac{TS_{LCM}}{ts(j)} \right)$$

$$W_j = \left\lceil \frac{t(i) + ts(i) - gt(j)}{ts(j)} \right\rceil$$

$$\sum_j W_j \le NM_{Cycle}(i) \le \sum_j W_j + (N-1)$$

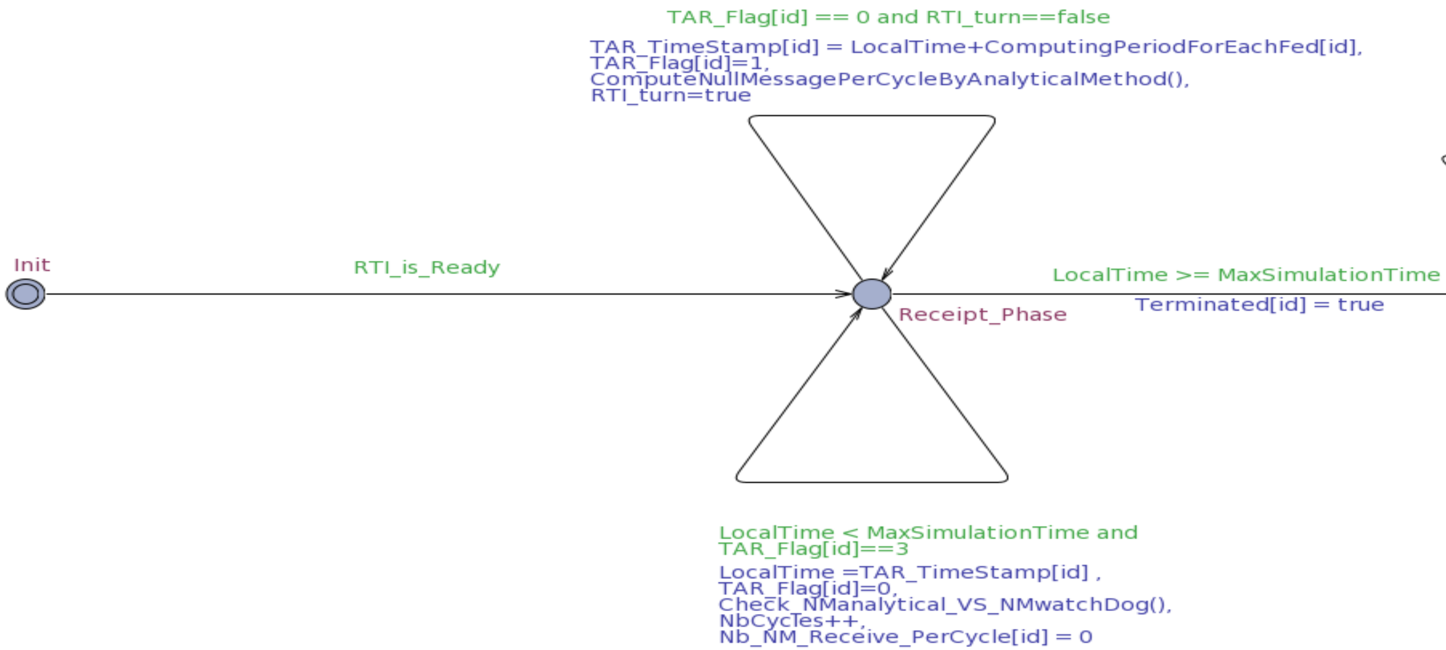
for ***i,j €[1, … , N]*** and ***i≠j***



$$NM_R(i) = \sum_j \left(\frac{TS_{LCM}}{ts(j)} \right) \rightarrow NM_R(4) = \frac{30}{30} + \frac{30}{10} + \frac{30}{10} = 7$$

FUTURE TRENDS (1/2)

- Systems simulated with HLA may have a discrete modeling:
  - characterized by a *given state*;
  - its behavior over time can be described by a *sequence of state transition*.
- We were interested in formalism of Finite and Temporized Automata with the ***UPPAAL*** tool to validate our approach for each part of the problem.



FUTURE TRENDS (2/2)

- First Results for Time Stepped Federate:
  - UPPAAL models for Federate and RTI are available;
  - Properties and Metrics *have been validated* by UPPAAL Verifier for 2, 3 and 4 federates;
  - Combinatorial explosion for more ...

- First Results for Event Driven Federate:
  - UPPAAL models for federate and RTI are under construction;
  - Verification for soon...

- Perspectives:
  - Investigate the Similarities and differences between NULL MESSAGE PRIME Algorithm and MATTERN one;
  - Check others formal techniques for validation.

➔ **Include these results to our general and global works on *real-time distributed simulations* (10E-SIW-011).**