Distributing Cyber-Physical Systems Simulation: The Satellite Constellation Case

Henrick Deschamps, Bastien Tauran, Janette Cardoso, Pierre Siron

5th Federated and Fractionated Satellite Systems Workshop
November 2–3, 2017, ISAE SUPAERO – Toulouse, France

Distributing Cyber-Physical Systems Simulation: The Satellite Constellation Case

Henrick Deschamps[*, ‡], Bastien Tauran[†, ‡], Janette Cardoso[‡], and Pierre Siron[‡]

[*]Airbus Operation SAS, Modelling and Simulation dept, Toulouse, France.
firstname.name@airbus.com
[†]Telecommunications for Space and Aeronautics (TéSA), Toulouse, France.
[‡]ISAE-SUPAERO, University of Toulouse, Complex Systems Engineering Dept, France.
firstname.name@isae-supaero.fr

16th October 2017

Abstract

The goal of this position paper is to contribute for the improving of Cyber-Physical System (CPS) simulations by introducing distribution. CPS use computations and communication tightly interacting with physical processes. So a CPS simulation needs to tackle with three kinds of simulations: the computational simulation, the physical simulation, and the communication simulation. In this paper, we will focus on the communication simulation, and its interaction with the two others simulations. We will draw a landscape of the existing concepts and technologies for distributing communication simulation, and then propose an architecture for interacting with the whole CPS simulation. We will apply this architecture to a simulation of a satellite constellation, where satellites can be simulated with different levels of precision, from the simple generic mathematical model to the heavy-featured CPS simulation.

Keywords
Satellite constellation; Network communication; Modelling; Simulation; Scheduling; HLA; CERTI; ns-2

1 Introduction

Cyber-physical systems (CPS) are systems integrating computational components, interacting with a physical plant and their environments. CPS also include communication networks between their components and sometimes with external systems. Vehicles, and more precisely in our context spacecraft such as satellites are good examples of CPSs, since they are driven by control loops, depending on their environments.

Such systems are costly to develop and might know incremental improvements during their life cycles. Thus, simulation is more and more used during the design process. Nevertheless, to receive the full benefit of a test executed in a simulated environment, one must prove that the simulated system is sufficiently valid, regarding some given needs.

Due to the hybrid nature of the CPS, different simulators might work on different parts of the simulation, and to allow the execution of the continuous components, these one must be discretized, fig. 1. Furthermore, multiple designers working on a single CPS, or multiple CPS interacting in the same environment, might use different simulation tools, depending on their domain. Finally, a given CPS simulation by itself might be sufficiently sophisticated to necessitate the use of interconnected simulators. Nowadays, simulation interconnection is a challenge.

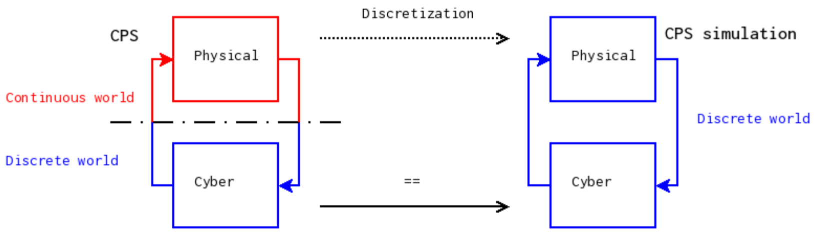

Figure 1: The heterogeneous nature of CPS, and CPS simulation

In this paper, we focus on the simulation of the network communications in and between a CPS simulation. In Section 2, we will present current approaches used for simulation of networked CPS. Then, In Section 3, we will propose an approach for implementing a proof of concept before the final case study. In Section 4, we will illustrate this distributed CPS simulation considering a satellite constellation simulation. Finally, conclusions are presented in Section 5.

2 Network communication simulation and CPS simulation

In this section, we introduce existing approaches for network simulation in CPS. We were able to identify simulators initially specialized in simulating systems or CPS, considering lately in their concepts the network communication simulation, and network simulator willing to introduce external mobile CPS.

2.1 Modelling network communications in CPS simulation

The integration of network simulation in CPS simulation has already be done in the past.

Using Modelica, in [1] the authors simulate FlexRay in CPS simulation. FlexRay is an automotive network communications protocol, designed as an alternative to Controller Area Network (CAN) for vehicle bus. Nevertheless, in this work, the authors use an approach that relies on the FlexRay concepts, and this is not applicable for other protocols.

In [2] and [3], the authors introduce the use of the High-Level Architecture (HLA) [4] to simulate a satellite formation flight, on a platform named PRISE, a French acronym for Platform for Research in Embedded Systems Engineering. As well as in [1], the model of communication is sufficient for the goal of the simulation, but the representativeness of the network communication is limited for more complicated simulations. Furthermore, a different implementation of HLA, ARTIS (Advanced Run-Time Infrastructure System), originally designed as a benchmark Run-Time Infrastructure (RTI), seems able to simulate network communication [5][6].

Network simulation is too complex for an efficient implementation in existing CPS simulations. It is not technically impossible to continue the integration of network simulations, especially in PRISE's simulations since its simulators can be event base, which is efficient for implementing network simulation, but the development costs might be high, and we should consider other approaches.

2.2 Modelling CPS in network simulations

The need of modeling CPS also exists in the use cases of a network simulator.

This is particularly true for vehicular networks, using network simulator 2 (ns-2) simulation [7], and OmNet++ [8]. Nevertheless, they do not consider a real simulation of CPS, but analytical models of their movements.

Similar approaches exist in satellite communications, for instance, ns-2 and satellite network [9] or OmNet++ [10][11][12].

These simplified CPS analytical models might be sufficient for the goal of the network simulation, but cannot be used in a context such as the design phases of a networked CPS component.

2.3 Interconnection frameworks

The last approach consists in interconnecting networked CPS simulations. Although the interconnection of simulation is widely covered in the literature [13], the implementation of these interconnections is not frequent when considering networked CPS simulation.

Due to ns-2 CPU usage limitation, the idea of using a framework for better performances has already been addressed with Parallel/Distributed Network Simulator (PDNS), using the HLA standard [14], or concepts for global improvements, using HLA too [15]. Other simulators, such as OmNet++ or ns-3 distribute computation [16][17], use a widespread framework, such as Message Passing Interface (MPI). Nevertheless, these interconnections are used to overcome the execution support limitation.

An exception is in [18], where HLA-OMNET++ is proposed as an integration of HLA in OmNet++ for heterogeneous environments. In HLA-OMNET++, an HLA compliant component is implemented in OmNet++, interfaced with a Java block to communicate with the HLA RTI. The HLA compliant component is a specialization of the OmNet++ abstract `cScheduler` class, which is the class encapsulating event scheduling in the network simulation.

3 Proposed approach

In this paper, ns-2 is the best candidate for simulating. We choose to reuse the PDNS approach with ns-2 and implement a scheduler in network simulation to obtain heterogeneous simulation, similarly to HLA-OMNET++[18]. The choice of ns-2 is driven by the application domain, ns-2 being the only network simulator providing a precise simulation of satellite networking[9]. It should be noted that ns-3 could have been a great candidate since its integration of satellites network simulations is maturing. Nevertheless, the integration of satellites network in ns-2 is currently the stablest one.

The IEEE High-Level Architecture HLA standard [4] targets distributed simulation. For the framework, we selected the CERTI implementation of HLA. CERTI is used in [3] and in our recent works on simulation scheduling formalism [19]. With HLA/CERTI, a CPS is seen as a federation (a distributed simulation) grouping several federates (simulators) which communicate via publish/subscribe patterns.

Recent works proved that the interconnection of simulation using HLA/CERTI is possible. In [20] and [21], the authors coupled HLA/CERTI with Ptolemy II, a modeling, and simulation framework for CPS. These works highlihted the complexity of coupling HLA/CERTI with a simulator, even if they are conceptually compatible. The main issue is the time management.

In [20], the authors propose a time synchronization semantic allowing the time synchronization between the environments. This semantic describe how the different simulators can advance in time simultaneously, In [21] the authors went further in the description of the interface, and they associate rules to the semantic that have to be verified during implementation. More precisely, the representation of the time in HLA/CERTI and Ptolemy II is different. HLA/CERTI uses IEEE-754 double precision floating arithmetic, while Ptolemy II uses a time resolution in Java double, multiplied by an integer time value. The difference of time representation and manipulation induce fragile interconnections where simulators can send an event in the past of other simulators.

Another issue is the communication interface between Ptolemy II and HLA. In [20], the authors introduce actors in Ptolemy, using the HLA services for publication and subscription.

3.1 Scheduler implementation

We chose a network simulator offering an easy way to implement schedulers, as well as good performance for satellite communications simulations: ns-2. Ns-2 being a discrete event simulator, this integration with HLA/CERTI is conceptually possible. Moreover, the works with PDNS are promising [14]. Moreover, ns-2 is under a GNU GPL license with specific exceptions, and the documentation of its scheduler implementation is clear [22][23].

To interconnect the scheduler with other federates, this one must be HLA compliant. More precisely, there are two points to take into account:

- The synchronized time advancement of ns-2 and the other simulator.
- The message interface for a sink/source of packets in a federate to ns-2.

Considering the time synchronization, [20] and [21] covers this point sufficiently.
Considering the message interface between the federates, we have to make a technical choice between:

- Describing all the fields of the messages.
- Directly sending the message.

The first case helps the manipulation of a message for sinks and sources as they can retrieve only the data they are interested in. The second case complexifies the sources and sinks of messages, but simplify the conversion of packets in ns-2. An example of a federation object model for the second case, used as a configuration file for the simulation, is described in listing 1.

Our objective being an increasing performance by reducing the ns-2 treatments, we choose the second case.

Listing 1: The common federation object model (the FED file) for packet interface

```
(FED
    ...
        (objects
                (class ObjectRoot
                        (attribute privilegeToDeleteObject reliable timestamp)
                        (class RTIprivate)
                        (class Message ;;
                                (attribute data reliable timestamp)
                        )
                )
    )
    ...
)
```

3.2 Development steps

The following steps provided are a high-level view of the simulation architecture. A key issue of our approach is the analysis of the job-specific aspect of the network communication, and then, the interfacing between the simulators.

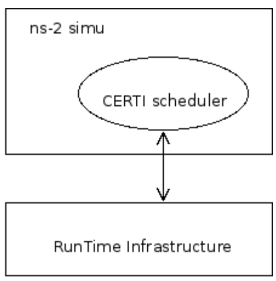

Figure 2: Implementation of a CERTI scheduler in ns-2.

- Step 1: Integrating CERTI scheduler in ns-2.
  The first step consists in implementing a CERTI scheduler able to send and receive packets and parameters through the RTI. This step will allow us identifying precisely the integration means in ns-2. Illustration in fig. 2.

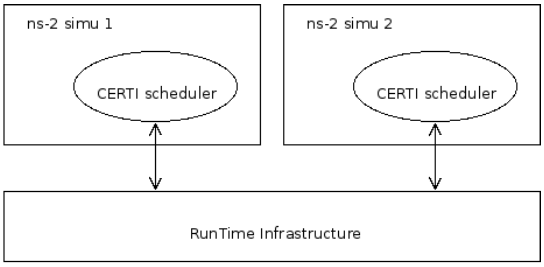

Figure 3: Connection of two ns-2 simulators with CERTI scheduler.

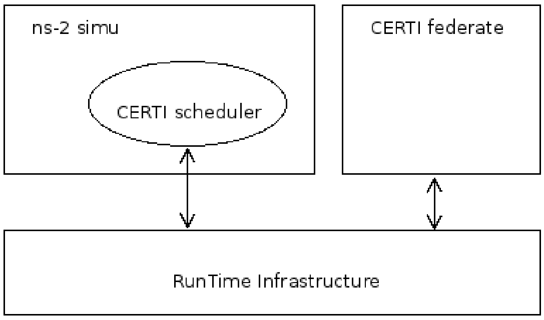

Figure 4: Connection of an ns-2 simulator with a global purpose federate.

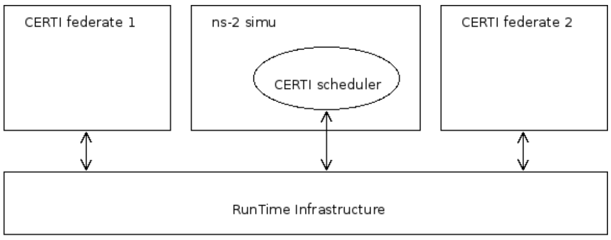

Figure 5: Communication of two federates through ns-2 simulation.

- Step 2 (optional): Distributing two ns-2 with CERTI schedulers.
  Once the scheduler is implemented, we should verify that an ns-2 simulation can be hosted on two ns-2 simulators, exchanging packets through the RTI. A geographic distribution of the network should be considered, where two communicating nodes are distributed on two ns-2 simulators, but a distribution depending on the communication stack can be considered too. Illustration in fig. 3.
- Step 3: Interconnecting ns-2 with CERTI scheduler, with a global purpose federate.
  During this step, we will more precisely identify the level of interconnection for the sending of packets and data from and to a federate that does not have a view on the network. A federate that know the data it wants to send, at the network level (for instance IP packet), and eventually some data for ns-2 to calculate delay on the link, such as the physical distance between two simulations in federates. Illustration in fig. 4.
- Step 4: Connecting two global purpose federates through a ns-2 with CERTI scheduler network simulation. This step validates the proof of concept, and the case study should be implementable. Illustration in fig. 5.

4 Satellite constellation simulation distribution

In this section, we introduce the case study, a constellation of satellites, were multiple satellites are forming a communication network, with given routes. The goal of this case study is to distribute the simulation on multiple CPU, keeping ns-2 as the core simulator for the network.

As the different satellites are independent, it is a natural boundary for the simulation distribution. A satellite can be a component of the simulation, assignable to a CPU.

The case study is composed of multiple instances of a satellite simulation, communicating through a network simulation, using ns-2, illustrated in fig. 6.

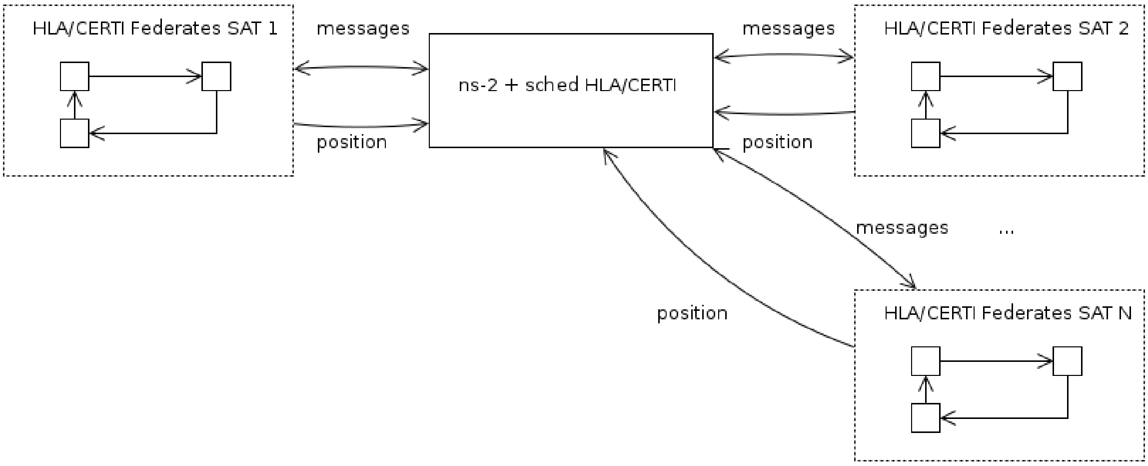

Figure 6: Illustration of the satellite constellation simulation distribution.

Each satellite uses ns-2 to exchange packets, providing to ns-2 their positions, while the routes are preconfigured in ns-2.

A satellite simulation instance will focus on calculating its position, and sending/receiving packets, while ns-2 will route packets between the satellites, simulating delays based on the provided positions. For the simulation to work, we have to modify the FED, and add a satellite position object and attribute, and to modify ns-2 to retrieve these positions. In the federation, the satellite model can be replicated as much as we need, allowing the validation of satellite constellation with a high number of satellites. Furthermore, federates could be implemented as sophistically as we need, allowing the validation of very complex systems and protocols.

5 Conclusion

In this paper, we proposed an approach and a case study to distribute CPSs simulation were the network is an essential part of the system.

Based on the scientific literature, we were able to identify the most difficult parts of the implementation of this approach:

- To define and to implement the time synchronization mechanism.
- To propose an interface for data exchange.

In the following, we will implement the ns-2 scheduler as presented in Section 3, and we will be able to implement the case study. This approach allows validating the satellite constellation by replacing the analytic component in ns-2 simulator by models in federates as complex as the simulation needs.

Acknowledgement

The work described in this paper is supported through an Industrial Agreement for Research Training — CIFRE — financed by the French National Association for Research in Technology (ANRT).

This work is also financed and supervised by Airbus Operation SAS.

The authors would like to thank Emmanuel Lochin from the ISAE-SUPAERO, University of Toulouse, for its helpful adjustments proposed.

References

[1] D. Henriksson and H. Elmqvist, “Cyber-Physical Systems Modeling and Simulation with Modelica,” 2011.

[2] J.-B. Chaudron, M. Adelantado, E. Noulard, and P. Siron, “HLA high performance and real-time simulation studies with CERTI,” 2011.

[3] J.-B. Chaudron, “Architecture de simulation distribuée temps-réel,” Ph.D. dissertation, Toulouse, ISAE-SUPAERO University of Toulouse, Jan. 2012.

[4] R. M. Fujimoto, “Parallel simulation: parallel and distributed simulation systems,” in *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society, 2001, pp. 147–157.

[5] L. Bononi, M. Bracuto, G. D’Angelo, and L. Donatiello, “ARTIS: a parallel and distributed simulation middleware for performance evaluation,” in *ISCIS*. Springer, 2004, pp. 627–637.

[6] L. Bononi, G. D’Angelo, and L. Donatiello, “HLA-based adaptive distributed simulation of wireless mobile systems,” in *Proceedings of the seventeenth workshop on Parallel and distributed simulation*. IEEE Computer Society, 2003, p. 40.

[7] “VanetMobiSim/Ns-2 Simulator | VANET/ITS Website (NEO).” [Online]. Available: http://neo.lcc.uma.es/staff/jamal/vanet/?q=node/9.

[8] M. Báguena, S. M. Tornell, Torres, C. T. Calafate, J. C. Cano, and P. Manzoni, “VACaMobil: VANET Car Mobility Manager for OMNeT++,” in *2013 IEEE International Conference on Communications Workshops (ICC)*, Jun. 2013, pp. 1057–1061.

[9] “ns-2 documentation – 17. Satellite Networking in ns.” [Online]. Available: https://www.isi.edu/nsnam/ns/doc/node196.html

[10] B. Niehoefer, S. Šubik, and C. Wietfeld, “The cni open source satellite simulator based on omnet++,” in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 314–321.

[11] “os3: Satellite tracking library for OMNeT++/INET-Framework,” Jul. 2017, original-date: 2013-08-14T10:05:09Z. [Online]. Available: https://github.com/inet-framework/os3

[12] “OS: The Open Source Satellite Simulator - Home.” [Online]. Available: http://www-os3.kn.e-technik.tu-dortmund.de/index.html

[13] A. Tolk and J. A. Muguira, “The levels of conceptual interoperability model,” in *Proceedings of the 2003 fall simulation interoperability workshop*, vol. 7. Citeseer, 2003, pp. 1–11.

[14] “PDNS - Parallel/Distributed NS.” [Online]. Available: https://www.cc.gatech.edu/computing/compass/pdns/

[15] G. F. Riley, R. M. Fujimoto, and M. H. Ammar, “A generic framework for parallelization of network simulations,” in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 1999. Proceedings. 7th International Symposium on*. IEEE, 1999, pp. 128–135.

[16] M. Stoffers, R. Bettermann, J. Gross, and K. Wehrle, “Enabling distributed simulation of OMNeT++ INET models,” *arXiv preprint arXiv:1409.0994*, 2014.

[17] J. Pelkey and G. Riley, “Distributed simulation with MPI in ns-3,” in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 410–414.

[18] E. Galli, G. Cavarretta, and S. Tucci, “HLA-OMNET++: An HLA Compliant Network Simulator.” IEEE, Oct. 2008, pp. 319–321.

[19] H. Deschamps, P. Siron, J. Cardoso, and G. Cappello, “Toward a formalism to study the scheduling of Cyber-Physical systems simulations,” in *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT) (DS-RT'17)*, Rome, Italy, Oct. 2017.

[20] G. Lasnier, J. Cardoso, P. Siron, C. Pagetti, and P. Derler, “Distributed simulation of heterogeneous and real-time systems,” in *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*. IEEE Computer Society, 2013, pp. 55–62. [Online]. Available: http://dl.acm.org/citation.cfm?id=2570877

[21] C. Michel, J. Cardoso, and P. Siron, “Time management of heterogeneous distributed simulation,” in *31st European Simulation and Modelling Conference - ESM'2017, October 25-27, 2017*, Lisbon, Portugal.

[22] “ns-2 documentation – event Scheduler.” [Online]. Available: http://nile.wpi.edu/NS/scheduler.html

[23] “ns-2 documentation – 4.2 Schedulers and Events.” [Online]. Available: https://www.isi.edu/nsnam/ns/doc/node32.html

Appendix A — Glossary

Table 1: Glossary

ARTIS	Advanced RTI System
CAN	Controller Area Network
CPS	Cyber-Physical System
MPI	Message Passing Interface
ns	Network Simulator
PDNS	Parallel/Distributed Network Simulator
PRISE	*Plate-forme pour la Recherche en Ingénierie des Systèmes Embarqués* Platform for Research in Embedded Systems Engineering
RTI	Run-Time Infrastructure