Verification of the Schedule Consistent Update Mechanisms of FTTRS with UPPAAL

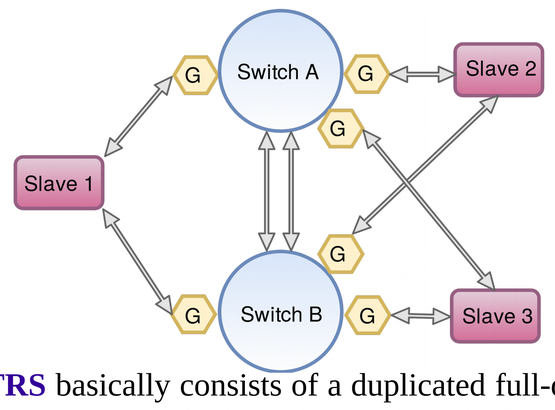
Abstract

Critical Adaptive Distributed Embedded Systems (ADESs) are nowadays the focus of many researchers. ADESs are envisioned to dynamically modify their behavior to support changes of their **real-time** and **dependability requirements** at runtime as the conditions of the environment in which they operate vary. To provide ADESs with an adequate communication infrastructure, our research group proposed the Flexible Time-Triggered-Replicated Star (FTTRS). **FTTRS** provides highly **reliable communication** services on top of Ethernet, while keeping the **adaptivity** benefits that the Flexible Time-Triggered (FTT) communication paradigm offers from a real-time perspective. This work formally verifies, by means of model checking, the correctness of the mechanisms FTTRS includes to enforce consistent changes of the communication scheduling at runtime.

Daniel Bujosa, Sergi Arguimbau, Patricia Arguimbau, Julián Proenza and Manuel Barranco
[daniel.bujosa, sergi.arguimbau, patricia.arguimbau, julian.proenza, manuel.barranco]@uib.es
Dept. de Matemàtiques i Informàtica, Universitat de les Illes Balears, Palma de Mallorca, Spain

This work is supported in part by the Spanish Agencia Estatal de Investigación (AEI) and in part by FEDER funding through grant TEC2015-70313-R (AEI/FEDER, UE). And also by SOIB, under the JP-SP 49/17 project (ESF, Youth Guarantee)

1. Introduction



FTTRS basically consists of a duplicated full-duplex Ethernet star in which each switch embeds an FTT master.

The schedule is stored in the database of each master (**SRDB**) and of each slave (**NRDB**). Masters isochronously transmit a Trigger Message (**TM**) to divide the communication in rounds (**Ecs**).

The TM is indeed replicated (proactively retransmitted several times) to provide high **reliability**.

Each EC includes a window for the TM replicas (**TMW**), a window for periodic traffic (**SW**), and a window for aperiodic one (**AW**).

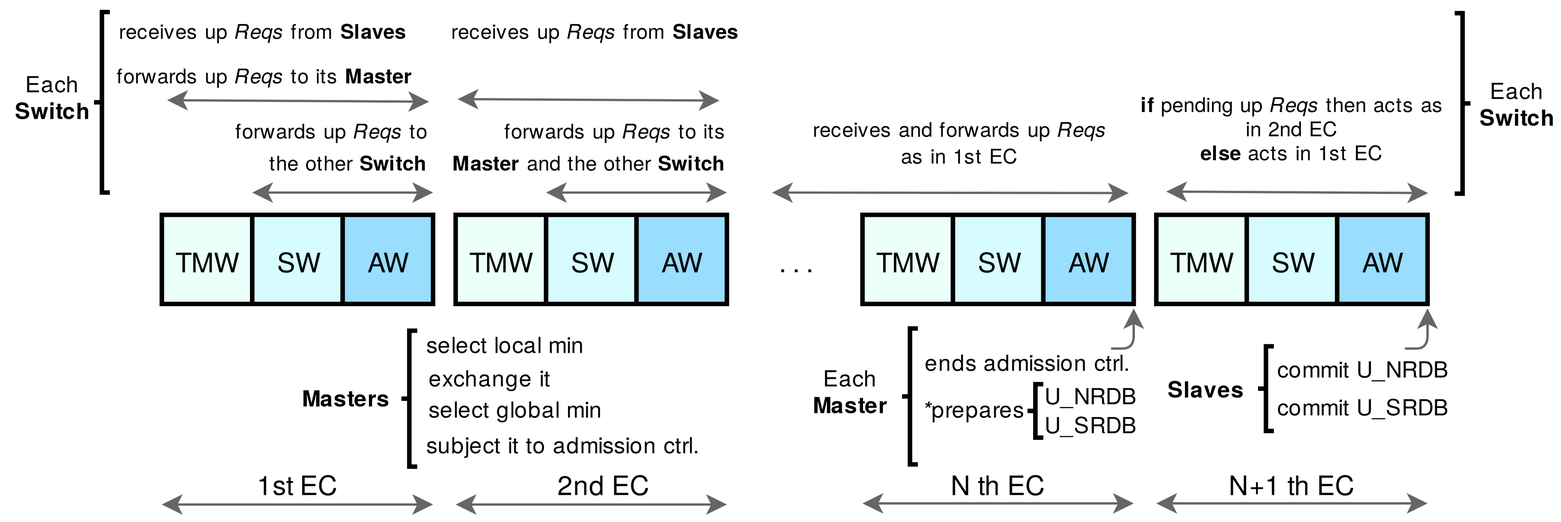
The TM specifies which periodic messages slaves have to transmit, according to the current schedule.

Slaves can ask for changes in the schedule, by sending an **Update Request**. **Masters** execute a **Schedule Consistent Update Mechanism** to consistently subject the Update Requests to admission control and to update all databases with the appropriate changes.

2. Objective

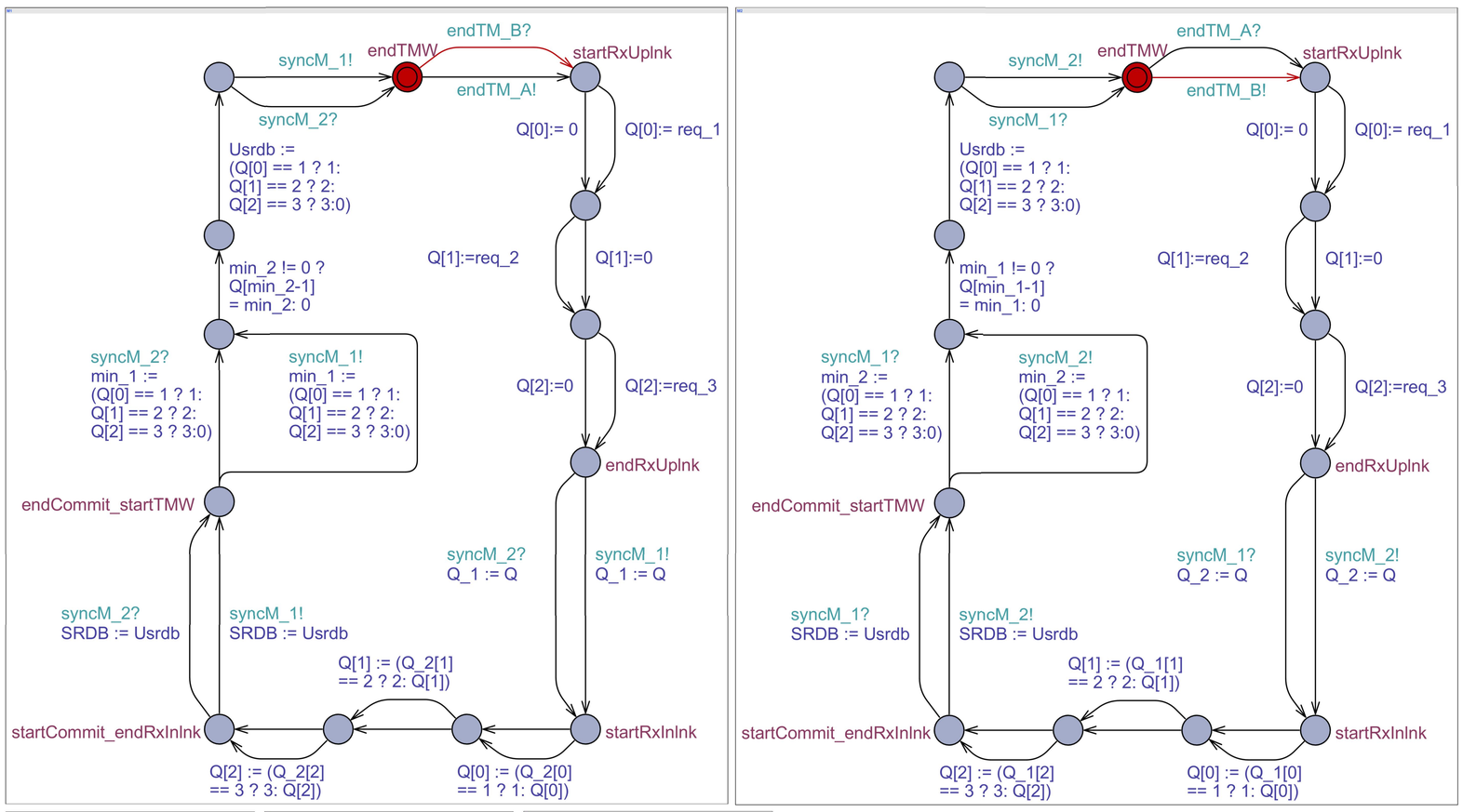
To model and formally verify the **Schedule Consistent Update Mechanism** of FTTRS by means of a model checker called **UPPAAL**, which is specially suited for real-time systems

3. Timeline of the Schedule Consistent Update Mechanism of FTTRS



Legend:
U_NRDB (Updates to be committed to NRDBs)
U_SRDB (Update to be committed to SRDBs)
U_NRDB is piggyback in the replicated TM of the next EC

4.Model of the Schedule Consistent Update Mechanism



Three nodes (N1, N2, N3) is the minimum that originates all kind of scenarios in the queues of the replicated masters (M1, M2). Although is possible to create inconsistencies in the queues with simply one or two nodes, the presence of three nodes allows to have different and common update requests simultaneously in both masters. On the other hand, four or more nodes would create the same kind of scenarios that we obtain with three nodes.

5. Model Verification

First, we verified the following safety property to check that the mechanisms do never lead to a deadlock: **A[] not deadlock**.

Second, we checked that both SRDBs are always consistent. For this, we verified that the following safety property holds: **A[] MA.SRDB == MB.SRDB**, where MA and MB respectively represent the switch/master A and B.

Finally, to further check that the just mentioned property is not only fulfilled in trivial cases, i.e. not only when the SRDBs are not updated but also when they are, we used the following reachability property: **E <> MA.SRDB != 0**.

6. Conclusion

The **Flexible-Time-Triggered-Replicated Star** represents a step towards developing networks that appropriately support future critical **Adaptive Distributed Embedded Systems**.

Thanks to FTTRS, the FTT communication paradigm based on top of Ethernet. Now it is not only possible to take advantage of the **real-time** and **operational flexibility** of FTT, but also of the **high reliability** FTTRS provides.

In this work we have formally verified the correctness of the most complex **FTTRS's consistency mechanism**, i.e. the one that guarantees that the traffic schedule is consistently updated at runtime.