HSN: a co-simulation Tool for Transaction Level Modelling of Networked Embedded Systems

Networked embedded systems represent a considerable portion of the number of embedded systems as they range from network intermediate systems (e.g., router and access points) to mobile phones and wireless sensor networks; for this reason their fast and efficient design is ever more strategic for manufacturers. The Hardware-Software-Network (HSN) co-simulation framework has been shown to be an efficient approach to improve design quality and shorten time-to-market. We now present a further improvement of this tool with the new following features:

- Linux-based environment;
- support for Transaction Level Modelling (TLM) of embedded systems;
- translation of simulated network packets into actual packets for integration with actual systems;
- support for the design of multimedia networked embedded systems with real-time communication constraints.

As a case study, the presented tool is applied to the design of a Voice-over-IP client. Furthermore, the presented tool and the case study have been successfully used in various Computer Engineering classes showing their validity not only for design but also for education.

Co-design and co-simulation workflow

The tool allows modelling both the *system* to be implemented and the *network* in which it will operate. Fig. 1 shows an abstract view of the co-design and co-simulation workflow. The joint support of Transaction Level Modelling for the system and of Network Modelling leads to a two-dimension design space. The vertical dimension addresses the refinement of the system model; during this process the network model is used both to drive architecture exploration and to test that communication requirements are met. The horizontal dimension addresses the task of modelling different kinds of networks which are used to validate the system after each refinement step. During the workflow, the designer provides the models for the required components and starts simulations. Simulation results are shown for each component being simulated. At the beginning, it is not yet determined which parts of the system will be implemented in HW and which in SW and, therefore, we talk about system/network co-simulation. After HW/SW partitioning the full support for HW/SW/Network (HSN) co-simulation is exploited.

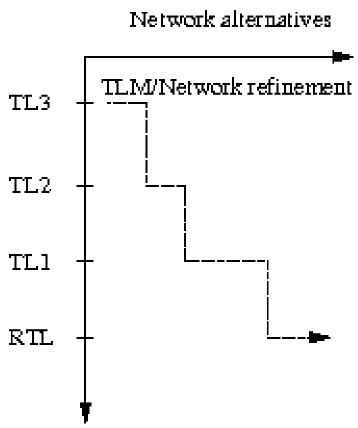

Figure 1. Co-design and co-simulation workflow

Component modelling

The models provided for the simulation can be developed using the HSN co-simulation tool or imported from outside. The network topology can be either graphically defined or described using the syntax of a well-known network simulator --NS-2-- whose models can thus be imported. The un-partitioned system model and the HW components are described in SystemC, a well-know extension of C++ supporting TLM. Software components, i.e., the operating system and application SW, are executed by the Instruction Set Simulator of the target platform. Source code, if available, can be cross-compiled for the target platform.

Case study: design of a Voice-over-IP client

The tool has been applied to a case study consisting in the design of a Voice-over-IP (VoIP) client. Fig. 2.a shows the block diagram of the whole System/Network scenario. The VoIP client captures voice samples, encodes them using an ADPCM compression scheme, and sends the encoded bitstream as a sequence of Internet Protocol packets to the voice player for decoding and reproduction. The size of the voice packets is a design parameter.

In the lower part of Fig. 2.a the NS-2 model of the network is depicted. Node 0 represents the VoIP client. Voice packets flow through a LAN link (10 Mb/s) link and a geographical link with constrained capacity. Since a minimum end-to-end bandwidth is required for the correct behaviour of the VoIP client, the capacity of this link is changed during the design process to validate the system. A concurrent constant bitrate (CBR) packet flow is also injected into the network to reproduce congestion on Node 2. The Tap Agent on Node 3 convert simulated voice packets into actual IP packets which are sent to a standard voice receiver running outside the HSN tool. Conformance to VoIP protocol standards can thus be checked against an actual implementation of the receiver.
Fig. 2.b shows the TLM/Network refinement in the 2-dimension space represented by System configurations and Network alternatives; different System/Network configurations are represented by boxes. A refinement of the System model leads to a shift in the vertical direction while a change of the Network model leads to a shift in the horizontal direction.
Through the HSN tool the designer can change the following aspects: abstraction level of System description (TL3-TL1), packet size, the capacity of the geographical link, and the presence of CBR concurrent traffic. Simulation results from the System model (computational performance) and from the Network model (packet loss rate and end-to-end delay) are used to drive architecture exploration and to validate the System at each abstraction level. Furthermore, the performance of the System can be perceptually evaluated though the standard voice receiver.
The architecture of the System at the lowest Transaction Level (TL1) is depicted in Fig. 3.

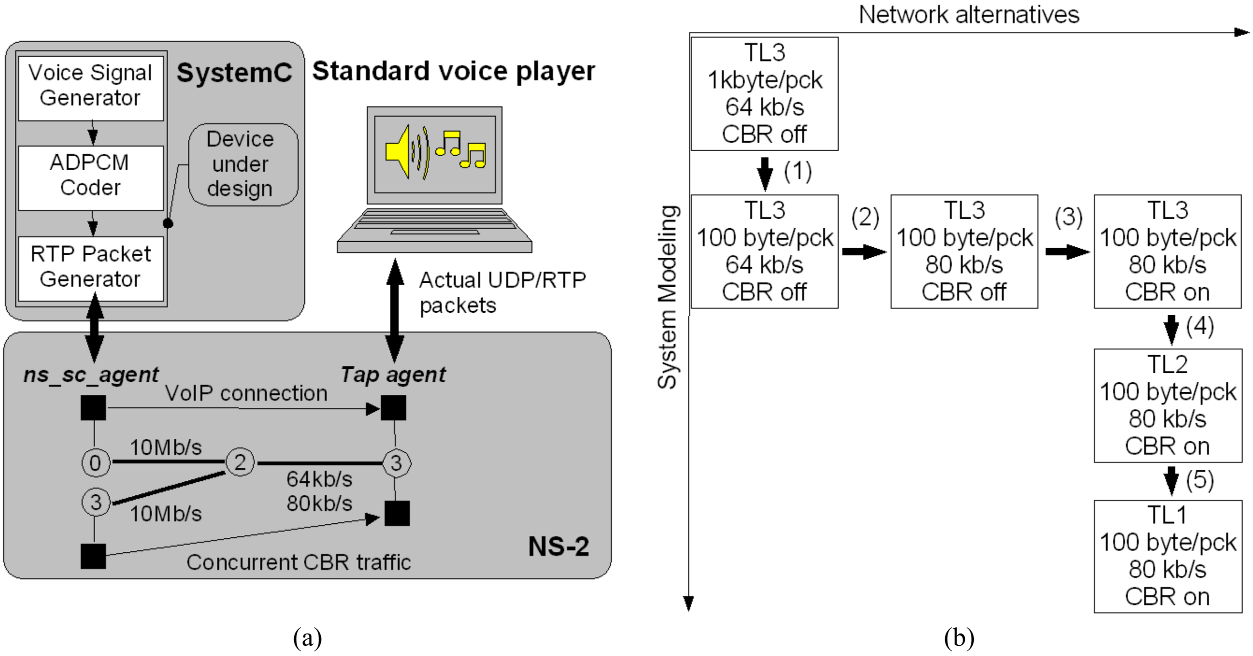

Figure 2. (a) Block diagram of the case study: the upper grey box represents the System model at the highest abstraction level (TL3) while the lower grey box represents the Network model. (b) Example of use of the HSN tool for the design workflow of the Case Study.

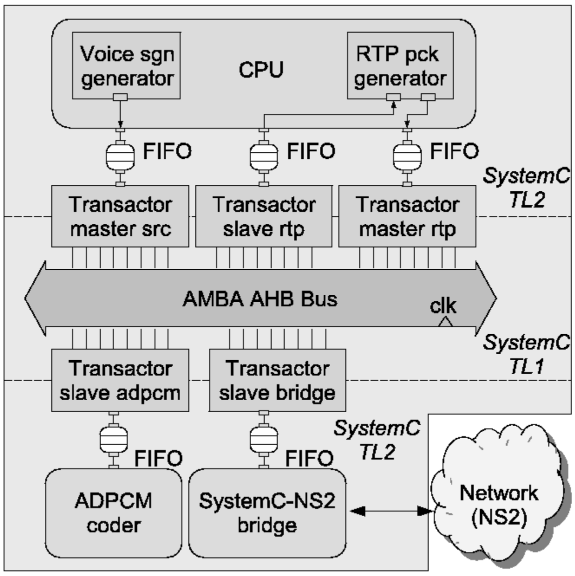

Figure 3. Architecture of the System at the lowest Transaction Level (TL1).