Heterogeneous Hybrid Networks Reliability Simulation

Pablo Gutiérrez Peón*†, Francisco Pozo†, Guillermo Rodriguez-Navas†
*TTTech Computertechnik AG, Vienna, Austria
†School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
Email: pablo.gutierrez-peon@tttech.com, francisco.pozo@mdh.se, guillermo.rodriguez-navas@mdh.se

This work aims at providing a novel tool HeNReS (Heterogeneous Hybrid Networks Reliability Simulation) to test the reliability of hybrid wired/wireless networks were both time-triggered real-time and non-real-time traffic travel seemingly over different transmission media. Nowadays, the inherent diversity of these networks makes the analysis more complex, specially with the addition of wireless links, susceptible to problems like fading, shadowing or interference, which are typically disregarded in wired links. A simulation of these networks usually requires the cooperation of multiple tools that do not share the same input/output interfaces. Our simulation tool facilitates this process providing integrated state-of-the-art tools where the user only needs to provide a set of simulation parameters in a single file to obtain the reliability of the network. The simulation tool is executed using a *makefile* that will perform a different simulation for every set of parameters given and returns performance measures in terms of reliability and message delays among others.

Internally, the simulation platform consists of four main software components: network and traffic generator, traffic scheduler, network simulator and results processing. In Figure 1 we can see how each software component interfaces with others using a file or set of files. In the next paragraphs we go more into detail into the function of each component.

The network and traffic generator is able to create any non-cyclic network topology with different traffic profiles. Note that it is possible to specify more than one network and traffic, which will create multiple simulation configurations. The network is composed of four types of devices: wired end-systems, wireless end-systems, switches and access points. End-systems are responsible to send and receive traffic, while switches and access points relay messages through the network, creating multi-hop paths between end-systems. Access points serve to interface between wired and wireless media. The traffic is generated in form of messages, containing information of its size, period and deadline, that can be sent from one end-system to any group of end-systems of the network.

The traffic scheduler implements a scheduling algorithm capable of generating schedules for large and complex networks. It is developed using a combination of segmentation approaches and a Satisfiability Modelo Theory (SMT) Solver. It takes as input the network and traffic generated beforehand and assigns the transmission times for all the real-time messages in the network while satisfying its constraints. In the case of non-real-time traffic, it is handled at runtime by the network simulator.

The network simulator is developed in the OMNeT++ discrete event simulation framework. The basic components of an OMNeT++ simulation are the modules, the pieces where the functionality is implemented. Modules are connected to each other using gates and exchange information using message passing. The modules are partially described using the NED language from OMNeT++, that specifies the parameters, gates and nested modules, while the functionality is programmed in C++. The OMNeT++ initialization file is used to give value to the module parameters. The values can be grouped with labels that define different configurations, each specifying a set of values for the module parameters. The four types of network devices are able to dispatch messages in a time-triggered manner following the corresponding link schedule while non-real-time traffic can dynamically take the remaining time. The wired and wireless devices are based on IEEE 802.3 and IEEE 802.11 respectively, with a modified medium access control (MAC) layer. Inside the MAC, queues are FIFO, and traffic of different classes is segregated into different queues. Several mechanisms are incorporated to increase the reliability of wireless links, including retransmissions and cognitive radio. To test the reliability of the wireless links, interference emulating Carrier Sense Multiple Acess (CSMA) and jamming at different levels of intensity, size of burst and frequency hopping is contemplated. The simulation can run with a graphical environment or using the command line. The network simulator provides results in three types of files: scalar and vectorial result files and log file. Scalar and vectorial result files retrieve statistics and vectors of raw data, including, e.g., number of packets sent and received, the instants when those packets are sent, the MAC to MAC delay, defined as the time between a message, is made available in the MAC of the source node until it reacher the MAC of the destination node. It serves to evaluate the overhead introduced by the MAC. For reliability, the result is provided in the form of percentage of successful message transmissions.. The log file includes entries about communication events (e.g., message sent/received) and the time at which they happened.

The results processing tool takes the simulation log file as an input and creates a report after evaluating whether the logic and timing of the events is according to the MAC protocol design and the schedule.

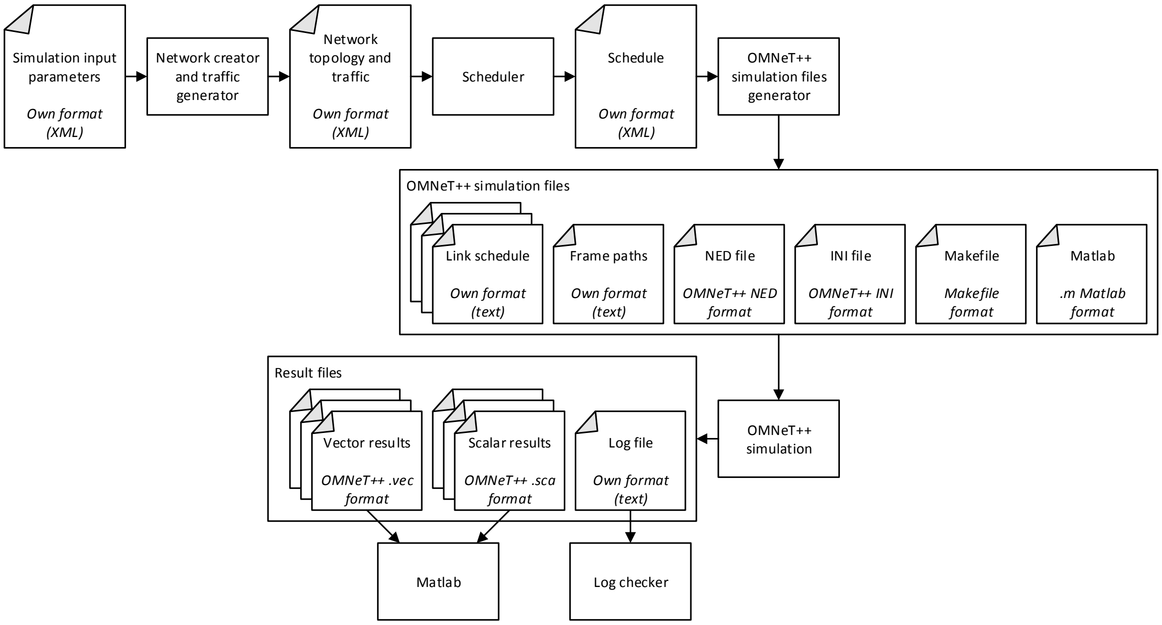

Fig. 1. Graphic representation of the tools flow implemented from the given simulation input up to getting the simulation results

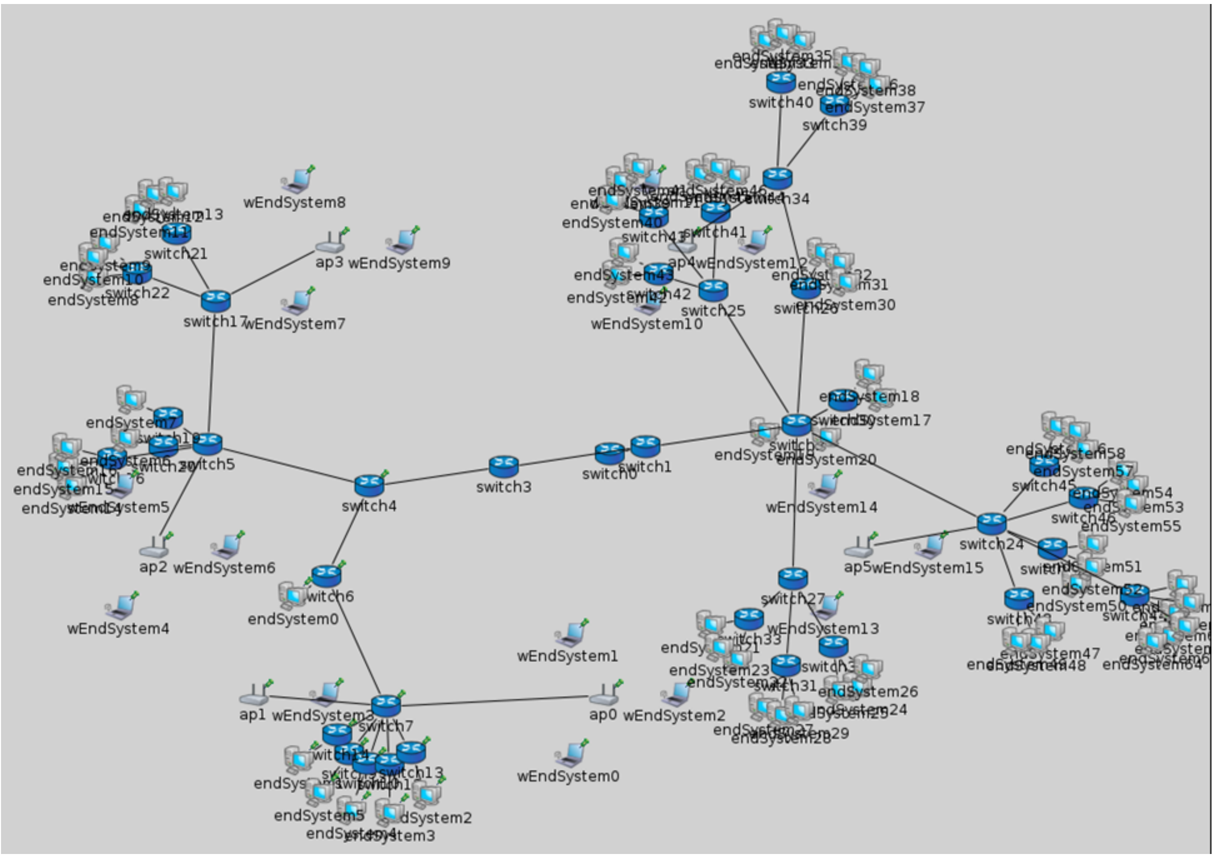

Fig. 2. Example of large network simulated containing 44 switches, 6 access points, 65 wired end systems, 16 wireless end systems, 212 wires dataflow links and 32 wireless dataflow links