WEXTOOL : Wireless Experimentation Tool

Wextool aims to set up, run and make easier the analysis of wireless experiments. It is a flexible and scalable open-source tool that covers all the experimentation steps, from the definition of the experiment scenario to the generation and storage of results.

1. Motivation

Evaluation of network protocols and architectures are at the core of research and can be performed using simulations, emulations, or experimental platforms. Simulations allow a fast evaluation process, fully controlled scenarios, and reproducibility. However, they lack realism and the accuracy of the models implemented in the simulators is hard to assess. Emulation allows controlled environment and reproducibility, but it also suffers from a lack of realism. Experimentations allow more realistic environment and implementations, but they lack reproducibility and are complex to perform. Wireless experimentations are even more challenging to evaluate due to the high variability of the channel characteristics and its sensitivity to interferences. The aim of Wextool is to make wireless experimentations easier to perform and analyze by automating some painful and menial tasks. In this way, the researcher can better concentrate his/her efforts on peculiar research and/or implementation issues related to his/her experimental scenario. Although Wextool was originally created for IEEE 802.11, it can be easily adapted to other type of wireless networks (e.g. Bluetooth, WiMAX) by creating new structures and the corresponding insertion modules.

2. Overview of Wextool

Wextool is based on an experimentation methodology [1] composed by the following steps:

- *Layout and Scenario Definition:*
  This step includes a precise description of the environment, the hardware and software components, the configuration of links and each of the participating devices, such as nodes, routers and access points. With all the devices already configured and tested, the experimentation dynamics must be defined in detail: what data to capture and when, when to generate traffic and what kind of traffic pattern, how many times the experiment must be repeated (i.e., the number of runs), and at which period. Wextool includes a graphical interface to help the experimenter defining the layout and the experimentation scenario. An XML file is then generated automatically, which will be used to automatically configure the nodes involved in the experiment.
- *Platform Calibration:* (yet to be implemented)
  Sometimes, it is important to understand the performance of wireless protocols to characterize the wireless channel and study if parameters are stationary during the experiments or between different runs. On this purpose, measurements using a test pattern can be done before (and possibly after) each run of the experiment.
- *Experiment execution and packet traces retrieval:*
  This step consists of performing the experiment on all the nodes specified in the experiment scenario. Real time monitoring can optionally be used to check the evolution

of key parameters like traffic load and packet loss during the experiment, so as to discover possible anomalies or divergences before the traces processing step. The runs are executed as many times as defined previously. All the devices must have their time bases synchronized (e.g. using NTP), to execute the tasks within schedule and to timestamp the captured packets as they traverse the network. During this stage, raw data from the network is acquired and stored to be further processed. At the end of the experimentation, the different packet traces are automatically sent to a central server.

- *Pre-processing of packet traces:*

  This step performs offline processing of the captured data. It includes synchronizing the packet timestamps from the packet traces, correlating and detecting missing packets and inserting all relevant information on the central database (merging of the data captured at different probes). Other type of statistics such as the ones from wireless drivers captured during the experiment can be inserted to the database at this stage.

- *Data Analysis:*

  Once the central database is built up, the analysis of statistics can start. Wextool includes built-in functions to compute common statistical functions such as temporal parameter computation (e.g. throughput, power, airtime, packet loss), packet loss correlation or cross-layer parameter calculation (e.g. power vs. packet loss). The tool is flexible enough to allow the experimenter to easily add missing functions for his/her specific experiment.

- *Data Packing and Storage:*

  The relevant data is then classified, organized and stored in an easily recoverable way. The package includes the raw data, network layout, system configuration setup and processed results. This will enable researchers to configure the same layout and setup to execute new runs of the same experiment, in order to possibly reproduce it later or at another place.

3. Requirements

- Hardware: A platform composed by wireless-enabled stations/laptops which allow packet capture. A wired infrastructure (e.g. Ethernet) is required for control (i.e. to send the commands, monitor the tasks or transfer the captured data).
- Software: Currently, Wextool has been tested under Fedora Core 8, with the *ssh* NTP services running on each node. Installation of packet capture tools such as *libpcap* and *tshark* are required.

4. Availability

Sources and binaries of Wextool 1.0 are available at http://planete.inria.fr/Software/Wextool/

References

[1] *D. Dujovne, T. Turletti*, *W. Dabbous*, "Experimental Methodology for Real Overlays", Real Overlays And Distributed Systems (ROADS) Workshop, Warsaw, Poland, July 11-12 2007, see http://hal.inria.fr/inria-00326400/en/.