Adaptive Decentralized MAC for Event-Triggered Networked Control Systems

Repeatability Guidelines

Contact: Mikhail Vilgelm
Email: mikhail.vilgelm@tum.de

February 23, 2016

This document is a part of the repeatability package accompanying the paper submission to the 19th ACM International Conference on Hybrid Systems: Computation and Control. It describes the basic steps to reproduce the numerical results of the paper.

1 Overview

Paper title: Adaptive Decentralized MAC for Event-Triggered Networked Control Systems

Authors: M. Vilgelm, M. H. Mamduhi, W. Kellerer, S. Hirche

This package covers the following elements of the paper:

- Figures 3-5
- Figure 7 and 8

Remaining figures were left out as they do not contain any numerical results. Note that we also do not explicitly guide on how to obtain Table 2 values, although it is possible to reproduce them with some configuration changes.

The remainder of the document is structured as follows: in Sec. 2 we describe preliminaries and installation of the necessary software. Sec. 3 explains the process of generating numerical data, and Sec. 4 continues with representing the data and generating figures as in the original paper.

2 Prerequisites

The numerical data for the paper were conducted using the event-based simulation framework Omnet++ [5], and further processed using Python data processing and representation packages.

We provide the instructions for the Ubuntu 14.04 or Ubuntu-based Linux distributions for repeating our results, although it is possible to use it on any platform with proper adjustments to the installation process.

Omnet++ framework. Omnet++ is available for all platforms as an Java/Eclipse-based framework. Its files can be downloaded from [1], and the instructions, along with installation prerequisites are available here [2]. For the user's manual, please see the documentation section on the project's website [3].

After the installation, the project's code needs to be imported from the repository [4]. Note that git needs to be installed to perform the clone:

```
sudo apt-get install git
```

Now, use **git clone** to clone the url in [4]. After the project is cloned, it should be imported into Omnet++. It can be done using the graphical interface of the framework: *File →Import →General →Existing Projects into Workspace*. Select cloned project folder as the root directory, and the project's name should be listed below. Click on *Finish*.

Python framework. Along with the standard Python packages, following needs to be installed: numpy, scipy, and matplotlib. Following command could be used:

```
sudo apt-get install python-numpy python-scipy python-matplotlib \
ipython ipython-notebook python-pandas python-sympy python-nose
```

3 Data generation

In this section we describe the overall structure of the simulation project, and how to use it to obtain numerical data.

3.1 Project structure

The project folder has the following subdirectories:

src/ - contains the source code of the project: (1) NED files describing the high-level modules and network architecture, and (2) C++ source files containing the logic of the modules.

simulation/ - folder with the simulation initialization files

export/ - contains the data resulting from the simulation

scripts/ - contains python scripts used for processing and plotting

3.2 Running the simulation

File **src/omnetpp.ini** contains run configurations along with their main parameters. Figures 3 to 5 have one corresponding configuration, while figure 7 and 8 consist each of three configurations. The configurations are named accordingly. In order to run the simulations and generate the data used for numerical evaluation, you must compile the project (*Project →Build All*), then select the **omnetpp.ini** file, and run appropriate configuration name. To create an appropriate run configuration, go to *Run →Run Configurations*, then create a new configuration as *Omnet++ Simulation* while choosing the *Config name* according to the figure you would like to reproduce. In order to speed up the simulation, it is advised to run it with the command line interface, with *record eventlog* and *debug on errors* disabled. Once you enable command-line interface, you should set *Run number* to *****, and *Processes to run in parallel* equal to the number of cores on the PC (the last step is optional, but allows to finish the simulations faster).

By default, the simulation results are written as ***.sca** files into the **src/results** directory. You can explore the results in the graphical tools of Omnet++, or go to the next step of generating the plots from the results

4 Data representation

Remark. *The scripts for processing the data assume that the names and locations of the simulation results are unchanged. They further assume that all necessary configurations corresponding to the figures were simulated (e.g., all three configurations for figure 7 and all three for figure 8).*

All necessary scripts are contained in the subfolder **scripts/**. There are two auxiliary files **preprocessing.py** and **ci.py**, which are used by the main scripts. In order to produce the plot, you must execute the corresponding python script. For instance, for figure 3 you should issue:

```
python figure3.py
```

To describe the process in short, **figure3.py** first calls the preprocessing function, to compile a single data file from all ***.sca** results (corresponding data will be saved in **export/** directory), and then plots from it the results.

References

[1] Omnet++ Download. `https://omnetpp.org/omnetpp`. Accessed: 2016-01-22.

[2] Omnet++ Installation Guide. `https://omnetpp.org/doc/omnetpp/InstallGuide.pdf`. Accessed: 2016-01-22.

[3] Omnet++ User Manual. `https://omnetpp.org/doc/omnetpp/manual/usman.html`. Accessed: 2016-01-22.

[4] Simulation source code: Adaptive Decentralized MAC for Event-Triggered Networked Control Systems. `https://github.com/mvilgelm/AdaptiveMAC.git`. Accessed: 2016-01-22.

[5] A. Varga et al. The OMNeT++ discrete event simulation system. In *Proceedings of the European simulation multiconference (ESM2001)*, volume 9, page 65. sn, 2001.