Tool for black-box testing OSPF implementations

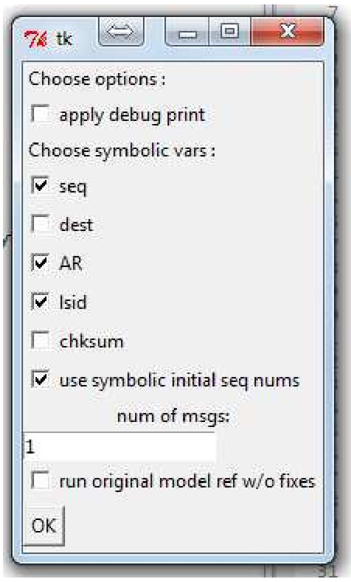
Instructions:

Requirements:

- Linux machine
- Python version 2.7
- Scapy

A) Test generation:

1. Download and install the z3 solver (https://github.com/Z3Prover/z3)
2. Download the concolic execution tool mini-mc (https://github.com/xiw/mini-mc)
3. Copy to the mini-mc folder the files from our tool within: test_generation/mini-mc-OSPF-model/
4. Configure the location folder for the generated tests within the file mc.py by updating the variable *tests_dir.* This is the location to which the generated test files will be written.
5. Run the model.py file, use the following configurations:



Choose options :

□ apply debug print

Choose symbolic vars :

☑ seq

□ dest

☑ AR

☑ lsid

□ chksum

☑ use symbolic initial seq nums

num of msgs:

1

□ run original model ref w/o fixes

6. The destination router can be configured from within the code by updating the variable: *dest_router* within the file model.py . The default is R1.
7. Output: after the run terminates, the output consists of:
   a. The set of generated test files within the specified location (see subsection 4 above)

    b. A summary file of the generated tests, named tests.txt

B) Black Box testing script:

1. Configure the location of the tests in the script testing_script /run_tests.py – assign the folder path of the tests to the variable ***testsdirr*** in the script.
   By default the script runs all tests in ***testsdir***, one by one. (The location should be the same as in A-4).
2. Activate a VIRL simulation with the given topology (‘topology1.virl’ in ‘topology’ folder)
3. Update the script run_tests.py as follows:
    a. Update the variable ‘VIRLHostIP’ to the IP address of the VIRL server host.
    b. Update the variable ‘VIRLGWIP’ to the IP address of the interface of the VIRL host that is connected to the flat network.
4. Run the script on the active simulation, by running:
   sudo python run_tests.py
   NOTE: The script should be run on the VIRL host.
5. After a test failure, if the routers do not recover in the restart procedure from the script, they should be manually restarted from VIRL. Then, the remaining tests should be run. Note that it requires adjustment to run only the relevant remaining tests. (This is to avoid consecutive test failures due to an earlier failure from which the routers did not recover).
   Alternatively, it is possible to configure the script to stop running after any single failure.

C) Deviations Report:

The file *example_script_output.txt* demonstrates the script output while running it with the generated tests. After each test run it is specified if the test passed or failed.

For each test file there is a corresponding out file that logs some information from the run of the test.

In case of a test failure, the log file may help figuring why the test failed (expected routers state vs. actual routers state).

The file: testing_script/ output_log_documentation.txt describes its content.

Alternatively, it is possible to compare the final LSDBs from the test file with the final LSDBs of the VIRL routers.

It may be necessary to re-run a failed test with packet capture to further investigate it.

* In the output example, test4 fails, and test5 fails due to the earlier fail of test4. The routers need to be restarted, and only then the remaining tests can be run.