APPENDIX – REPRODUCING IOF

This tutorial enables the reader to run a single repetition of one experiment using a pre-built Fabrikant topology with a Fabrikant-based MRAI strategy, which produces a convergence graph similar to the one in Fig. 4. Reproducing all the results would require several experiment hours and several testbed nodes, so we limit this tutorial to a single experiment. The on-line resources include the configuration files required to reproduce all the results in this work. Please follow the instructions on our website to run different experiments and to learn how to conduce further research using our tools[8]. The main assumptions for this document are the following:

- The reader has a general knowledge of Linux systems;
- The reader has knowledge of jFed[9] and how to use it;
- The reader has an account on a Fed4Fire+/GENI authority and is able to reserve resources on imec Virtual Walls[10];
- The reader has his own ssh public and private key associated with the iMinds Authority account already configured, respectively in ~/.ssh/iminds.pub and ~/.ssh/iminds.key.

What follows has been tested on GNU/Linux Ubuntu 18.04, to help those running a different operating system we provide a pre-configured Virtual Machine on our website.

First of all, we have to set up the system for the experiment (not necessary on the Virtual Machine). The first thing to do is to clone the repository with the whole project, with the following commands:

```
mkdir ~/src && cd ~/src/
git clone https://github.com/internetonfire/\
iof-tools.git
```

Now you will find inside the `src` folder a new folder called `iof-tools`. Inside this folder you'll find all the resources needed to execute the experiments. As a first step it is necessary to set up the environment, installing all the required software and libraries. This can be done running the following command (again, not necessary in the VM):

```
./configure_env.sh
```

As previously mentioned, we assume that the user has an iMinds account, with the public and the **unencrypted** private keys stored in `~/.ssh/iminds.pub` and `~/.ssh/iminds.key`, respectively. If not, please follow the instructions provided inside `~/src/iof-tools/README.md`.

Inside the iof-tools folder, you will find a folder named *experimentFiles*, where you find the configuration files created by the configuration generator that we used in the paper. Here we use such pre-configured files, but the IoF website and the source repository includes detailed instructions for generating such topologies.

Before launching the experiment, we have to reserve the resources on the Testbed. The tools include a script (`gen-rspec.py`) that can find available resources and generate an .rspec file for jFed automatically. For our Fabrikant experiment, only two machines are needed and the fastest way to reserve them is to use a ready-to-use rspec file inside the repository `utils/2nodes.rspec`. Run jFed launching `jFed-Experimenter` from command line, then open the file inside jFed and click on "Run". Give a unique name to the experiment and once the nodes are available, save the rspec file by clicking on the "Save Manifest" and store it in the iof-tools folder with the name "demo.mrspec".

Now you can set up the experiment environment with the following command (substitute `IMINDSUSER` with your iMinds authority account username):

```
python3 gen-config.py -r demo.mrspec -u \
    IMINDSUSER -k ~/.ssh/iminds.key
```

Now you can install all the software on the nodes (this can take up to 15 to 20 minutes to complete) by running

```
./setup-nodes-environment.sh
```

After the installation and the configuration of the nodes you can deploy your experiment running:

```
cp -r experimentFiles/fabrikant/\
bird-config-files/0_fabrikant_f17n-dest/ .
./deploy-experiment.sh -d \
    0_fabrikant_f17n-dest
```

After the deploy you can start the experiment running:

```
ssh -F ssh-config node0
./run-experiment.sh -a 19 -n 17 -r 1
```

The first argument is the AS that triggers the change, the second one is the link for which the weight is changed (the AS_Id of the neighboor identifies the link), and the third is how many repetitions to perform.

After the experiment is complete, exit the ssh session and fetch the results with

```
./fetch-results.sh
```

This script will take care of copying the logs from the Testbed to your local machine. When prompted by the script whether the results are for a Fabrikant topology, type "y". Now the results can be found inside the folder `iof-tools/RESULTS/`.

Once you have all the logs you can parse them with the script inside the folder logHandlers.

```
cd logHandlers/parser/
python3 log_parser.py \
-f ../../RESULTS/run1/logs/* -c -t > logs
```

With this tool it is possible to average multiple repetitions of the same experiment (see `README.md`). Here we only have a single repetition to plot, and we can do so by running

```
gnuplot -e "outfile='fab.pdf'; \
    inputfile='logs'" \
    ../../plotsGenerator/Gnuplot/plot_logs.gnuplot
```

The output is a file showing the number of ASes that has reached convergence over time, together with the number of `UPDATE` messages, similarly to what is shown in Fig. 4.

[8] https://iof.disi.unitn.it [9] https://jfed.iminds.be - If you dont't have jFed installed you can download it from http://jfed.ilabt.imec.be/downloads/stable/jar/jfed_gui.tar.gz
[10] https://authority.ilabt.iminds.be