CS219 - Network Verification, Synthesis, and Creative Habit - Winter 2019

Project: Analysis of a Network with Batfish

Batfish is a network validation tool that provides correctness guarantees for security, reliability, and compliance by analyzing the configuration of network devices. It builds complete models of network behavior from device configurations and finds violations of network policies (built-in, user-defined, and best-practices). For more information refer to the paper A General Approach to Network Configuration Analysis and the Github page Batfish.

❖ Getting Started with Batfish

Batfish can be used easily with the help of its Docker Container. To get the latest Batfish "allinone" Docker container:

```
$ docker pull batfish/allinone
```

Note: In Linux make sure when Docker is being installed the ownership of the ∼/.local/share/jupyter directory is set to user but not root.

The next thing to do is to create a local data directory. This is a folder on the host machine running the docker container, where Batfish will persist data across container reboots.

```
$ mkdir -p data
```

Start the docker container which will start the Batfish Service. We will use a Python Client to interact with the Service.

```
$ docker run -v $(pwd)/data:/data -p 9997:9997 -p 9996:9996 batfish/
   allinone
```

Next, you need to install Pybatfish (the Python SDK) in order to interact with the service. Batfish documentation recommends to install Pybatfish in a Python 3 virtual environment but for our purpose you can ignore it and just make sure to use Python 3 when using Pybatfish.

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --upgrade git+https://github.com/batfish/
   pybatfish.git
```

People interested in getting familiar with the Jupyter Notebooks provided with the Pybatfish clone the Pybatfish repository and install Jupyter and then start the notebook.

```
$ git clone https://github.com/batfish/pybatfish.git
$ python3 -m pip install jupyter
$ cd pybatfish/jupyter_notebooks
$ jupyter notebook
```

Jupyter notebooks shows how to use Batfish for different analysis and validation tasks. The text and video description of these notebooks can be found in the pybatfish jupyter_notebooks readme.

People can use other methods they are a comfortable with, e.g., an IDE like PyCharm, Visual Studio or an interactive Python shell to skip Jupyter installation. The complete documentation on how to use Pybatfish with Python is available here.

We will be working on the data set similar to the one given in networks/example folder. Place the given **example_Modified** in the networks folder before answering the following questions and set SNAPSHOT_PATH = "networks/example_Modified".

1. SNMP Handling:
   (a) One of the hosts can accept SNMP Requests. Figure out which host it is and explain why it can handle SNMP requests?
   (b) For each router mention whether it can reach the SNMP Host.
   (c) Did you notice any forwarding loops while answering the above question? If there is a loop can you suggest a configuration change to fix it without breaking the DNS Service on Host1?
   (d) After the above fix is the SNMP host reachable in an optimal number of hops from each reachable router?
   (e) If you have decided to stop SNMP services for other ASes(AS1 and AS3)then,
      i. Suggest a configuration change in the control plane.
      ii. Suggest a configuration change in the data plane.
      iii. Which one of the above methods is preferred and why?

   For the next two questions use the original example by setting SNAPSHOT_PATH = "networks/example" and then initialize it again.

2. Use interface properties question to report the pairs of equivalent routers(ex: as2core1 and as2core2) which have differences among their corresponding interfaces. Based on your intuition which properties of interfaces are expected to be different?

3. For each router report the maximum number of other routers(and which ones) that can be down without losing connectivity to the DNS Service on Host1.