OSPF: GENI Extension

Implementation and testing on the GENI network testbed

Lab 6.3: CS3210 - Computer Networks Lab

Instructor: Krishna M. Sivalingam
Due Date: 19th April, 6 PM, 2015
Jan – May 2015, Indian Institute of Technology Madras

1 Overview

This document describes the changes you need to make to your OSPF Local implementation to get it running on GENI, and also describes the GENI setup that you will be using to run your programs.
Please read this document carefully. If you have any doubts what-so-ever, feel free to put them up on the discussion forum.

In the local implementation, you will run N instances of the same process with different parameters (assuming an N node topology), but on the same computer. On the other hand, with GENI, you will ssh into each of the N computers, and run exactly once instance of the OSPF process on each one. The parameters fed to the program instance will depend on the node that it is running on.

2 Implementation Details

This section describes the changes between the GENI implementation and the local implementation.

- The input file `infile` that is fed to the process at each node, will be of the following format.

5	10
0	2
2	3
3	4
4	7
...	

The first entry on the first line specifies the number of routers (N). The node indices go from 0 to $(N-1)$. The second entry on the first line specifies the number of edges.

Each subsequent row contains the tuple (i, j). This implies a bidirectional link between nodes i and j. The minimum and maximum capacity is no longer needed. You will use the actual round trip time (Explained later) of the hello packet to estimate the link capacity metric.

- In the local implementation, the host name that is used throughout is `localhost`, and it was the port number that the process was listening on that defined the node's identity. In GENI, every node should listen on the same port number, say 20000 (not 20000 + i, like in the local implementation). The hostname of the node is `node-x`, where `x` is the node number. `x` is fed as the parameter through `-i` to the program. So when `node-x` wants to communicate with `node-y`, in the local implementation, `node-x` would send UDP packets to `(localhost, 20000+y)`. In the GENI implementation, `node-x` should send UDP packets to `(node-y, 20000)`.
- **Round trip time**: For each HELLO packet that a node-i sends along link i-j, it measures the time taken to receive the corresponding HELLOREPLY packet - RTT **milliseconds**. Now the cost for link_{ij} is set as RTT. Note that this is not bidirectional. cost of link_{ij} need not be the same as cost of link_{ji} at a given instant of time.
- Output: In addition to printing to the file, also print the routing tables to the terminal screen.

The rest of the specification is exactly the same as that of the local implementation, including the format of the output.

3 The GENI framework

We have set up six master slices named `CtOS1, CtOS2, ... CtOS6`. Around 10-15 students have been alloted to each slice. The allocation is available on moodle, and you can also see the slice allotted to you, on GENI (Slice membership). Each of the slices has a spawned-and-ready 7-node topology, which you will use to test your code. The ssh commands to log in to the nodes of the master slice are available in the `Details` tab in the slice. You will ssh into these computers, load your programs into them (having a bitbucket/private git repo with your code and cloning the repo in the GENI node might make life easier than scp-ing each time), and run them. Note that each of the GENI nodes also has access to the Internet.

Members sharing the same slice have been given independent (non-root) accounts on each of the nodes in the slice. For your programs to avoid port-interference with each other, each roll number will be assigned a single port number (20000 + last two digits of roll number Eg., 20060). All your OSPF programs will listen only on that port number for protocol packets.

Do not create any more slices of your own, and do not add resources to your existing slice (named roll_no) without requesting permission from one of the TAs. Your current roll_no slice will expire automatically in a few days. Let it expire; do not renew it unless asked to.

IMPORTANT: Please do not try anything funny. The GENI nodes are to be used for the assignment and for the assignment ONLY. All your actions on each of the nodes is logged, including information about all the traffic flowing through each of them. Any misuse of the resources could result in your account being revoked, and a zero credit in the GENI part of the assignment.

4 What to Submit

Create a folder `Lab6-CS1xBabc`, the main directory.

- Source code files
- Corresponding Output files, showing the routing table entries for all routers.

5 Grading

- Local Implementation: 75 points
- GENI Implementation: 25 points

Note: This is an INDIVIDUAL assignment. If you are having difficulty with the assignment, please talk to the TAs/the instructor. Downloaded Code from the Web (not even for Dijkstra's algorithm) will NOT be considered for grading and such action will lead to academic penalties.