CISC 861 – Fall 2011

Programming Assignment – PARSEC Simulation
Due: 9/30/2011 (Friday) by 11:59pm

1 Overview

In this programming assignment, you will develop a version of the *GOSSIP1(p,k)* gossip protocol [1] in the PARSEC language [2], and simulate its behavior for different values of p and k. The details of *GOSSIP1(p,k)* is described in the first 4 paragraphs of Section II on the 2nd page of [1].

2 Requirements

2.1 Simulation Setup

You will simulate a network of 400 nodes located at an evenly-spaced 20×20 grid as shown in Figure 1. Every node is assigned a unique ID of the form (x, y) that is corresponding to its position on the grid topology. We assume that the radio transceiver on each node is capable of directly communicating with four other nodes on the four different directions. I.e., a packet broadcast by a non-border node (x, y) is received by nodes $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, and $(x, y-1)$.

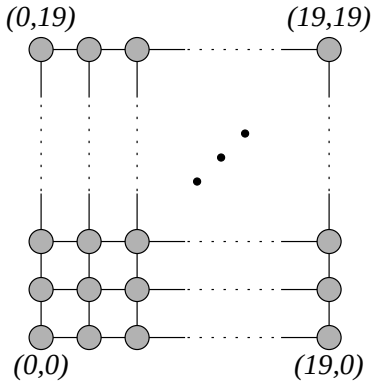

Figure 1: Simulated Network Configuration.

The node (6,6) is chosen as the source which generates 50 packets to be delivered to all the other nodes in the network via the Gossip protocol.

2.2 Simulation Results

There are no constraints on the format of the raw outputs generated by the simulation executable. However, you should be able to process the outputs by some means (e.g., awk/perl scripts, etc.) and plot the following graphs:

1. **Delivery ratio as a function of gossip probability.** For each execution of the simulation (with a different seed), the delivery ratio is calculated as the number of non-duplicate packets successfully received by the receivers divided by the expected number of (non-duplicate) packets to be received. The x-axis represents different gossip probabilities, while the y-axis is the delivery ratio. Multiple (3 to 4) lines should be shown for different values of k.
2. **Delivery ratio as a function of hop distance to source.** This is similar to the first graph except that the x-axis represents distances from the source in terms of number of hops. Since more than one node may have an equal distance from the source, the delivery ratio for a particular distance is the average among all the receivers located at that distance. Multiple lines are shown for different (p, k) settings.
3. **Broadcast reduction ratio as a function of gossip probability.** This represents the ratio of the actual number of broadcasts versus the number of broadcasts expected when the gossip probability is one. Again, multiple lines should be shown for different values of k.

3 Submission

You are required to submit all the source files, a README file, and a Makefile via email in the form of a uuencoded tar file. In particular, the README file should contain a brief description on your simulation program's operational interface. The following steps explain the submission procedure.

1. Put all the source files of your project #1 into a subdirectory named `<your_login_name>_1`.
2. At the current working directory, do

   ```
   tar cvf <your_login_name>_1.tar <your_login_name>_1
   ```

3. Mail the file `<your_login_name>_1.tar` to `cshen@cis.udel.edu` as an attachment with `Subject: PARSEC submssion`.

You must submit your files as described above. Other formats will be rejected.

You will also submit a **report** describing the design of your simulation program, including message types used, internal data structures, and graphs presenting simulation results with discussion. (Please do NOT attache the PARSEC program code.)

4 Grading

You will be graded on the correctness of your code (80%), and its in-line documentation, readability, and structure (20%).

5 PARSEC Usage

- PARSEC is installed at `/m/degas/research/simulators/parsec/solaris` for CIS **Solaris sparc** machines, such as `stimpy` (`mudskipperN`).
- Set the environment variable `PCC_DIRECTORY` to `/m/degas/research/simulators/parsec/solaris`
- Add `$PCC_DIRECTORY/bin` to your path.
- Make sure that you use the `gcc` compiler located in `/usr/local/gnu/gcc/bin/gcc` on CIS machines.
- Use command `pcc` to compile PARSEC programs, say `gossip.pc`.
- You will need to change the maximum stack segment size before running your program by executing the following commend.

```
$ ulimit -s 200000  % increase stack size limit in ksh
$ ulimit            % increase stack size limit in csh
```

References

[1] Zygmunt J. Haas, Joseph Y. Halpern, and Li Li. *Gossip-Based Ad Hoc Routing*. In Proceedings of IEEE INFOCOM 2002, New York, NY, June 23–27 2002.

[2] R. Meyer and R. Bagrodia, *PARSEC User Manual*, `http://www.cis.udel.edu/~cshen/861/notes/manual.pdf`