Lab 2: CONGA in htsim
CS740 Fall 2024

Released: Oct 10 2024
Due: Nov 6 2023 11:59 PM
Total points: 15

Important notes

- All answers and code that you write for this lab must be your own work. Plagiarism is a serious violation of university statutes and all answers will be thoroughly checked for it.
- Ensure that your answer has proper headings and subheadings wherever necessary.
- Late policy:
    - Up to 30 minutes late – lose 0% points
    - Up to 24 hours late — lose 10% of points
    - Up to 48 hours late — lose 30% of points
    - Up to 72 hours late — lose 60% of points
    - Beyond 72 hours — lose 100% of points

Overview

`htsim` is an open-source network simulator. It is designed to simulate large-scale networks and analyze their performance characteristics (such as flow completion time, throughput, packet loss). `htsim` has a range of features, including support for custom network topologies, multiple traffic sources and sinks, and a variety of routing and congestion control algorithms. It also provides a flexible and extensible architecture that allows researchers to customize the simulator for their specific needs.

Learning Outcomes

After completing this lab, students should be able to:

- Understand the FatTree/Close based data center network architecture;
- Understand how `htsim` works and use `htsim` to perform simulation;
- Design and implementation CONGA in `htsim`;

Part 1: Testbed setup

You can find the `htsim` source on canvas files

- **wget https://pages.cs.wisc.edu/\~mgliu/CS740/F24/labs/lab2/lab2-skeleton.tar.gz**

`htsim` walk through

The key components of `htsim` include the

- Workload, which determines flow size based on workload type and distribution such as Pareto, uniform, enterprise, or data mining;
- Flowgenerator, which generates flows with a given flow rate and size and creates TCP connections between the source and destination;
- Networking primitives, which include Queues and Pipes used to construct the packet testbed;
- Eventlist, which schedules events for the simulation;
- Testbed, which is used for the simulation.

`htsim` is a network simulator that operates on events rather than mimicking physical switch behavior. For example, when a packet is ready to be sent from a queue, instead of immediately transmitting it, the simulator schedules a sending event to occur after a calculated time based on the packet size and linkspeed. The simulator's behavior is considered to be discrete.

You can use the following two examples to understand how the simulator works. It will also help you understand how to build the CONGA environment.

- Test_single_link.cpp
- Test_fat-tree_testbed.cpp

To run `htsim`, just:

- **make**

And the executable will be in the same directory. You might encounter some compiling issues, like missing headers. Check the clang version and see what is missing (e.g., libstdc++-12-dev).

- **clang -v**

Part 2: CONGA implementation

- Your task is to implement CONGA in `htsim`.

- Network configuration:
  - 12 Core switch with 24x40Gbps port
  - 24 leaf switch with 12x40Gbps uplink port and 32x10Gbps server-facing port.
  - Each server will just be connected to 1 leaf switch with a 10Gbps link

- You need to implement the following:
  - Build the simulation testbed based on the given configuration and sample testbeds (fat-tree and single-link)
  - A distributed congestion awareness mechanism as CONGA provided
    - You may need to define an leafswitch.cpp (leafswitch.h) to maintain the Congestion-to-leaf table and congestion-from-leaf table.
    - The leaf switch will be able to determine which uplink to take with the local table.
    - The congestion information exchange mechanism as described in CONGA (congestion feedback)
    - A way to determine the extent of congestion along the packets' path. (DRE or other similar approach that serve the objective)

- The following pieces from CONGA are not required:
  - The flowlet, so you can just assume the flow will work as flowlet in CONGA paper.
  - VXLAN tunneling: you don't have to follow the exact header definition mentioned in the paper as long as you can encode the necessary information.

Part 3: Performance Report

You will evaluate your own implementation in detail and compare with the ECMP. You need to report the following results for four workloads (i.e., uniform, pareto, enterprise, dataming):

1. Flow completion time for all flows varing the load for 10/20/30/40/50/60/70/80/90/100 % cases;

2. Flow completion time for small flows (<100KB) varing the load for 10/20/30/40/50/60/70/80/90/100 % cases;

3. Flow completion time for large flows (>=100KB) varing the load for 10/20/30/40/50/60/70/80/90/100 % cases;

Specifically, you will provide 12 figures (3 figs for each workload) with load (from 10%-100%) as x-axis and the average flow completion time (FCT) as y-axis. You can find a similar figure in the CONGA paper (fig10). Please compare with ECMP and discuss why your solution outperforms/underperforms ECMP.

==========

Submission Instructions

You must submit(one submission for each group):

- The skeleton code with your implementation including the client and server source code along with their makefiles to compile them respectively.
- Your measurement results and answers to the questions from Part 3—all measurements and answers to questions in the file `report.pdf`
- A `README` file with the names and CS usernames of all group members, assumptions made, and how to run your application

You must submit a single tar file containing the above. To create the tar file, run the following command, replacing username1, username2 (just username1 if you are doing the lab by yourself) with the CS username of each group member:

- `tar czvf username1_username2.tgz htsim report.pdf README`
- `tar czvf username1.tgz htsim report.pdf README`

Upload the tar file on the Lab2 tab on course's Canvas page. Please submit only one tar file per group.