MP0: Event Logging Report

Paul Pok Hym P Ng (ppng2)
Bruno Seo (sbseo2)

February 9, 2020

1 Cluster number

Our cluster number is 15.

2 Instructions for running code

Please follow the instructions written as below.

Server Node

1. Upload logger.py to your main server node.
2. Run logger.py by using the following command.

```
    $ python3 logger.py <N>
```

$\langle N \rangle$ represents the number of nodes you wish to log simultaneously.
3. Wait client nodes to be connected
4. Once all clients have disconnected the logger will wait a few seconds and begin dumping the information into textfiles in the same location as `logger.py`

Client Nodes

1. Upload generator.py and test-client.py to each client node.
2. Run the following command.

```
    $ python3 -u generator.py 0.1 | python3 test-client.py
```

0.1 represents 0.1 Hz so once every second.
Use Ctrl+C to terminate the client.

Graphs

All graphs that were generated are are generated via the `*.ipynb` files in the repository. The graphs have already been pre-generated there so there is no reason to re-run. However, if one desires to do so they would only need to change following variables and recalculate.

1. `NUM_OF_NODE` : This represents the number of nodes
2. `TIME`: The amount of time the test was running for

Then one only needs to re-run the code in each section.

Please keep in mind the following. In order to ensure that fresh data is always used, please delete all previous run's node text files to ensure no data gets leaked into the new run.

3 Delay and Bandwidth Measurement

Once all N nodes are terminated, then our logger will generate N files in the logger directory labeled nodeN.txt.
Each contains logs for each node. The data format is as follows.

```
nodeTime nodeData loggerTime bytesTransfered
```

nodeTime is the time when the msg was generated on the node, *nodeData* is the data sent, *loggerTime* is the time which the logger received and parsed the msg, and finally *bytesTransfered* is the number of bytes that was sent by a particular node.

Delay

We measured delay by using the difference between the timestamp of server and the timestamp of the client by each second. In short, we used *loggerTime - nodeTime.*

Bandwidth

In terms of bandwidth, we obtained it by sum of *bytesTransferred* on each second.

Graphs of the evaluation

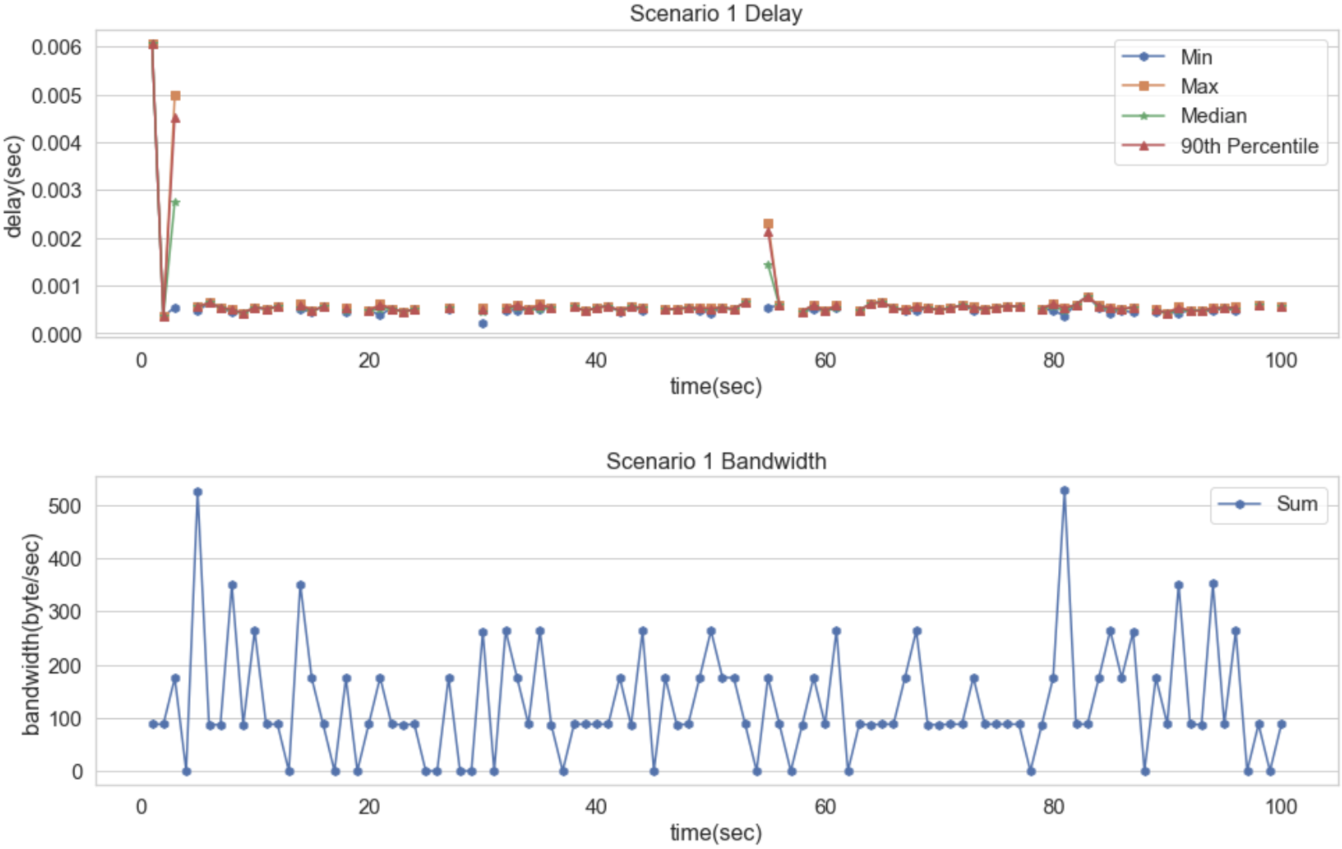

Figure 1: 3 nodes, 0.5 Hz each, running for 100 seconds

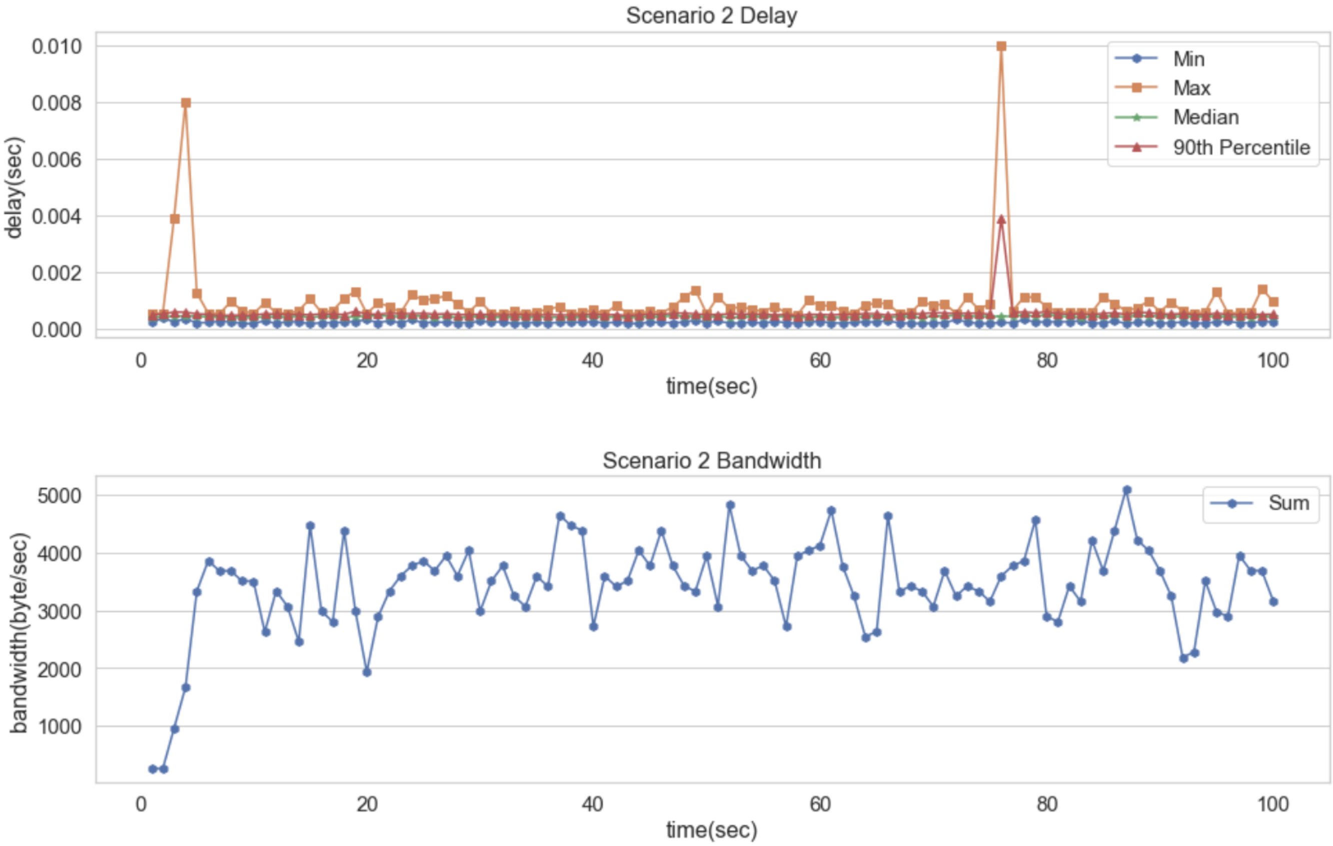

Figure 2: 8 nodes, 5 Hz each, running for 100 seconds