

Dumb Distributed Network

Quinn Gieseke (V00884671), CSC 466

April 6, 2021

1 Final motivation

- Monitor status of sensors
- To allow for every smart node to have as current a value for every sensor in the network as possible.
- Ensure that data is transmitted reliably between smart nodes.
- Ensure that data becomes stale quickly.
- Must ensure that stale data does not persist, ie gets distributed between nodes.
- Be scalable.
- Have low overhead.

2 Sensor Description

- Produces single integer value for data (can be expanded later).
- Monotonic counter per data output.

2.1 I/O

Write only datagram.

3 Smart Node Description

- Monotonic counter per sensor read.
- Keeps state for every sensor
 - Last data value
 - Last sensor tick
 - Approximate last node tick
- Bitmap for stale sensors

3.1 I/O

Dedicated threads for sensor reads and node reads.

Produces a heartbeat pulse every second which includes(for every up to date sensor):

- Sensor data
- Sensor tick for data
- Negative offset for node tick

4 Simulator Description

- Routes all information through named pipes
- Maps locality, so that sensors and nodes only communicate through spacially near nodes.
- Incurs message drop at set rate.

5 Tuneable parameters

- Number of nodes and sensors
- How long it takes for a bit of sensor data to become stale (in Node Ticks)
- Maximum size of spacial map
- Maximum distance for sensors/nodes to be able to transmit (Ideally recieve roughly full coverage between the two)
- Drop rate for communication.

6 Results

Each smart node is capable of sustaining 4200 sensor reads per second (scaled to 4 smart nodes).

To get around this, each sensor created new results at 1000 times per second, instead of upping the number of sensors to reduce operating system overhead.

All further tests were restricted by my CPU or simulator code, not the actual protocol.

7 Further steps

8 URL

https://github.com/QGieseke/CSC466_Distributed_Network