Dumb Distributed Network

Quinn Gieseke (V00884671), CSC 466

March 8, 2021

1 Problem statement

As computers have miniaturized and grown more efficient, networks have needed to advance to include many computing units of varying power and complexity. One particular issue that has arisen is existing network protocols are ill suited to large networks of low power computers, as seen in such instances as smart homes and distributed sensor networks. The relative scarcity of stable links and a heavy reliance on power savings means that protocols like TCP become nearly unusable.

2 Existing Solutions

As interest over IOT devices has grown, many protocols have risen to attempt to overcome these challenges. 6LoWPAN is an extension on IPv6 to allow sporadically connected devices to efficiently connect. Z-Wave is a closed source mesh network relying on low-power radio transmissions instead of the more common WiFi or Bluetooth standards, giving it some advantages in data transfer rates. Zigbee is a similar protocol, a main competitor.

A major drawback for each of these existing standards is that they each require fully featured microcontrollers at every node in the network in order to receive, transmit, and broadcast messages in order to maintain network connectivity. This requirement necessarily limits the simplicity of edge nodes due to required computing power and energy consumption needed.

3 My Approach

The main idea behind my approach is to break one of the fundamental assumptions of a network, and that is removing the need for two-way communication between edge nodes. By requiring each distributed sensor to only broadcast, and never needing to listen to replies, both the necessary power consumption and computational power should plummet, enabling the use of truly tiny and low power solutions.

Of course, the network as a whole must still be smart, so I propose a 2-layer solution, where the dumb edge nodes simply broadcast their status, and smarter internal nodes receive those broadcasts and pass them on to their intended destination. However, since smart internal nodes may be relatively scarce in comparison to edge nodes, I believe this is a useful trade-off. Internal nodes would be responsible for:

1. Reliable transmission of data through the internal network
2. Monitoring of status of edge nodes
3. Flexible receiving of edge node transmissions (no strict mapping between edge and internal nodes) [Cleared]
4. High scalability for receiving and transmitting data. [Cleared]

An additional goal I have imposed upon myself is to allow the smart nodes to be entirely mobile within the network. This has caused me to increase the scope of the custom simulator I am using, as well as complicating my plans for the smart-node algorithm. Essentially network partitioning becomes even more likely, but also much of the data being collected has limited value over time, so aggregation measured per smart-node are also a consideration.

The original motivation for this network was for a factory with an extensive sensor network being able to handle all the input. However, during the process I realized that a similar network could also be used in circumstances where the network is not fixed (perhaps communicating over radio), but there is a high sensor concentration in the area anyway, such as during ecological studies, a modern battlefield, or any other outdoor application. However, this mostly undermines my original assumption that all smart nodes would be working to pass the data reliably out of the network into the internet as a whole, and instead must work instead to update as many peer nodes as possible with the relevant information while overcoming much more intermittent networks.

4 Rough timetable

- 1-2: Research + Formalization of goals [Cleared]
- 3-4: Finalize dumb node design and formulation of smart node design [Expanded Scope]
- 4-6: Finalize smart node design.

4.1 Added tasks:

- Create simulator for asymmetric network & node reorganization [Cleared]
- Make it multithreaded [Cleared]

5 URL

https://github.com/QGieseke/CSC466_Distributed_Network