GPS Mesh Peer-to-Peer Network SRS

By:

A.C.

R.G.

I.M.

N.N.

Version: 1 Date 11/14/2017

Table of Contents:

Rev. History:

GPS Mesh Peer-to-Peer Network | v1 | 14 Nov 2017

GPS Mesh Peer-to-Peer Network SRS | v2 | 30 Jan 2018

1. Introduction:

Purpose:

The purpose of this document is to outline the software requirements for a GPS peer-to-peer (P2P) mesh network using Raspberry Pi0w units in conjunction with a compact GPS module and battery pack or power source.

Conventions:

We will refer to the peer to peer network simply as the GPS-mesh or mesh. Additionally, we will refer to the Raspberry Pi0ws combined with the GPS unit and power source simply as a node.

Audiences and Reading Suggestion:

The audience for the document is the designers and developers of the GPS-mesh, their project manager Dr. Bover, the product customer Dr. Clauson, and any future teams that may work on this project.

References:

Vision and Scope document

https://www.open-mesh.org/projects/open-mesh/wiki - batman reference(NO LONGER USED)

2. Overall description:

Product Perspective:

The nodes will be a niche product for sailboat race tracking. The mesh will be composed of a peer to peer network using raspberry Pi0ws with GPS chips to create the nodes in the network. The nodes will send and collect GPS location information from each other using the on board Wi-Fi adapters functioning in what is referred to as ad-hoc mode. There will be one observing computer collecting data from all nearest nodes. This observer will also need to have a Wi-Fi chip capable of operating in ad-hoc mode. The scope of this project encompasses both the functionality of the nodes and the stationary observer.

Product Features:

1. Peer to peer communication using Wi-Fi adapters.

2. GPS tracking for individual nodes.

3. Race replay functionality.

User Classes and Characteristics:

Boat races in the bay.

Key Functions:

- Raspberry Pi0w's maintain peer to peer network.
- Raspberry Pi0w's use GPS module to determine position.
- Raspberry Pi0w's broadcast their current known network state every 2 seconds.
- Raspberry Pi0w's collect GPS coordinates from other nodes and update the unit's stored network state, then broadcast this state to its neighbors every 10 seconds.
- Stationary observer computer collects and displays current network state.

Key Requirements:

- All devices require network cards capable of ad-hoc mode operation.

Operating Environment:

Nodes will be on the deck or mast of a boat sailing in the local bay and potentially the ocean. Nodes will be exposed to wind, salt water and rain water during regular use. It is assumed that the observer computer will be protected from wind and weather by being located in an interior location, whether that is the cabin of a ship, or in a building or in a vehicle on land.

Design and Implementation Constraints:

- Nodes will have a connected GPS node.
- Nodes will have no display.
- Nodes will have Wi-Fi capability.
- Nodes will operate in 'ad hoc' mode.

- Nodes may lose connection, but should still update their own data with available input from its self and neighbors. It will broadcast its coordinates every 2 seconds, and network state every 10 seconds.
- Nodes will need to be battery powered, with enough stored energy to last the duration of a sailboat race. Estimated at 30 minutes to 48 hours.
- Nodes will be run on Raspberry Pi0w running Raspbien.
- Observer will be laptop computer running a Linux based OS.
- Assumptions and Dependencies:
- Assume nodes will have ample power for the duration of the race.
- Peer to peer communication will be connected via Wi-Fi adapters in ad-hoc mode.
- Depends on retrieving GPS information from the GPS node.
- The observer will need to come into range of at least one node by the end of the race to gain race information.

User Documentation:

User Manual – Will describe power on and set up procedure for the nodes and observer to create the mesh. Potentially will include some troubleshooting steps should something go wrong.

3. System Features

3.0 Peer-to-Peer Connection

3.0.1 The system shall have the Pi0w nodes connect and communicate directly using the batman-adv routing protocol. Top priority.

3.0.2 Nodes shall send data, receive data and update telemetry from data received other nodes

3.0.3 The nodes shall send data to all nodes in range when it is connected to the mesh.

3.0.4 The nodes shall record data even when not in range of other nodes in the mesh.

3.1 Observer Display

3.1.1 There shall be an observer (device with a screen) that will serve as a master record. Second Priority

3.1.2 Pi0w nodes will relay data to observer, observer will be ready to receive data from Pi0w.

3.1.3 The observer will display the most current positions of nodes.

3.1.4 The observer could display a history of recent positions of nodes.

3.2 Node Tracking

3.2.1 Each node will track its own position at a prespecified time interval using the connected GPS module.

3.2.2 Will store its own position as well as other node positions in a local data structure on disk.

3.2.3 Will delete data that was logged more than 24 hours ago.

4. External Features:

User Features:

The observer will display nodes position time. Potential GUI to track and display node data in real time. The nodes will not have a display. The nodes should have a power button and and indicator light to display status whether connected or not connected to the mesh.

Hardware Features:

Nodes will use GPS modules to obtain GPS position and speed. Nodes will be housed in weatherproof containers.

Software Features:

Nodes will broadcast data to all other connected nodes in real time. Nodes will dynamically connect and disconnect from the mesh as they come within range of other nodes. These newly connected nodes will automatically receive future updates from the rest of the mesh.

Communication Interface:

Nodes and observer computer will communicate over UDP to relay positional information about nodes.

5. Other Non-functional Requirements:

Performance:

The nodes should connect to the observer within 30 seconds of activation.

Safety Requirements:

Hardware protective casing must withstand non-static water environments.

Quality Requirements:

Availability: Nodes should be available and working during the entire length of a race or testing session. This could be anywhere from 30 minutes to 48 hours.

Usability: To ensure our system is easily useable, we will time a new user to see how long it takes them to start up the system (or a small sub-slice of the system i.e., 2 nodes connecting to observer). Assure the new user is able to complete the connection and setup process for two nodes and the observer in less than 5 minutes.

Integrity: Ensure good cases/enclosures to preserve physical integrity of the system. In operation, our mesh will only allow the preconfigured nodes to connect to each other, and the observer.

Reliability: The system should always deliver correct messages, if it is in range of other nodes. Out of range nodes will update their information once they have gotten back within range of the mesh and observer.

Efficiency: Nodes should have minimal delay between transmission/receiving. A message should be received (if any nodes in range) while the data in message is still correct on the sender. Increase/decrease in send/ping interval should allow this to be achieved easily. It will heavily depend upon how long processing GPS input takes.

Maintainability: We will ensure maintainable code by explicitly commenting the code functionality and documenting each module. After modifying and creating a stable version a complete list of changes and fixes will be added to the wiki.

Portability: Our devices are Pi0w's, the code should be able to be applied to a different model of Pi and still function properly. The observer software should be able to run on any modern OS (Windows, UNIX-based systems, and OS X). For bonus, observer software could be a phone app, in which case it would need to be useable on Android and iOS.

Reusability: Have a portable container so it can be used to plot coordinates for any sets of vehicles capable of carrying the device.

Testability: Every line of code can be tested to assure correctness/accuracy. Will need to figure out how to check and be sure GPS coordinates are correct.

Security Requirements: Encryption and decryption of broadcast messages at Application level. Message securing protocols established during setup.

6. Other Requirements:

Appendix A: Glossary

GPS-mesh/mesh:	Peer-to-peer network constructed by nodes using UDP communication protocol.
Node:	Raspberry Pi0w with GPS module in enclosure.
Observer:	Portable device with screen to display node information. Will have Wi-Fi adapter capable of operating in ad-hoc mode.
System	Combination of nodes and observer connected and communicating.