

Pedestrian density and flow tracker using Raspberry PI

Abhishek Dubey
Assistant Professor
Computer Science and Computer Engineering
Vanderbilt University

Senior Research Scientist
Institute for Software Integrated Systems



Tel (615) 343-7472 Fax (615) 343-7440
1025 16th Avenue South | Nashville, TN 37212
www.isis.vanderbilt.edu



VANDERBILT UNIVERSITY

The problem

- Understand the density of pedestrians in a region
 - Defined the amount of people per unit of area within a certain time interval
- Understand the flow of pedestrians between two locations
 - Defined as the number of people moving one way through an area of interest within a certain time interval.

Why do we want this information?

- Optimize the traffic flow based on the expected pedestrian traffic.
- Automatically adjust the signal timings based on the crowd density near a traffic intersection.
- There are many other uses.

How do we do it?



Everybody has one of these

How do we do it?



Everybody has one of these

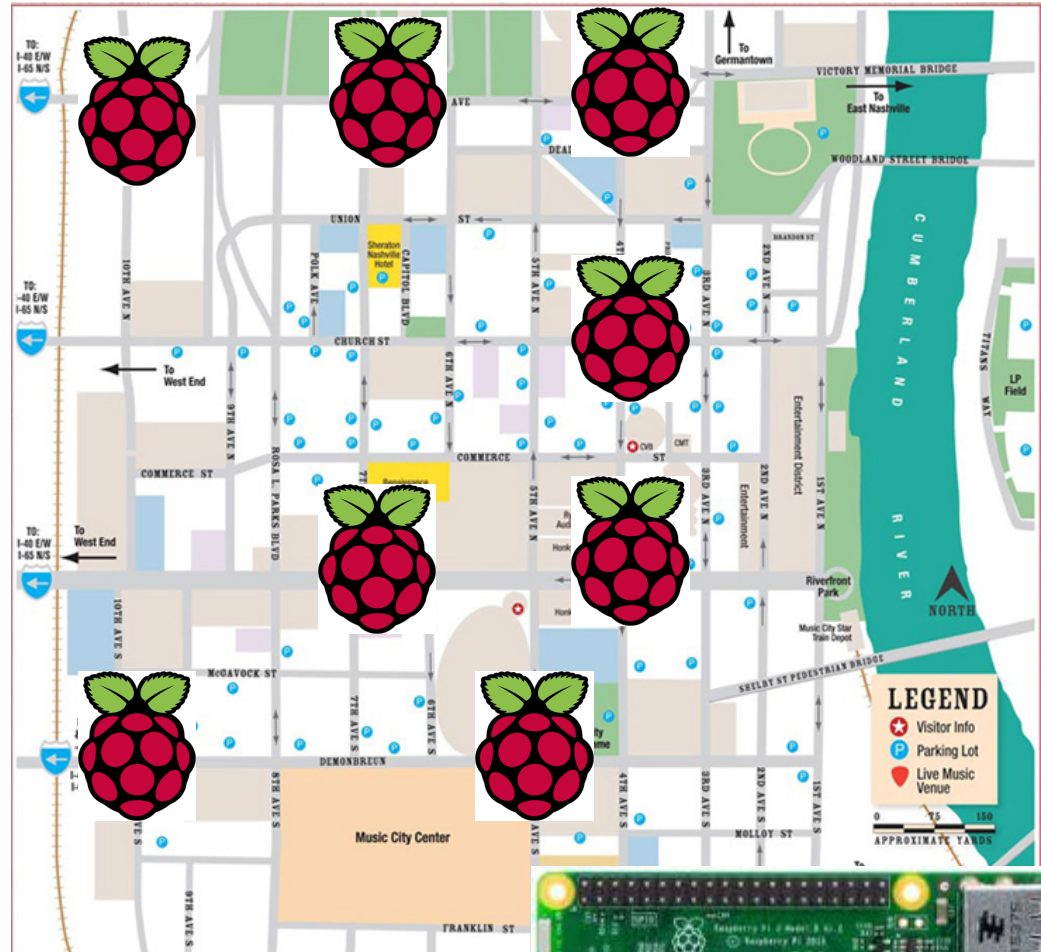


And they are connected. They also have a unique network address

How can we do it?



Track the address using a network of raspberry pis

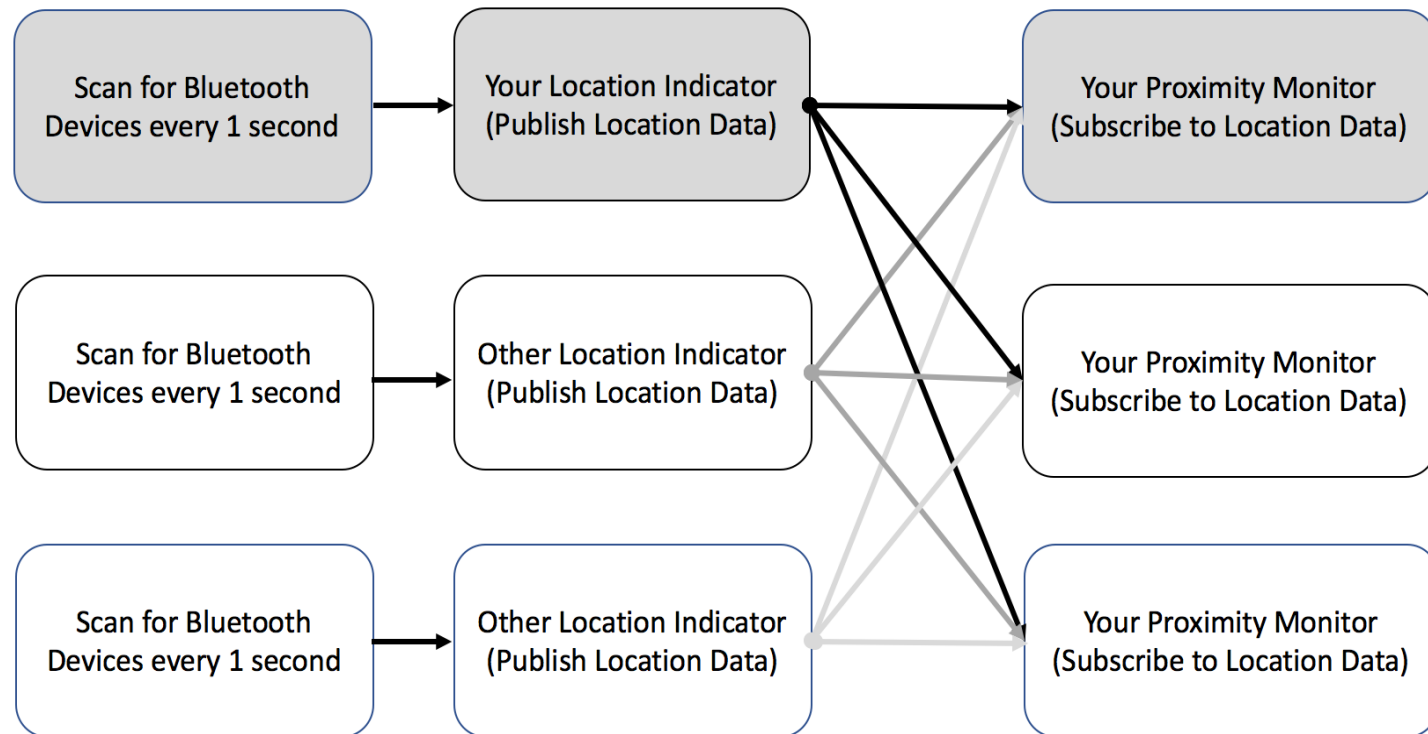


Questions?

Location Announcer Example

- Application
 - Location Indicator
 - Scans for Bluetooth devices every second
 - When device is found, publishes the information for others to see (LocationData)
 - Proximity Monitor
 - Listens for LocationData published messages (in other words, subscribes to these messages)
 - Discovery Service will match Publishers with Subscribers based on message labels (such as Location Data)
- Location Data
 - Unique ID for Bluetooth Device (media access control address or MAC), labeled as 'hash'
 - Computing Device ID, labeled as 'location'
 - Time when Bluetooth Device is found by computing device, labeled as 'timestamp'
 - Using this information each device can compute the flows and density.

Location Announcer Example (cont.)



Location Announcer Example (cont.)

To run the application, open three separate terminal windows

1. Start Discovery Service using the
 - `./rundiscovery.sh` [from the blue observer-collector folder] in a new terminal.
2. Start the Location Indication component (in LocationTransmitter actor)
 - `cd riaps-app; ./runtransmitter.sh`
3. Start the Proximity Monitor component (in LocationReceiver actor)
 - `cd riaps-app; ./runreceiver.sh`

Devices Found (Location Indicator)

```
root@raspberrypi:/home/riaps/blueobserver-collector/riaps-app# riaps_actor LocationAnnouncer LocationAnnouncer.json LocationTransmitter
INFO:17:15:20,821:LocationIndicator:(PID 7294)-starting LocationIndicator, Fri Jul 28 17:15:20 2017
ERROR:17:15:21,833:riaps.run.devc:No response from devm service Resource temporarily unavailable
INFO:17:16:25,931:LocationIndicator:7294:found{'timestamp': 1501280185.9272106, 'hash': 'D0:25:98:6A:68:F3', 'location': 'pi3'}
INFO:17:16:37,143:LocationIndicator:7294:found{'timestamp': 1501280186.4210649, 'hash': '9C:20:7B:F2:64:7B', 'location': 'pi3'}
INFO:17:16:44,392:LocationIndicator:7294:found{'timestamp': 1501280186.4210649, 'hash': '2C:B4:3A:08:78:54', 'location': 'pi3'}
□
```

Received Indication of Devices Found (Proximity Monitor)

```
riaps@raspberrypi:~/blueobserver-collector/riaps-app $ sudo -E riaps_actor LocationAnnouncer LocationAnnouncer.json LocationReceiver
[sudo] password for riaps:
INFO:17:16:11,108:ProximityMonitor:(PID 7316)-starting ProximityMonitor, Fri Jul 28 17:16:11 2017
ERROR:17:16:12,115:riaps.run.devc:No response from devm service Resource temporarily unavailable
INFO:17:16:25,933:ProximityMonitor:7316:on_proximityupdate {'hash': 'D0:25:98:6A:68:F3', 'location': 'pi3', 'timestamp': 1501280185.9272106}
INFO:17:16:37,146:ProximityMonitor:7316:on_proximityupdate {'hash': '9C:20:7B:F2:64:7B', 'location': 'pi3', 'timestamp': 1501280186.4210649}
INFO:17:16:44,394:ProximityMonitor:7316:on_proximityupdate {'hash': '2C:B4:3A:08:78:54', 'location': 'pi3', 'timestamp': 1501280186.4210649}
□
```



Exercise

- Extend the model to identify and announce to all when a specific id has been seen.
- Steps to follow
 - declare a new message called announcement
 - Create a new publisher called announcer (message announcement) in the LocationAnnouncer/pmonitor.riaps Location indicator component.
 - Create a new subscriber called announcer in ProximityMonitor to receive the message.
 - Extend the process_observation method in Location Indicator.py publish a message when a specific tag [see whiteboard] is found.
 - The publish message step will be self.announcer.send_pyobj(msg) where msg has to be constructed like
 - msg="found tag"
 - Implement the on_announcer method in ProximityMonitor.py. See the line observation = self.proximityupdate.recv_pyobj() in the file to check how to receive the message.

Questions?