Argus

Senior Design II
Spring 2019

General Information

Team Members

- Matt Ihlenfield: ihlenfmt@mail.uc.edu
- Anthony Jantzen: jantzean@mail.uc.edu
- Pat Millott: millotpg@mail.uc.edu
- Kyle Trout: troutkt@mail.uc.edu

Project Adviser

- Dr. Dharma Agrawal

Abstract

- More and more devices are emitting wireless traffic
  - Having access to these devices' locations can be extremely useful
- GPS is inefficient and unreliable
- Argus collects location information using devices' WiFi connection
  - Uses signal strength and time of flight to obtain an accurate reading

Project Background

Purpose

An application that will allow for indoor location tracking where GPS is not reliable, providing a cross-platform user interface.

Goals

- A listing of devices on a network including MAC address, hostname, and manufacturer
- For each device, it's approximate location listed, as well as a visual rendering
- Live cross-platform interface with location mapping and notifications
- Location accuracy within 3 m

Intellectual Merits

- Locates devices accurately indoors
  - This technology is not widely available as GPS is unreliable indoors.
- Provides streamlined interface to both monitor and manage devices.
  - This will use a heatmap overlay on the device mapping view to show hot spots of activity.

Broader Impacts

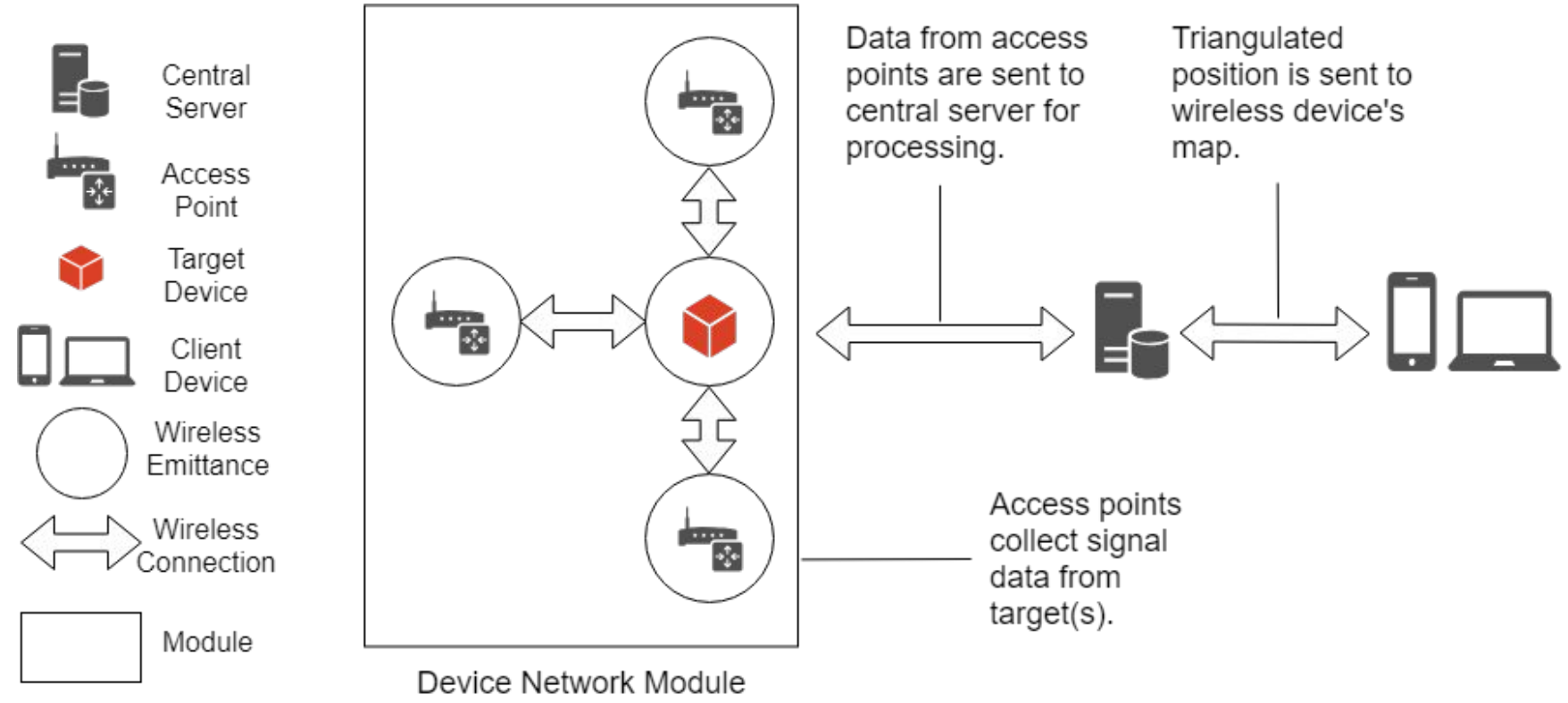
- Efficiency
  - Increases efficiency for conference or event planners to alleviate hot spots of congestion
  - Eases troubleshooting of devices for network administrators.
- Security
  - Increases monitoring capabilities for IT specialists on IoT networks.

Design Specifications

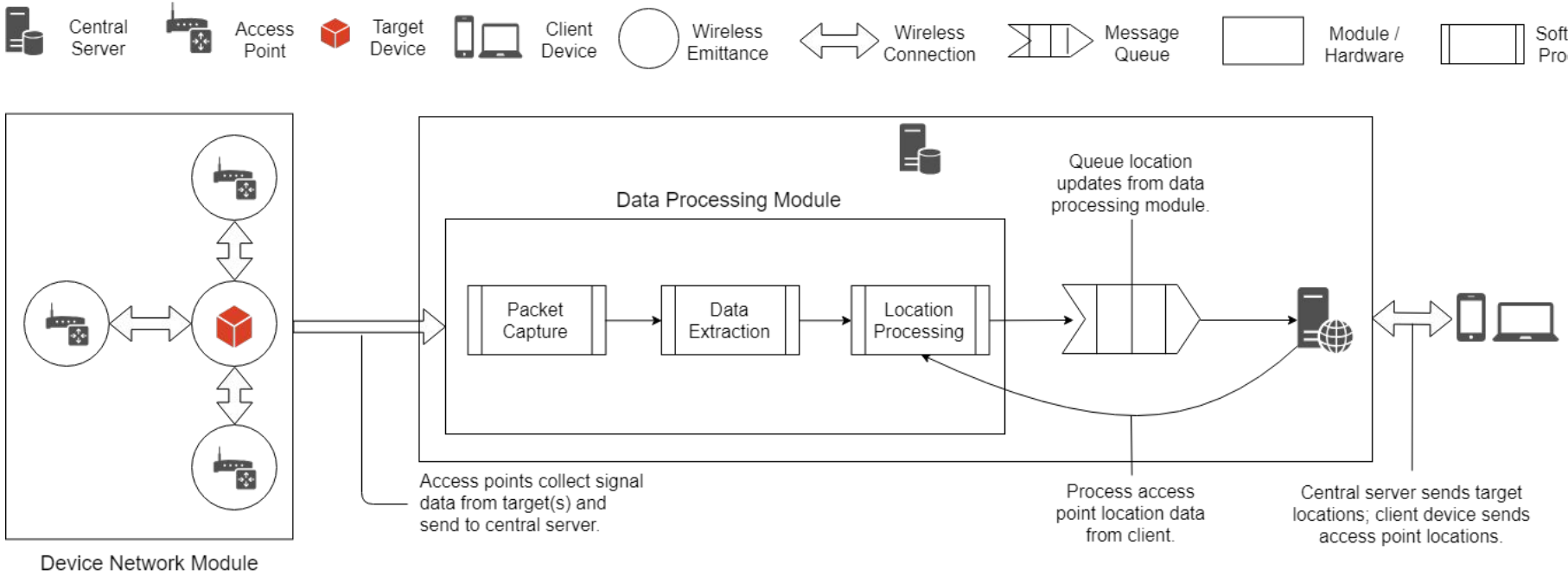
Project Design can be broken down into the following modules (depicted in the diagrams on the next few slides):

- Network Module: contains devices connected to the network including central server, access points, and target devices
- Central Server: contains processes and submodules necessary for packet capture and location calculations, as well as a web server for administration
- Web Server: contains the web application which accesses the device information database and displays device locations, configurations, and notification options for the administrator
- Client Module: contains the admin’s device which houses the web application frontend used to view devices and configure the network and notifications

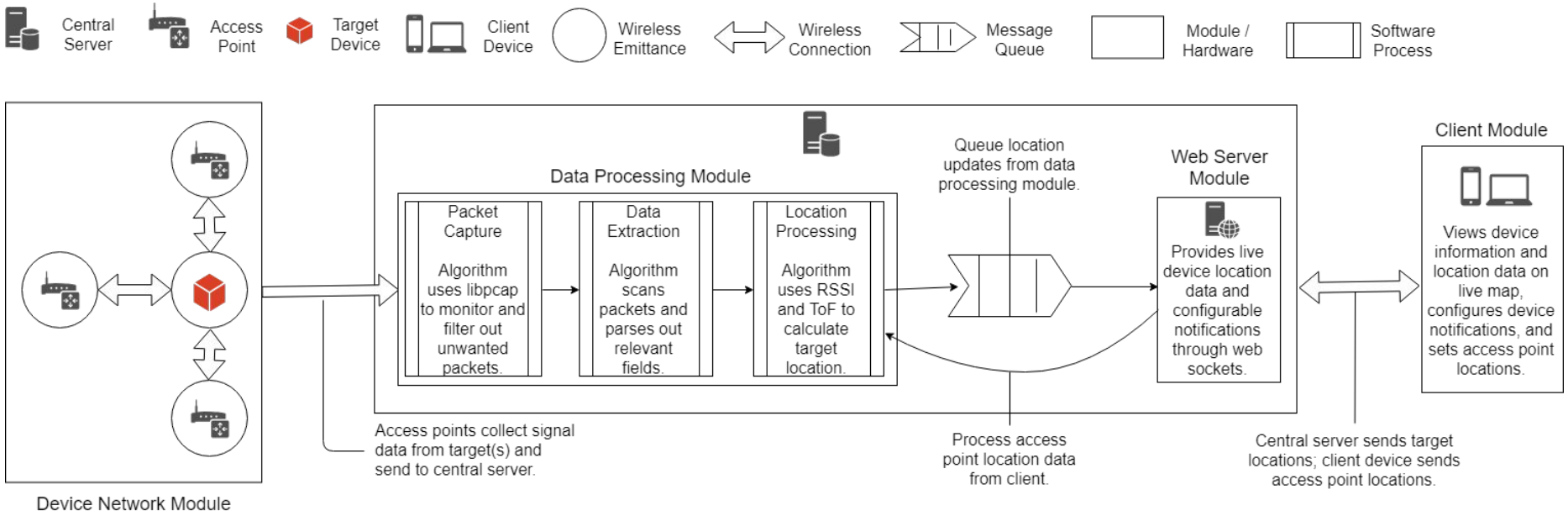
High Level Design



Mid Level Design



Low Level Design



Technologies

- **Python:** language used for data/location processing with the following libs
  - Scapy: used for network packet processing
  - Scipy: used for optimizing location estimates
- **OpenWRT:** framework running on network access points to capture packets from devices
- **NodeJS:** JavaScript runtime engine and web server used for hosting the web application
- **React JS:** JavaScript library used for building out the user interface of the web application
- **Heatmap JS:** JavaScript library used for creating the dynamic heatmap of device location concentrations

Milestones

1. **Set up hardware:** Determine hardware requirements for the project and implement basic set up of components from project design including access points, central server, and possibly a client device for testing purposes.
2. **Create modules:** Create modules defined in the project design. Define classes for each stage of the data processing module. Set up web server to communicate with message queue, client device, and data processing module. Set up web framework to be used for user interface on client device.
3. **Collect and extract data:** Write class to monitor, capture, and filter packets received from access points. Write algorithm to scan packets and parse out relevant data for location processing.
4. **Process location:** Research methods of efficient device location triangulation/trilateration. Write algorithm to use RSSI and ToF to calculate target device's location from extracted data.
5. **Set up communications:** Establish message queue and in-memory data management system to prepare and store location data on web server. Implement communication pathways between web server and client device.
6. **Implement user interface:** Design and develop the user interface for the client device, including features listed previously like device location and information listing. Create configurable notification system based on status of target devices or access points. Generate a method for setting up facility map of access points for end user.
7. **Test and document project:** Perform unit and integration tests of all modules in the project design. Document project according to course assignments. Conduct end user tests to gather feedback, document necessary instructions, and adjust implementation accordingly.

Deliverables Timeline

- **Deadline 1 - 01/16**
  - Obtain required hardware components for project, such as access points, central server, and client/target devices.
- **Deadline 2 - 01/30**
  - Set up central server and access points on OpenWRT framework.
  - Set up NodeJS server within central server.
  - Obtain test data from access points for web developers to work off of.
- **Deadline 3 - 02/13**
  - Write module to capture packets on access points and send them to central server.
  - Develop list view of nearby devices for showing general device information.
- **Deadline 4 - 02/27**
  - Write module to process packets and and retrieve relevant data on central server.
- **Deadline 5 - 03/13**
  - Write algorithm to locate devices from processed packet data.
  - Develop map view to show nearby devices.
- **Deadline 6 - 03/27**
  - Integrate heatmap.js library with map to show device location concentrations.
  - Integrate all modules together into cohesive process.
- **Deadline 7 - 04/10**
  - Write and perform unit and integration tests for design modules.
  - Wrap up documentation of the project.

Results

- Captured RSSI information for test devices on established network
- Calculated approximate distances using RSSI information
- Calculated approximate device locations using trilateration
- Deployed user interface with list view and map base
- Created heatmap overlay to show device concentrations

Challenges

- **Parallel workflow between UI design and lower level systems**
  - Obtained test packet capture as early as possible in the development cycle for our web developers to have some data to work with.
- **Efficient distance computation**
  - Used free space path loss algorithm with RSSI information to compute distances from the access points
- **Error accounting in trilateration calculation**
  - Minimized mean standard error by using Python library *scipy* functions in conjunction with an initial location estimate to achieve a more optimal location estimate
- **Snapshot decision of packet time interval**
  - Analyzed test packet capture across three access points and added configurable time interval for packets based off use case