EECE 451 – Mobile Networks and Applications

Spring 2015-2016

Application Project Requirements

Objective: Design and Implement a Mobile Application (Android or IOS) with communications capabilities.

Application Capabilities:

- Checks the connectivity state of various networking protocols on the device (Bluetooth, cellular, and WiFi) and displays their status.
- Detects devices with Wi-Fi P2P enabled and displays related info about them.
- The phone has to identify itself, using a MAC address or a phone number. The phone number can be obtained programmatically through the application itself, or interactively through an option menu provided to the user. More points will be awarded for the approach with automatic retrieval of the phone number.
- Lists in a Grid the set of previously detected devices along with associated information (latest time of detection within a configurable time frame). Allow for the ability of sorting the entries by one of the columns.
- When the phone detects a device, it keeps track of the length of detection (i.e., time range), and stores this information locally (deviceID, last time of detection, and last time range). It also keeps track of the number of times it detected this device, and the cumulative duration of detection (i.e., including current and all past detection events). Also, when a device is detected for the first time, give the user a change to tag it with a descriptive name, and so afterwards, refer to these devices through the user interface using that name.
- Have each device send any new information it gathers about detected other devices to a root server. The server will therefore contain records about each device and its neighborhood history.
- The server will therefore:
    - Collect "device neighborhood" information from the various mobile devices in the network (device name, date and time of last detection, frequency of detection, and cumulative time of detection). You will need to design the communication protocol between the server and the phone.
    - Runs an algorithm that creates a graph that connects all the devices in accordance with the information they sent. Each vertex (node) in the graph must have a label (name of mobile device, if available; else its id), and each edge (link) must have a pair of weights:

(frequency of connectivity, cumulative duration). The graph is generated upon a request from a mobile user (mobile device), and should be generated in accordance with the following parameters:

- Date/Time range: This has to do with generating a graph of the devices that connected to each other between March 1, 2016; 8:00AN and March 10, 2016; 4:00pm.
- Concerned devices: Specifies one device, which will create a star graph for all devices that connected to the specified device; or all, in which case the graph will include all the devices in the server.
- Extra: this is up to you to define (if you can think of one or more additional parameters). An example is a geographic location. You will gain additional points for such an option.

  - Allow the server to cache previously created graphs, so that when a graph is requested by a given user, the server can serve the cached graph if no new neighborhood information have been sent by phones.

- Moreover, enable users when they register with the server to receive graph update notifications. If a user has turn on the notification option, whenever any other user has asked the server to generate a graph, the server will send a notification through an SMS message natively (from within this application, not through any other default SMS app) to all phones that turned on the notification option.
- Once you have completed the development, move around with your phone so that it collects and sends to the server a non-trivial number of connection events. You will get additional points if you show through your results how connections increase within given times (e.g., breaks between classes) and in particular locations (e.g., in front of College Hall).

Evaluation and Grading:

Given the level of proficiency of Android programming among the group members, a group may fully implement some of the above requested capabilities, all of them, or add more capabilities that serve the same theme ("Device neighborhood Information Management"). A "perfect" grade is however based on accomplishing all of the above objectives.

Due Dates:

- A progress report is due on April 14 (to be uploaded to Moodle) - a demo may be requested to show progress.
- The project code is due on May 8 (to be uploaded to Moodle), and a **demonstration** is required on May 9 (all project members are required to participate, explain the details of their contributions, and answer questions about the code).