BusPlan

ECE 445 Design Document

Scott Liu, Connor Lake, and Aashish Kapur
TA: Evan Widloski

Objective

Buses are a form of standard mass transit service and are usually dispatched at fixed frequencies [9]. For example, MTD deploys their buses roughly once every 10 minutes during school hours and once every 30 minutes otherwise [6]. However, during rush hours we have observed buses to be completely packed; while on other occasions we have also found buses to carry only very few passengers. This poor load-balancing results in inefficiencies at multiple stages: losing existing customers due to overwhelmingly full buses, deterring potential customers due to crowded first impressions, operating at extremely light loads, etc. Herbert argued in his article within the context of urban bus transportation: "Fewer riders lead to less frequent service leads to fewer riders" [13]. Guihaire and Hao also speculate the car drivers might consider switching to public transport if we have a less crowded system [9]. It is therefore imperative that we develop a solution to address this commonplace issue.

Introducing the BusPlan: a network of smart detectors that actively survey the amount of people waiting for a bus, gather information, and send this information to a central unit that performs aggregation and analysis. Ultimately, the BLB will be able to find patterns, predict trends, and provide relevant information back to the bus company, who will use this to make strategic business decisions.

Background

Four stages of public transport operations have been identified by Ceder in his book Public Transport Timetabling and Vehicle Scheduling [3]:

1. Network route design
2. Setting timetables
3. Scheduling vehicles to trips
4. Assignment of bus crew

These steps together form a well established mode of public transport operation planning, and has already been adopted successfully worldwide. However, our team believes that the heart of the aforementioned problem lies within steps 2 and 3. The innovative network of devices that we propose would disrupt the traditional workflow, and greatly enhance the system's productivity and efficiency. By automatically detecting the amount of passengers waiting at a bug stop frequently and accurately, we would be able to obtain a sequence of numbers taken at successive equally spaced points in time. Furthermore, by analyzing and manipulating this time series data using modern tools, we would be able to achieve powerful forecasts that would replace timetables and scheduling completely. We believe that this would provide significant value to the bus company.

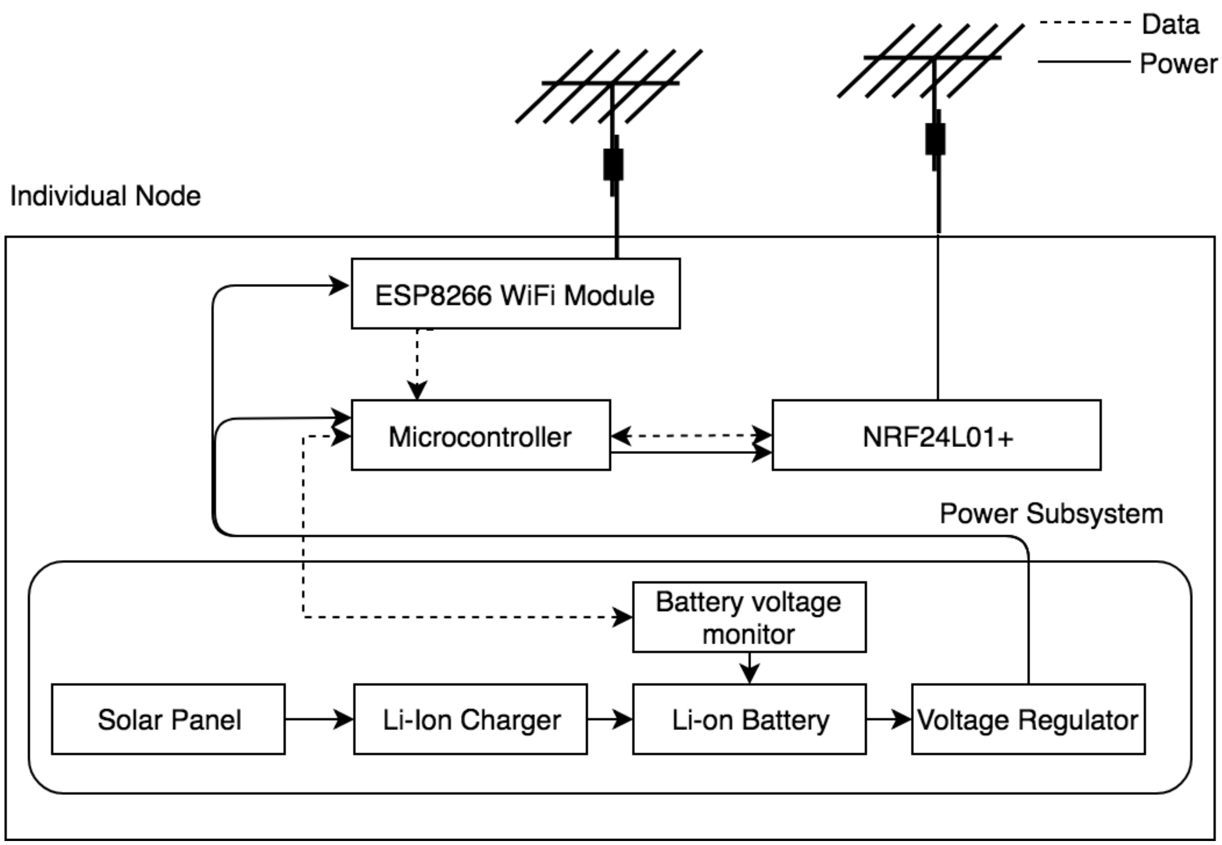
Many theories have already been proposed by experts to find an optimal way to form timetables or schedules[1] [10] [4] [8]. Some propose laying out optimal networks [5] while others focus on forming timetables after the fact [11] with real world case studies. They all offer incredible insight into formulating ideas to tackle this formidable challenge. However, most of them fall short when considering a truly dynamic environment that is subject to frequent change. Even theories that incorporate computer programs [16] [12] have been developed without this particular consideration. This is purely because they can only ever study past, outdated information as opposed to a stream of present, ever-changing information.

The buses that MTD use to provide service are already digitally augmented with GPS based technology [15]. It would be completely reasonable to add additional features to further enhance its efficiency. We also recognize the importance of a minimum bus service frequency [2] and will cater to delivering a model which takes this factor into account.

High-Level Requirements

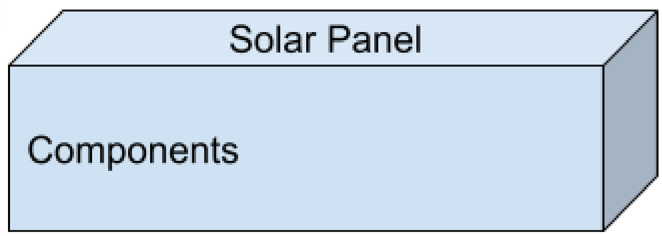
- Nodes must be able to collect an accurate measurement of number of people at bus stops by listening to WiFi probe signals
- Nodes must be able to relay data to a centralized server through other nodes
- We must be able to make predictions with reasonable accuracy and issue bus deployment recommendations that effectively alleviate the problem statement

Block Diagram



Each detector has three main parts: a power supply, a control unit, and a transmission unit. The power supply ensures that the system is powered continuously. The WiFi module in the control unit listens for probe beacons from nearby wifi stations (phones, laptops, etc). The radio chip in the transmission unit connects and relays data from one node to surrounding nodes. The control unit reads the data from the wifi chip and intelligently caches and sends it using the radio chip.

Physical Design



The physical design should be a flat and thin box with a large surface area on top for the solar panel, preferably around the size of a macbook trackpad(4” x 5.5”), with a height approx. equal to a laptop as well. Smaller is better for our device, so we will try and fit the components into as small of a case as possible.

1. Power Unit
    a. Solar Panel
        i. Each detector will be powered using a solar panel, which will charge the battery and power the device. Bus service runs 7am to 2am the following day, however the load at the edge hours will be much less, so we may have the scanning frequency lower than normal, saving power. Worst case, the solar panel must provide enough power for 19 hours of use.

Requirements	Verification
Outputs 300mA-1A between 4.40V-7.50V in no more than full sunlight (110,000 lux [10])	Place a solar panel in 110,000 lux. Confirm that the sunlight has no more than this intensity using a 1% tolerance photoresistor

	Measure the open-circuit voltage with a voltmeter, ensuring that it is below 7.50V Terminate the solar panel with a resistive load such that the voltage drop is 4.40V Ensure that the current through the load is above 300mA using an ammeter in series

b. Li-ion Charger

i. The charger takes power from the solar panel and converts it to the correct voltage/amperage for the battery. It makes sure to use constant current or constant voltages stages to charge the battery without charging it too fast.

Requirements	Verification
Li-ion battery charges to 4.16-4.23V when a continuous 4.4-7.0V input voltage is applied Charging at maximum current and voltage can be sustained below 125°C	Discharge a li-ion battery to 3.7 V cell voltage Charge the battery at the output of the AAT3693 from an input of 7V without limiting current At the termination of the charge cycle, signified when the charge status pin of the AAT3693 goes high, we will ensure the battery is charged between 4.16-4.23V Throughout the charging cycle outlined in verification 1.B-C, observe the temperature. Use an IR thermometer to ensure that the IC does not reach temperatures greater than 125°C

c. Li-ion Battery

i. The lithium-ion battery will store charge in order for the node to operate, taking in charge from the solar panel and delivering a constant voltage from the power subsystem.

Requirements	Verification
Must Store > 4200mAH of charge	Connect a fully-charged Li-ion battery with a test circuit discharging the battery at 300mA per hour, and make sure that the battery voltage is above 3.7V for the entire time

d. Voltage Regulator

i. This integrated circuit supplies the required 3.3V to the node system. This chip must be able to handle the peak input from the battery (4.2V) at the peak current draw (~300mA).

2. Control/Receiver Unit

a. ESP8266 Chip
    i. The ESP8266 is the brain of our node, as it contains a processor and a wifi chip. The wifi chip will listen to probe requests to track the number of people at a particular bus stop, and the processor will run the code that logs the mac addresses, and eventually send it using the transmission unit.

3. Transmission Unit
    a. NRF24L01
        i. This NRF24L01 is the primary means of communication for our node, as it's used for communicating between the node and the data collection node on the bus.
4. Central Server
    a. API / Database / Web interface
        i. The central server will run on a docker stack on a linux server. It will be exposed via an API that will listen for new data from the "drive-by" node. It will store this information in a database and make it available in a web interface and also through the ELK stack.
    b. Security protocol
        i. In order to securely transmit the data form the nodes to the server, we plan on using asymmetric cryptography. Each node will have the public key of the central server and encrypt all the data it sends to the drive by node with this public key. Only the server then can decrypt the data once it arrives.
    c. Data Analysis and Prediction
        i. The data will be collected in a way such that each entry corresponds to a timestamp. This way we will have a time series where we can train a LSTM model. We will use this model to make predictions.
    d. Scheduling algorithm
        i. To schedule the buses, we will first predict the amount of people required at each bus station, and then followed by finding out how many bus should be deployed in advanced to ensure that every bus has the optimal load.

Cost and Schedule

- Labor
    - Rate: $50/hour
    - Total estimated hours
        - Connor: 80/mon
            - 80 x 2.5 = 200
        - Scott: 80

            - 80 x 2.5 = 200
        - Aashish: 80
            - 80 x 2.5 = 200
        - Total: 240
            - 240 x 2.5 = 600
    - $50 * 600 = $30,000
- Parts
    - ESP8266 from ebay
        - Description: Wifi chip for listening to probe beacons
        - Part #: “Does not apply”
        - Quantity: 5
        - Cost / item: $3.11
        - Total: $15.55
    - ESP8266 from amazon
        - Description: Wifi chip for listening to probe beacons
        - Part #: “Does not apply”
        - Quantity: 2
        - Cost / item: $6.5
        - Total: $13
    - NRF24L01+PA+LNA
        - Description: Radio chip for transmitting captured data to drive-by node
        - Part #: “Does not apply”
        - Quantity: 5
        - Cost / item: $2.53
        - Total: $12.65
    - ATMEGA328P
        - Description: Microcontroller for controlled logic between NRF radio chip and ESP8266 wifi chip, only used if necessary
        - Part #: “Does not apply”
        - Quantity: 2
        - Cost / item: $1.88
        - Total: $3.76
    - 4200mAh Battery
        - Description: 4200mAh standard Li-ion rechargeable battery
        - Part #: Panasonic 565068
        - Quantity: 2
        - Cost / item: $10.88
        - Total: $21.76
    - Voltage Regulator
    - Battery monitor
    - Solar panel
    - Battery charger - AAT3693
    - PCB

- Container

Schedule

Date (by end of week)	Scott	Connor	Aashish
9/24	Research for the project proposal	- Set up gitlab repos, add basic shell / template for development - Start writing some basic API code for accepting MAC addresses	
10/1	Generate dummy data to simulate the passenger count time series	- Finish basic API for accepting data from the "drive-by" node	Design and order the PCB
10/8	Start writing code for data analysis	- Finish basic API for devices to use	
10/15	Put the node together after the PCB arrives. Start testing and debugging.	- Basic frontend to display / manage data - ELK setup	Put the node together after the PCB arrives.
10/22	Work on fixes from the previous week	- Work on getting multiple devices to talk to each other - Get encrypted communication setup	
10/29	Arrange several nodes together to simulate potential scenarios and collect data		
11/5	Collect more data, field test		
11/12	Perform analysis on field test results, train machine learning models	- Goal to have full working setup with multiple devices gathering data and drive-by node fetching data and uploading to master server	
11/19	Prepare and present predictions	- cleanup, final touches, productizing	
11/26	Prepare final demo	Prepare final demo	Prepare final demo
12/3	Prepare final report	Prepare final report	Prepare final report

Discussion of Ethics and Safety

We are collecting lots of data, and data that is uniquely identifiable and easily tied to an individual. Because of this there is an ethical dilemma of how the data will be used or available to companies that will pay for the service. We plan to keep the data anonymous and detached from any PII (personally identifiable information). We plan to look at the data as a whole, not on an individual level. Our goal is to optimize bus routes and give bus companies insights into how their customers are using their service, not identify and track individual people or devices. We also will have each node encrypt all the data it sends using the public key of the server to make sure nobody has access to the data besides the central server.

Because our devices will be outdoors in potentially harsh conditions, we need to make sure the internals of the device are properly insulated from rain, excessive heat, and other harsh conditions that could cause the battery to explode or ignite. We plan to use a rather small lithium ion battery, but even so they can release a large amount of power in a short amount of time is the battery is ruptured. We plan to use a properly insulated container that will be sealed to prevent water or snow to enter.

Citations

Special thanks to our TA Evan for suggesting we use the esp8266. It is the cornerstone of our project and without knowledge of the chips existence we could have done the project in a much less optimal way.

[1] Simon J. Berrebi, Kari E. Watkins, and Jorge A. Laval. A real-time bus dispatch- ing policy to minimize passenger wait on a high frequency route. *Transportation Research Part B: Methodological*, 81:377 – 389, 2015. Optimization of Urban Transportation Service Networks.

[2] Larry A. Bowman and Mark A. Turnquist. Service frequency, schedule reliability and passenger wait times at transit stops. *Transportation Research Part A: General*, 15(6):465 – 471, 1981.

[3] Avishai Cede. *Public Transport Timetabling and Vehicle Scheduling*, chapter Chapter 2, pages 30–57.

[4] Avishai Ceder. Bus frequency determination using passenger count data. *Transportation Research Part A: General*, 18(5):439 – 453, 1984.

[5] Avishai Ceder and Nigel H.M. Wilson. Bus network design. *Transportation Research Part B: Methodological*, 1986.

[6] Champaign Urbana Mass Transit District. 1 yellow: Weekday - day. https://mtd.org/maps-and-schedules/ routes/1-yellow-weekday-day/. Accessed: 2018-09-13.

[7] Association for Computing Machinery. ACM Code of Ethics and Professional Conduct. https://www.acm.org/code-of-ethics. Accessed: 2018-09-14.

[8] B. Gavish, P. Schweitzer, and E. Shlifer. Assigning buses to schedules in a metropolitan area. *Computers & Operations Research*, 5(2):129 – 138, 1978.

[9] Valerie Guihaire and Jin-Kao Hao. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251 – 1273, 2008.

[10] Jan Owen Jansson. A simple bus line model for optimisation of service frequency and bus size. *Journal of Transport Economics and Policy*, 14(1):53–80, 1980.

[11] W. Lampkin and P. D. Saalmans. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *Journal of the Operational Research Society*, 18(4):375–397, 1967.

[12] Joao Mendes-Moreira, Luis Moreira-Matias, Joao Gama, and Jorge Freire de Sousa. Validating the coverage of bus schedules: A machine learning approach. *Information Sciences*, 2015.

[13] Herbert Mohring. Optimization and scale economies in urban bus transportation. *The American Economic Review*, 62(4):591–604, 1972.

[14] Institute of Electrical and Electronics Engineers. Ieee code of ethics. https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: 2018-09-14.

[15] University of Illinois Faculty and Services. University of illinois faculty and services: Transit. http://www.fs.illinois.edu/ services/more-services/tdm/transit. Accessed: 2018-09-09.

[16] N. Seshagiri, R. Narasimhan, S. L. Mehndi- ratta, and B. K. Chanda. Computer generated time-tables and bus schedules for a large bus transport network. *Transportation Science*, 3(1):69–85, 1969.