UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

IOT SYSTEMS FOR TRAVEL TIME ESTIMATION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

MOHAMED TAMER ALAMIRI
Norman, Oklahoma
2017

IOT SYSTEMS FOR TRAVEL TIME ESTIAMTION

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

____
Dr. Hazem Refai, Chair

____
Dr. Thordur Runolfsson

____
Dr. Kam Wai C. Chan

To my parents, Rawa and Saaid, who have always been loving, caring, and supporting me in every way.

To my dear brothers, Amjad and Samer.

I dedicate this work to you all!

Tamer Alamiri

Acknowledgements

I would like to extend my heartfelt appreciation and gratitude to my dear advisor, *Dr. Hazem H. Refai*, for the endless support and insightful guidance he has provided throughout my *Master Degree* at *The University of Oklahoma*.

I would also like to express my sincere gratitude to the distinguished members of my committee. *Dr. Thordur Runolfsson*, and *Dr. Kam Wai C. Chan.* Thank you all.

I must acknowledge all my fellow students who supported me with their friendships and unlimited assistance.

Special appreciation goes to the OU-TULSA family of professors, students, and staff, as well as to Michelle Farabough for her assistance in editing this dissertation.

Table of Contents

List of Tables

List of Figures

Abstract

Travel time (TT) is a major concern for commuters and those who implement traffic management systems. This indicator provides information to operators about road conditions and helps travelers plan trips, avoid congestion, and save fuel. A number of technologies are currently used in the transportation industry for estimating TT and detecting vehicles (e.g., radar, inductive loop detectors [ILD], Bluetooth [BT] scanners, cellular phone signals, surveillance cameras, and others). This research centers on ILD and BT scanners for providing a real-time TT estimation system.

This thesis proposes Internet of Things (IoT) system architecture for detecting vehicles and estimating real-time TT. Several techniques were investigated for re-identifying vehicles based on inductive loops by preprocessing signature signals and searching for correlations to estimate TT between two locations. The proposed ILD system will be accompanied by a BT scanner, serving as a complementary system to compare and validate TT estimation efficiency.

This research introduces new approach for vehicle re-identification by computing Relative Entropy and Pearson correlation between ILD signatures, and then estimating TT based on the highest correlated signatures. To clear measure noise, TT for vehicles is assumed to follow the same pattern within a certain time frame. Thus, TT values are arranged in time series groups before applying a spike detection algorithm to determine the TT range with the highest number of vehicles. A data spike is considered for estimating TT. Given that the number of vehicles within the spike is greater than number of vehicles in all other data groups, TT will be the mean value of TT within the spike.

Chapter 1: Introduction

The popularity of utilizing just-in-time manufacturing processes and complex supply chain networks highlights the value of travel time (TT) reliability to businesses, as well as commuters and individuals traveling for leisure. An increasing emphasis on customer-based performance metrics makes network and TT reliability essential measures for rating transportation agency performance and determining the effectiveness of mitigating roadway congestion and transportation delays. TT is also a time-based performance measure of transportation quality, aiding departments of transportation in planning highway expansions and scheduling maintenance activities. TT has proven reliable for traffic surveillance on a day-to-day basis for detecting traffic bottlenecks, incidents, work zones, special events, and other major traffic conditions.

Probe-based data has primarily been used by departments of transportation to measure TT on road segments and compare TT between commercial and passenger vehicles. Probe-based TT data is historical data collected on a daily basis in 5-minute intervals [6].

Ever evolving technologies and the rapid growth of the Internet of Things (IoT) systems during the past decade have made the precise measurement of real-time TT easier than ever before. Many technologies are now implemented to detect vehicles and estimate TT-per-vehicle in real time (e.g., Bluetooth [BT] scanners, surveillance cameras, toll tag readers, inductive loops detectors, cell phone signals, and GPS, among others) [7], [8].

IoT systems form a network for the physical world, which enables various embedded devices to connect to the internet and exchange data in real time. This technology senses and remotely controls different types of systems. IoT systems have been gaining more attention in the last few years in both academic and industrial domains. Currently, IoT applications range from health monitoring and smart grid systems to intelligent transportation systems. [9], [10], [11].

This research presents an overview of IoT system architecture for two devices that estimate real-time TT—one based on BT technology and the other based on inductive loop detectors (ILD). Both consist of on-ground units placed on various highway segments that have been identified throughout the state of Oklahoma. The units are connected to either BT scanners or ILD. The former can detect passing vehicles equipped with BT devices; the latter senses passing vehicles based on the magnetic field generated when a vehicle passes over an ILD. Both systems send their data to the cloud for estimating real-time TT.

An algorithm is proposed using the magnetic signatures of ILD to re-identify vehicles crossing a highway segment in Oklahoma City. The algorithm preprocesses the data to clean the noise; estimates vehicle speed and length; applies speed-based normalization for vehicle signatures; and then calculates TT between origin-destination pair by correlating signatures using Pearson correlation and relative entropy. A spike detection algorithm is applied to cancel out noise in detected TT values.

The main goal of this project is building a sustainable IoT system for estimating real-time TT by evaluating BT technology and ILD performance. Evaluation is based on the following factors.

1. Data penetration
2. Distance-based TT estimation accuracy
3. Individual vehicle TT estimation accuracy

4. Alert detection (e.g., short term for accidents and long term for road work)
5. Speed estimation
6. TT reliability (i.e., probability of a traveler to complete a given origin-destination trip within a prescribed time).

This thesis is organized as follows. Chapter 2 provides necessary background information for describing the motivation for the research presented in this thesis, including

- TT and traffic metrics, describing data metrics required by ODOT for estimating TT, monitoring roadways, determining the efficiency of road conditions, and planning maintenance procedures,
- Methods for TT estimation, describing systems currently used to estimate TT, as well as technologies available for vehicle detection and TT estimation, and related research work, describing related work for estimating TT using BT and ILD.

Chapter 3 describes the IoT system used to collect the data, including

- IoT system overview with a brief history of IoT systems,
- IoT system design concepts, describing different types of IoT systems, as well as design concepts that should be included, and
- REECE system architecture, describing the REECE IoT device and briefly discusses hardware system and software design.

Chapter 4 describes methods for estimating TT using BT scanners, including

- Background of BT technology,
- Ubertooth hardware used for sniffing vehicles equipped with a BT device, and

- Vehicles detection based on BT devices and BT TT analysis, discussing estimated TT values for reliability and alert detection.

Chapter 5 describes ILD systems and data, including

- ILD overview, describing hardware components and measurement methods,
- ILD signature physical description, describing signatures generated by ILD and the features that identify a certain vehicle, the quality of an ILD signature, and signature uniqueness,
- Vehicle identification utilizing ILD Signatures, describing methods used for preprocessing ILD signatures, as well as correlation methods for re-identifying vehicles based on their signatures, and
- ILD TT analysis, discussing estimated TT values relative to reliability and alerts detection.

Chapter 6 provides a comparison for TT values between ILD and BT, including

- TT reliability, comparing BT and ILD, and
- Real-time TT estimation, comparing the quality of data reported by the BT and ILD systems, as well as the frequency of reporting accurate TT estimations.

Chapter 2 – Background

TRAVEL TIME AND TRAFFIC METRICS

Travel time (TT) is one of the best indicators of traffic system performance. An accurate real-time traffic surveillance system is crucial for determining metrics for intelligent transportation systems (ITS) [15], [16]. TT information aids individual travelers with travel route decisions, improves roadway efficiency, increases transportation systems safety, and maximizes existing transportation infrastructure capacity. Retailing and shipping businesses are better prepared to manage their delivery systems and optimize shipping costs.

The U.S. Federal Highway Administration (FHWA) predicts a 23% increase in vehicle miles traveled by 2032 [12]. According to the World Health Organization (WHO), 1.25 million people die and up to 50 million injuries occur each year on the world's roads [13]. Furthermore, 44% of the U.S. roadways are classified as congested. According to the 2015 Urban Mobility Scorecard report [14], traffic congestion costs the U.S. $160 billion each year. Therefore, it is of great importance to monitor traffic in a robust and accurate manner.

A reliable traffic monitoring system should detect traffic conditions not only in free flow but also when bottlenecks result from increased merging/diverging demand at on/off-ramps and lane drops or when traffic becomes congestion due to exceeding demand for limited roadways. A more accurate TT estimation system should detect highway performance per lane, providing short-term alerts due to accidents or storms and long-term alerts for work zones and infrastructure maintenance. [17], [18].

Following is a list of traffic monitoring technologies currently used by transportations agencies to estimate TT.

1. Automatic license plate readers
2. Global positioning system
3. Cell phone signal monitoring
4. Crowdsourcing
5. Vision-based vehicle re-identification systems
6. BT identification detectors
7. In-pavement magnetic detectors

METHODS FOR TRAVEL TIME ESTIMATION:

Automatic License Plate readers for travel time estimation:

Automatic License Plate Matching (ALPM) techniques are based on collecting license plate numbers with a time stamp at a collection points (i.e., check points), and then matching the numbers between consecutive check points by computing the difference of arrival to estimate TT [20].

Several methods have been introduced for this purpose. The most commonly used method utilizes cameras to capture vehicle images, identify the license plates, and automatically process the license plate text/numbers. Cameras are typically equipped with infrared lighting to facilitate image capture during daytime and nighttime [21], [22]. Cameras are often installed on automatic toll collection points and inside law enforcement vehicles.

ALPM can determine TT based on a large sample of vehicles. This method is useful for understanding variability of TT among vehicles within the traffic stream.

Another benefit of this technique is equipment portability. It should be noted that ALPM provides a continuum of TT data only during short collection periods (e.g., 15-minute average of continuous data), and TT data is limited to locations where video cameras can be positioned or other means of visual data capture is possible.

Travel Time based on Global Positioning System:

TT based on a Global Positioning System (GPS) utilizes probe vehicles for monitoring traffic and estimating TT (i.e., probe vehicles report locations, speed, and time stamps to the main system). TT between two locations is estimated using a dynamic Bayesian network after properly scaling the routes that vehicles travelled, and then matching the map discretization [23], [24]. Systems that report these types of data measure traffic performance on a monthly and yearly basis and categorize the information under a historical data reporting system.

The major disadvantage of a GPS-based system is the low penetration rate of traffic flow (i.e., a vehicle sends one sample every minute, and number of samples at each road segment is three to eight samples every 5 minutes). Also, vehicles do not provide detailed information about roadway delays.

Cell Phone Signal Monitoring:

Cell phone signal monitoring takes advantage of the fact that mobile phones must conform to the standards of the global system for mobile communications (GSM), which has the ability to measure the signal strength from the associated cell tower and the six strongest neighboring cell towers [26] (i.e., received signal strength [RSS]). Recently, cellular networks have reached 100% coverage in all major cities with slightly lower coverage on highways that connect cities. Although high cellular technology usage and the ability to measure cell phone fingerprints using cell phone signals creates a rich environment for estimating TT by leveraging GSM and 4G networks, disturbances in radio propagation caused by movement and changes in atmospheric conditions cause fluctuations in received signal strength. The result is erroneously associating cell phones with cell towers over time.

Determining TT using cell phone signals can be estimated by determining the location of the vehicles on road segments based on RSS fingerprints, and then estimating the direction and speed of a vehicle based on the changes of the signal power levels [25].

Crowdsourcing:

Widespread accessibility to the internet and the influence of social media networks are increasing the importance of connecting virtual communities for knowledge sharing. Two types of internet platforms can be leveraged for traffic surveillance. The first type relies on gathering geo-tagged pictures, video, and posting information that people share on social media networks about traffic related news, like accidents and congestions [27], [28]. The second uses dedicated navigation web-based services (e.g., Google maps, Bing maps) in which commuters agree to share their location coordinates when using the service [29]. Although the former reports information and images, its use for real-time traffic monitoring is unsafe, as it encourages motorists to use their cell phone while driving. Doing so will distract a driver from paying attention to the road. The latter is becoming more reliable for estimating TT and planning travel routes, as an increasing number of people are using such web-based service. This method does not require drivers to proactively share information.

Vision-based vehicle re-identification systems:

Vision-based, re-identification systems consist of multiple video cameras strategically placed to record traffic flow at consecutive locations. Video images of vehicles are captured by applying several object recognition algorithms so that recognizable features (e.g., color, vehicle dimensions using 3D and 2D boxing, light shape, manufacturer trade mark) can be extracted. This method has proven highly successful for estimating TT [30], [31].

Notably, however, many vision-based, TT estimation studies use simulated data because current technologies for identifying vehicles are not compatible with systems that require low-power consumption, low data traffic, and low cost.

Related Research Work:

Bluetooth Identification Detectors:

A BT device can be connected via spectrum scanning mechanism to determine its Media Access Control addresses (MAC address), which is a unique 48-bit identifier that is available in the coverage area [32]. Next, a hand shaking step with a selected device can be deployed to authenticate the connection and commence data transfer. Similarly, Bluetooth Traffic Monitoring Systems (BTMS) scan the spectrum at specific checkpoints to capture all available BT devices and temporarily store their MAC addresses with a detection time stamp. When the same MAC address appears at another checkpoint, BTMS measures TT between the two checkpoints by calculating the difference of detection times [35].

BTMS is unable to estimate TT, however, unless the MAC address of a BT device is detected at both an origin and a destination point. Even so, BTMS is a valuable tool for Intelligent Transportation Systems (ITS) due to its simplicity in estimating TT and low cost for deploying BT scanning sites.

Applying BTMS in the traffic domain started some 10 years ago with the goal of enhancing ITS applications (i.e., using TT to perform traffic light management, suggesting alternative routes to avoid work zones). The city of Houston, TX is well known for its comprehensive BTMS operation [4], [39].

The WHO reports [34] that using cell phones while driving is a major cause of accidents, primarily because their usage distracts drivers. As a result, many countries now restrict the use of cell phones and ban drivers from using hand-held mobile smartphones. Consequently, the 12 leading, global vehicle manufacturers have agreed to install BT devices for supporting hands-free smart phone communication [33]. The number of vehicles using BT devices continues to rise significantly, which improves the accuracy of BTMS as BT device penetration in the traffic flow increases.

In spite of the rapid growth in applying BTMS in ITS, BT TT estimation suffers measurement spatial error due to the fact that BTMS calculates zone-to-zone instead of point-to-point TT. Spatial error occurs when detector coverage area is determined by antenna type, transmission power, and environmental conditions, which affect vehicle distance, TT, and velocity measurements [36], [37], [38].

Recommended suggestions for improvement include using a directional antenna, shortening the coverage area of the BT detectors, or using a received signal strength indicator (RSSI) for locating the vehicle within the coverage zone. Directional antennas improve accuracy because BT detectors are limited from detecting BT devices traveling on unmonitored roads. However, using RSSI as a measurement metric is not a viable approach because each BT chip class transmits a variable power. Also, shortening a detector's coverage area reduces the number of detected vehicles. Increasing a detector's transmission power increases the number of detected vehicles, such that spatial error can be compromised when averaging the values of all detected vehicles.

In-pavement magnetic detectors:

Magnetic detectors, especially ILD, were first introduce as part of traffic monitoring back in the 1960s, and they have since become the most widely used sensor in traffic management systems. ILD creates a magnetic signature of a detected vehicle when passing over the sensor [41]. ILD has aperture, which means that information about the vehicle geometry is "masked" in its magnetic signature due to the averaging properties of ILD. Quantitatively, this process is characterized by spatial transfer function (i.e., spatial impulse response) of the sensor. Information recovery from the signal is conditioned by the knowledge of this transfer function [40].

ILD applications are beneficial when high-resolution data is required. For example, the method has been applied for point-based measures (e.g., vehicle counts, classification) [46]; section-based measures (e.g., emission estimation, TT estimation, speed), and origin destination-based measures (e.g., long distance TT estimation). Authors in [45] proved vehicles could be classified according to generated ILD signatures. Their work developed a novel approach for classifying vehicles into five classes by applying discrete Fourier transformation to clear the noise. Principle component analysis was used for decorrelation and dimensionality reduction. Finally, parameters were classified into a supervised, three-layer backpropagation neural network. Accuracy reached 64% without the use of a fault threshold and 85% when the threshold was applied to the correlation metric.

Common practices for estimating TT using inductive loop detectors include measuring volume and speed at detection locations and assuming identical traffic conditions prevail along the entire segment between detection points. Many studies have proven that these estimates are not accurate and may contain significant errors [5], [43], [44] [47].

Few studies have investigated origin-destination TT estimation based on vehicle re-identification using inductive loops signatures. A well-known study by researchers at the University of California at Irvine developed a real-time, inductive loop signature-based vehicle re-identification algorithm (RTREID-2) that utilized piece-wise slope rate to compare similarities between signatures [3]. The algorithm was tested on a dataset collected from three checkpoints located 0.33mi and 1.3 mi apart during peak morning hours travel between 6:30 and 10 a.m. The system was equipped with video recording, and data was divided into nine discrete time periods, each 5-minutes in length at 6:35, 7, 9, and 9:35 a.m. for uncongested road status and 7:35, 8, 8:10, 8:20, and 8:35 a.m. for congested road status. A number of preprocessing steps (e.g., data cleaning of corrupted signatures, vertical normalization, noise elimination by dropping the lower 20% values of the magnitude, data interpolation using cubic spline method ensuring each pair of signatures was equal in length) were applied to the signatures prior to comparing piece-wise slope rates. The final step in the algorithm subtracted differences in the piece-wise slope rates and selected correlated signatures based on the minimum of the total differences between signatures. Only 43.8% of vehicles were re-identified using this algorithm with 75% of TT accuracy in the free flow state and 52% of TT accuracy in congested state.

Another study [48] used cross correlation as a factor to re-identify vehicle signatures. Data was limited to signatures captured on lead and lag loops at each checkpoint and validated with a video recording as a ground truth. Low pass filtering was applied to reduce signature noise, and speed-based normalization was used to preprocess the signatures. Finally, cross correlation was implemented to compare lead signature on checkpoint 1 with lead signature on checkpoint 2; lag signature on checkpoint 1 with lead signature on checkpoint 2; lead signature on checkpoint 1 with lag signature on checkpoint 2; and lag signature on checkpoint 1 with lag signature on checkpoint 2. The goal was determining highly correlated signatures and estimating TT based on detection time differences. TT results were clustered into two groups, separating accurate TT measures from erroneous measures. Under controlled traffic for 1 hour and 15 minutes and 0.7 miles distance, this algorithm provided 40.57% TT accuracy.

Chapter 3 – IoT Systems

IoT Systems overview

The term Internet of Things (IoT) was first introduced by Kevin Ashton in 1998 [49]. However, the idea of connecting things to the internet was first conceived in the 1970s. In the last few years, however, IoT has grown rapidly and gained more attention with improvements like embedded hardware capabilities and the expansion of internet coverage and bandwidth. The number of things connected to the internet has now reached six billion with expected growth to 20 billion by 2020 [19].

IoT has been implemented in multiple contexts (e.g., smart houses and offices, e-health, intelligent transportation systems, autonomous automobiles). The impact of IoT systems on everyday life, along with anticipated advantages of providing users with insights about their daily routines to aid them in making important decisions, remains to be seen. IoT system data currently aids businesses by informing about rapid changes in the marketplace and industries through enabling automation for faster product prototyping and manufacturing [50].

Many challenges to IoT systems related to technology and societal aspects remain. Lack of standard system architecture affects compatibility between IoT systems. Often, deploying additional hardware or software is required to connect various devices. Remote accessibility to IoT systems makes security a serious concern (e.g., hacking health IoT systems makes individuals vulnerable to IoT hack attacks). Also, the increased amount of data recorded about individuals' personal lives raises concerns about user data privacy.

IoT Design Concepts

Having a reliable architecture for easy connectivity, control, and communication between things, the cloud, and other IoT systems is paramount [51]. Before implementing an IoT system, many fundamental needs should be considered about hardware, cloud connectivity, security, and embedded operating systems (OS).

Hardware Devices:

Hardware devices should be selected according to necessary processing power for sensing data. The system should manage severe environment in intended areas for deployment. Power consumption should be estimated in the prototyping phase, as different methods for powering the devices affect operating time. Designing an IoT system powered directly from the grid is quite different from designing an IoT system powered from a rechargeable battery connected to a solar panel. In the latter, the available amount of energy related to the weather conditions is required. A single use, regular battery has a number of constraints, which increases device operational time. Internal storage memory should be ample for running the OS, storing data, and maintaining system log records. A final consideration for selecting hardware components is to consider the intended method for connectivity with other hardware devices. Many applications require combining multiple embedded devices from various manufacturers over USB, ADC, GPIO, and/or RS232. Design considerations include minimizing the number of wires between devices. This is especially important, since wires are prone to errors due to interference in any embedded application.

Wireless Connectivity and Internet:

Given multiple wireless IoT devices in a single location, best practice calls for utilizing a local area network between them using BT or Wi-Fi. Internet access should be available for each device, saving data traffic usage and reducing the number of registered public IPs. Notably, in some scenarios, this design consideration is disadvantaged by slowing down the connection between devices and the cloud.

Real Time Operating Systems (RTOS):

An IoT device OS should be designed to operate for a reasonably long time without reboot. Embedded software should be optimized, and the required number of system services should not exceed device CPU and RAM capabilities. The OS should have a watchdog time to detect instances when the device stops working and requires rebooting. A second watchdog protocol should be designed specifically for detecting internet access and rebooting the connected internet broadband modem in the event that internet connection is lost. Another important feature is automatically updating software without human intervention (over-the-air updates).

Security:

Security is a major concern for IoT. As such, it should be considered at each design phase—not as an afterthought. Critical security issues include data ethics, privacy, and liability [52]. Solutions for improving security include:

a. Encrypting the Linux box storage memory,
b. Encrypting messages transmitted between the cloud and IoT embedded devices, and
c. Securing the update process.

REECE IoT System

REECE (Roadside Embedded Extensible Computing Equipment) is a Linux box designed to store and process sensor data connected via different types of interfaces (e.g., ADC, USB, RS232). REECE controls communication between sensors and the cloud using a websocket connection to facilitate real-time sensing and environment monitoring.

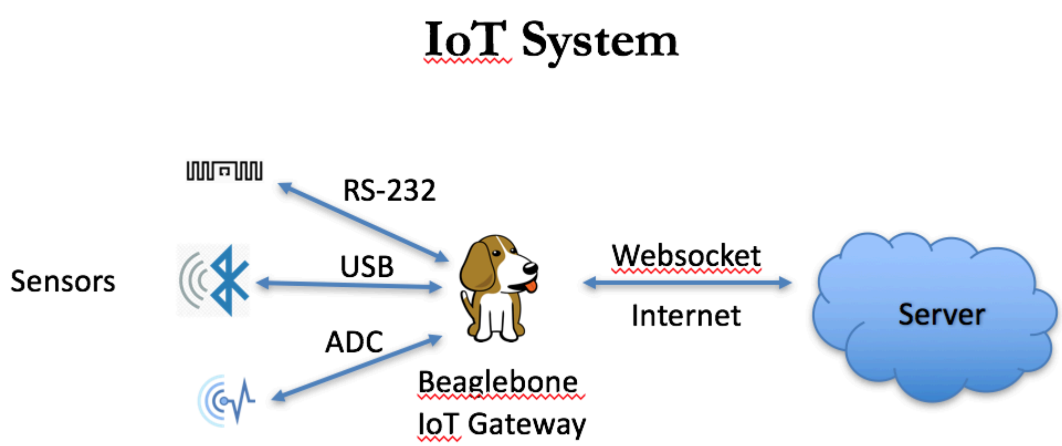


Figure 1. IoT system architecture.

REECE hardware architecture, depicted in Figure 1, is composed of:

1. Arm cortex Linux-based development board (e.g., BeagleBone Black),
2. Extension cape from Innovative Traffic Systems & Solutions (ITSS), LLC, and
3. Internet broadband modem,

BeagleBone Black (BBB) is a low-cost development platform with 1 GHz ARM Cortex A8 preprocessing power, 512MB DDR3 RAM, 4GB 8-bit eMMC onboard flash storage, and 2x PRU 32-bit microcontrollers. BBB has USB client for power and communication, USB host, Ethernet, HDMI, and 2 x 46pin headers [53]. Figure 2 details BBB on board components [54].

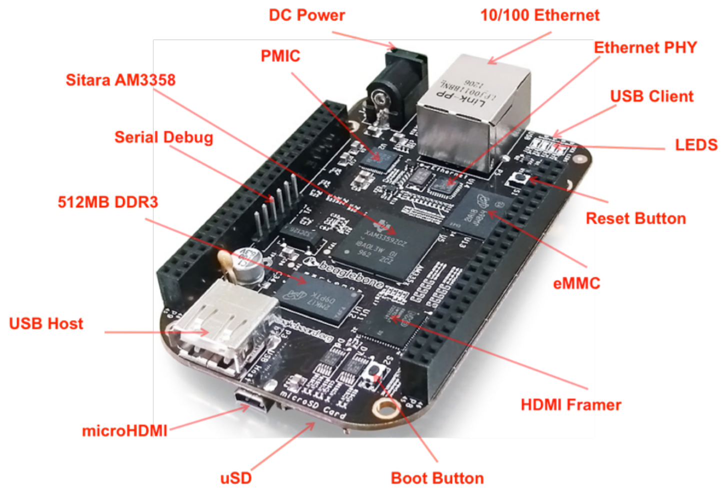


Figure 2. BeagleBone Black components.

For the *extension cape,* a PCB board was developed ITSS, LLC (See Figure 3; http://itrafficsystems.com) extend the number of peripherals for BBB from one to four USB ports; b) expanded the UART pins to two RS232 ports; and c) provide easy access to the onboard analog digital channels.

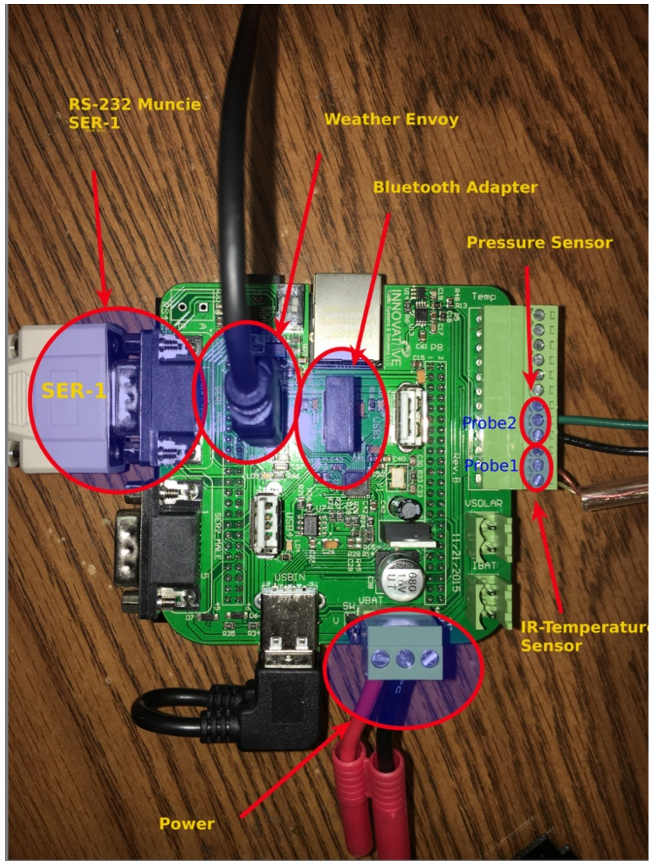


Figure 3. BBB extension cape by ITSS, LLC

The *internet broadband modem* is the most important component of any IoT system, as it provides connectivity to the internet. Various types of broadband modems were tested. Table [1] provides a list of tested broadband modems and their performance.

Table 1. Deployed Modems.

Modem	Operator	Performance
Netgear 341U	Sprint	Modest Performance; loses connectivity overnight and requires reboot every 12 hours.
Netgear 340U	AT&T	Good Performance; remains connected 24-hours-a-day
Franklin U772	Sprint	Poor Performance; crashes after 2 hours of connectivity
Sierra Wireless AC250	Sprint	Modest Performance; loses connectivity overnight but it requires to be reboot every 12 hours.
ZTE Velocity Hotspot	AT&T	Great Performance; remains connected more than 30 days without disturbance
Sierra Wireless LS3000	Verizon	Great Performance; remains connected more than 30 days without disturbance

REECE software architecture is powered by a customized Linux OS for running applications, managing interfaces, and connecting to the internet. The proposed system's Linux OS follows the general Linux hierarchy structure (See Figure 4) and has four main components:

1. Linux kernel,
2. System management tool,
3. Database management tool, and
4. Javascript engine for real-time connectivity with the cloud.

The *Linux kernel* is the core software of any Linux OS. It controls hardware resources (e.g., CPU, main memory, data storage devices) via multiple subsystems (e.g., process scheduling subsystem, memory management subsystem, virtual files subsystem, network subsystem). A number of kernel distributions are provided by open source communities; others are provided by manufacturers (e.g., Texas Instruments) to support proprietary hardware systems. Two Linux kernels, namely 3.8 and 4.1, were tested and proved stable over long-term deployments. Kernels between 3.9 and 3.18 had significant problems that caused the IoT system to reboot or stop working only 10 hours after deployment.

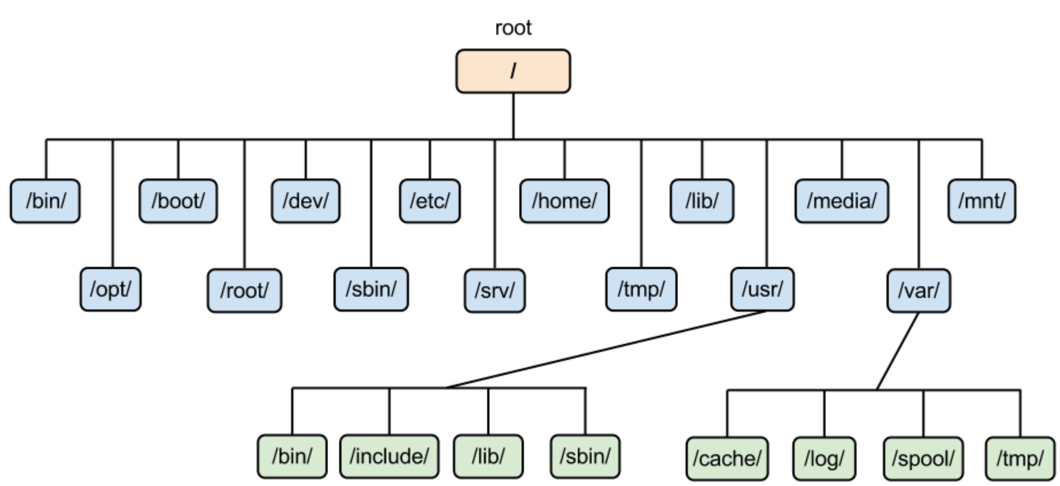


Figure 4. Linux system hierarchy structure.

The *system management tool* operates as a background process to self-start activities in the user space after booting up. The system was adopted in the Linux OS after a large number of Linux distributions replaced older system daemon tools (e.g., Unix System V and Berkley Software Distribution (BSD) with Systemd serving as an initialization system). Note that the system is written **systemd**—not **system**

D or **SystemD,** as it is system daemon and resides under Unix/Linux **d**, which is lower case [55]. Systemd is a cost-free software. Its system management tool can be redistributed or modified under GNU Lesser General Public License. Systemd provides aggressive parallelization capabilities, uses B-bus socket to start services, monitors processes using Linux control groups, and includes automatic logging daemon to save system activities as system journals.

The importance of having a lightweight *database management tool* has emerged as a key concept in IoT systems for processing data on the edge device before sending it to the cloud, saving internet data traffic and offers additional beneficial information. The tool increases IoT system reliability by decreasing data loss. Collected data is maintained as a backup source when connection with the cloud is interrupted or lost. The project reported herein used MySQL lite, which proved to be a very powerful, embedded rational database management tool for the proposed IoT system. It is an open source SQL databases implementation for embedded systems; supports various data types, provides single user access sufficient for IoT applications, and consists of one file on the storage disk [56].

Real-time connectivity with the cloud is supported by a javascript engine. Device-to-cloud communication involves an IoT device connected directly to an internet cloud for exchanging data and controlling message traffic. Many protocols have been used for device-to-cloud communication (e.g., HTTP, CoAP, TLS, DTLS, TCP/IP, UDP/IP). Javascript is currently dominating the physical world due to its event driven feature, non-blocking model. Node.js [57] is a lightweight engine used in the proposed system and paired with the Websocket communication protocol to complete the communication app architecture. Websocket is a full standalone duplex communication TCP-based protocol. The apps only relationship to HTTP was during the handshake stage when interpreted by HTTP servers. Figure 5 illustrates the handshake and data transfer sequence. The Websocket protocol enables interaction between the embedded device and the cloud, allowing messages to be transmitted while keeping the connection open [58].

A *Javascript Object Notation (JSON) message* format is used to transfer data from REECE to the cloud. JSON is a standard format that uses human readable text to transmit data objects consisting of attribute value pairs [59]. A typical communication system will consist of four different types of messages: registration message, single data sample message, multiple data samples message, and acknowledgment message. A description of each is provided below.

1. Registration message: A message sent from REECE to the cloud for validating the security key with the cloud and authorizing REECE to transfer data to the server.
2. Single data sample message: A message sent from REECE to the cloud for one measurement of data, often in real-time.
3. Multiple data samples message: A message sent from REECE to the cloud for data collected as a group of measurements, often when there is no connection with the cloud.
4. Acknowledgment message: A message sent from the cloud to REECE to confirm that data sent to the cloud was received without interruption.

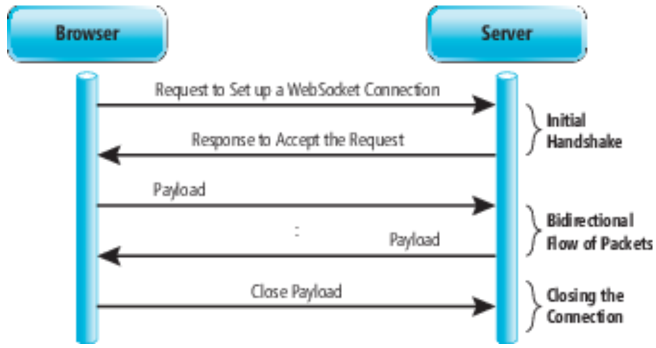


Figure 5. Websocket connection.

Maintaining continuous connectivity with the cloud is the primary concern for any IoT system. Hence, it is important to mention the worthiness of developing a watchdog timer for the internet modem. Internet broadband modems might fail to keep a constant internet connection with cloud for any number of reasons (e.g., losing coverage, saving energy in rural areas during the overnight period via ISP shutting down base stations, or addressing hardware fault due to the modem). These considerations prompted the development of a timer to verify internet connectivity and reboot the modem in the event that internet connection failed. Figure 6 describe the watchdog work flow to check internet connectivity.

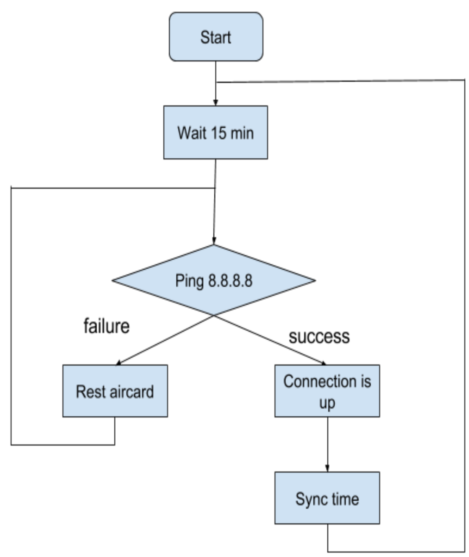


Figure 6. Internet connectivity watchdog.

Chapter 4 - Bluetooth For Travel Time Estimation

Background

Brief idea on Bluetooth Protocol:

A Bluetooth (BT) device can function as either a master or slave at any given point in time. As master, the device initiates data exchange; as slave, the device responds to a master. A BT network (i.e., piconet) can be either point-to-point, including one master and one slave, or it can include several slaves and one master. Notably, a slave can request a role change at any time and become master of the piconet, causing the master to switch to slave mode. Earlier generation BT devices transmit 1Mbps (i.e., basic rate); later generations transmit up to 3 Mbps, which is referred to as Enhanced Data Rate (EDR).

Frequency of Bluetooth devices:

Although BT devices operate in the 2.4 GHz unlicensed band, they do not transmit at a fixed frequency. Instead, they hop across a range of 79 frequencies (from 2402 to 2480 MHz), staying at each frequency for 625 us (i.e., one time slot) and followed by a hop to another frequency.

Hopping sequence is not in order. Rather, it is calculated from the master's device clock, as well as the BT device MAC address. Since all devices in a piconet must hop at the same sequence, all will follow the hopping sequence of the master. In other words, the piconet master is responsible for synchronizing connected slaves.

MAC address of a Bluetooth device:

Each BT device has a unique MAC address with 48 bits, consisting of three parts, as explained in Figure 7:

NAP 16 bits	UAP 8 bits	LAP 24 bits

Figure 7. BT MAC address structure.

1. NAP (Non-significant Address Part) is 2 bytes and refers to the manufacturing company.
2. UAP (Upper Address Part) is 1 byte and is detected by observing several BT packets.
3. LAP (Lower Address Part) is 3 bytes and is the one part transmitted with every BT packet.

BT sniffers are interested in only UAP and LAP, as these are sufficient to identify and connect to a BT device. The easiest part to detect is LAP, as it is transmitted as clear text (i.e., not encoded) in the header of every packet.

Bluetooth Packet Contents:

Figure 8 depicts a simplified BT packet:

Access Code (72 bits)	Header (10 bits)	HEC (8 bits)	Playload (0 2745) bits	CRC (16 bits)

Figure 8. BT packet structure.

The following information is contained in the BT packet.

- LAP of node master device is found in the Access Code.
- Header contains the MAC address of the slave device with transmitting packet.
- HEC (Header Error Check) and CRC (Cyclic Redundancy Check) are used to calculate the Master device UAP.
- Payload is data.

LAP of piconet master is included in every transmitted BT packet. This is important because, as shown below, Uberooth can detect LAP addresses for transmitted packets.

Ubertooth

Ubertooth (See Figure 9) is an extremely inexpensive ($119 USD) BT sniffer invented by Micheal Ossmann. Although the device's main purpose is decoding BT packets, it is also responsible for determining the master piconet MAC address, obtaining the packet's Signal to Noise (SNR) ratio, uncovering the master device's hopping sequence, and, finally, utilizing the spectrum analyzer to show any Wi-Fi or BT activity in the 2.4 GHz band.

Figure 9. Ubertooth.

Ubertooth consists of an antenna, RF front end, and a wireless transceiver composed of two integrated circuits, which are responsible for conditioning the received signal and preparing it for processing by the micro-controller (i.e., ARM Cortex-M3)—also located on the Ubertooth board [60].

Travel time estimation methodology using Bluetooth

To estimate TT based on detected vehicles traveling from checkpoint A to checkpoint B, a set of processes must be applied on the measurements. Figure 10 represents a typical BT traffic monitoring system.

Note: since the term MAC address is more recognized than LAP. This thesis uses the term MAC to describe detecting a BT device. To clarify, readers should understand that the sniffer only detects the Lower Address Part (LAP) of the BT MAC address.

System Installation:

Multiple Ubertooth sniffers were deployed with variant distances on highways to detect vehicle MACs.

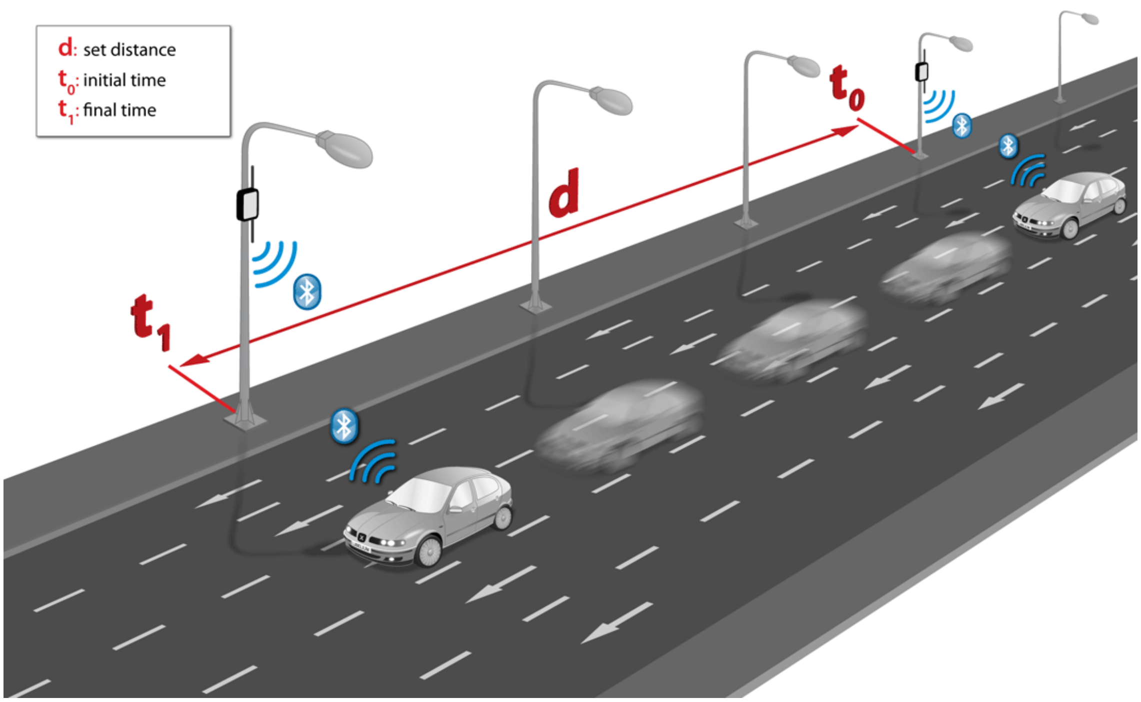


Figure 10. Bluetooth traffic monitoring system.

Removing unwanted duplicated MACs:

Ubertooth was configured to sniff BT device MACs traveling on the highway. Since sniffing MACs occurs in just microseconds and estimated time for a vehicle to remain in the Ubertooth sniffing zone is more than 1 second, the same MAC might be detected multiple times. Consequently, modifications to the proposed system were made to eliminate duplicated MACs before uploading to the cloud. A buffer can store 128 MACs. Hence, for every newly detected BT device, a window search occurs inside the buffer looking for duplicates among the last 32 MACs. Two considerations were made. First, eliminate Ubertooth dongle MAC. Second, state when the index of buffer is smaller than the size of the searching window (i.e., index of the buffer < size of search window 32). A search operation continues to look for duplicates among LAPs stored in the end of the buffer. Figure 11 describe the flowchart for eliminating BT duplicates.

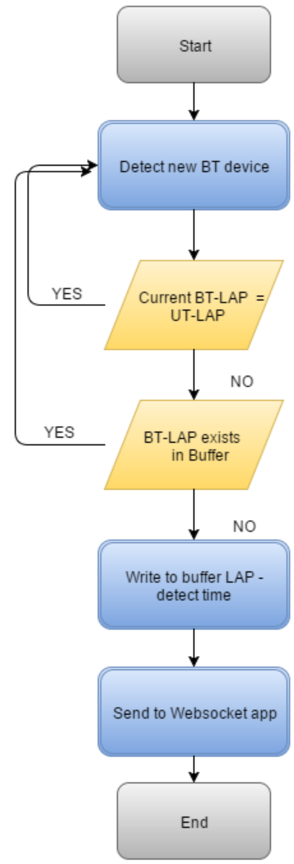


Figure 11. Flowchart of BT eliminating duplicates.

Data Collection:

Collected data is saved on REECE before transmitting it to a server on the cloud using a simple MySQL lite database. Each MAC address is accompanied by a time stamp and unique ID. The database table structure shown in Figure 12.

Field	Type	Null	Key	Default	Extra
Seq	int(11)	NO	PRI	NULL	auto_increment
MAC	char(12)	YES		NULL	
Det_TIME	datetime	YES		NULL	
Trans_TIME	datetime	YES		NULL	
Exp_TIME	int(11)	YES		NULL	
ACK_TIME	datetime	YES		NULL	

Figure 12. Bluetooth MySQL database on REECE.

The database consists of one table with six columns, as follows.

- Seq: Sequence of BT device in the database
- MAC: MAC of the detected BT device
- Det_TIME: Detection time of BT device
- Trans_TIME: Transmission time of information packet to the server
- Exp_TIME: Time to determine it is unnecessary to send detected BT device information; fixed value not in use
- ACK_TIME: Acknowledgment time wherein REECE receives a message from the cloud containing sent packet sequence.

Vehicle Re-identification:

A vehicle can be re-identified by searching for the same MAC address at consecutive checkpoints. Matched MAC addresses provide a TT value based on differences in detection times. Vehicle speed is based on A known distance between checkpoints.

TT per vehicle can be calculated using the following equation:

$$TT_{per\ vehicle} = DT_b - DT_a$$

where DT is the detection time at one checkpoint.

Outliers can occur when one vehicle travels between two points on a different journey. For an accurate estimation, outliers must be identified and removed. The following three-step process is proposed for this purpose.

1. Find TT distribution of observed segment.
2. Calculate the mean TT and standard deviation σ.
3. TT is considered an outlier given it meets the following equation:

$$TT_{per\ vehicle} > TT_{mean} + 2\sigma$$

<u>*Travel time estimation:*</u>

TT can be estimated by finding the mean of the distribution of TT per vehicle measurements. Figure 16 shows the distribution of TT over one segment.

Experiments and Analysis Results

Many experiments were conducted to collect and analyze travel data to achieve an accurate estimation for TT. Using single and multiple BT stations, we can estimate number of vehicles, TT, and speed estimation.

<u>*Multiple Site Deployment:*</u>

The deployment was tested during March 2017 in Tulsa, OK. Details are summarized below.

1. The testing deployment had seven units running throughout the city of Tulsa, OK during a10 day period.

2. TT calculated between the selected locations were approximately equal to Google TT estimates.
3. Traveling vehicles between locations were detected at distances of 5 and 10 miles.

Data insights were made following data collection. Daily average of detected vehicles was over 5.000 per day. Table 2 provides detailed information for each unit. Figure 13 shows the location of units deployed in the city of Tulsa.

Table 2. Tulsa, OK Deployments

station ID	Daily average Detected Vehicles	Antenna	Location and Distance from Highway
BT-054	8790	5 dbi internal	Side highway, apx 1 meter far
BT-061	5140	5 dbi internal	Under Bridge , apx 8 meters far
BT-062	4560	3 dbi internal	Above highway, apx 15 meters far
BT-063	6010	3 dbi internal	Side highway, apx 1 meter far
BT-064	3950	5 dbi internal (tilted)	Side highway, apx 1 meter far
BT-065	6160	3 dbi internal	Side highway, apx 1 meter far
BT-066	5550	3 dbi internal	Above highway, apx 5 meters far

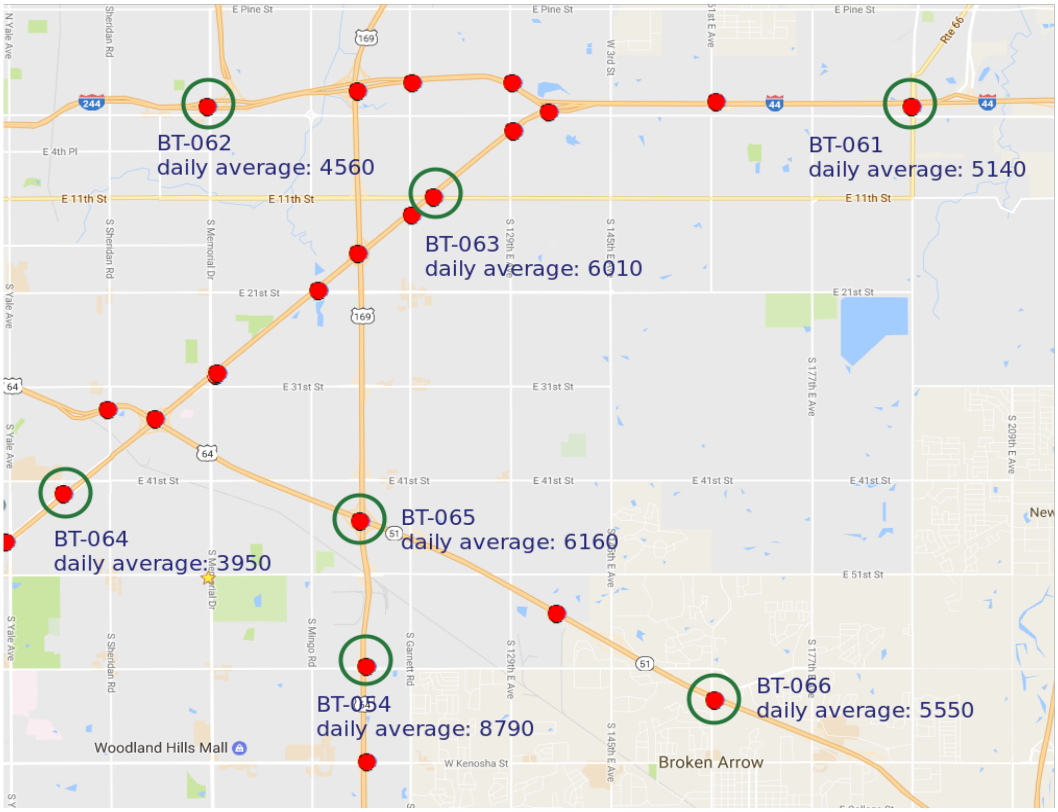


Figure 13. Locations of deployed units in Tulsa, OK.

Deployment-calculated TT was approximately equal to Google TT estimates. An average of over 200 vehicles were detected per day between two locations among all deployed units.

Number of detected vehicles depends on a number of factors, including:

1. distance between the two locations,
2. traffic flow,
3. attractions approximate to deployed units (e.g., downtown, main apartments complexes, city exits), and
4. antenna type/distance between cabinet and the highway.

Figure 14 and Table 3 provide detailed information for detected vehicles traveling between two locations.

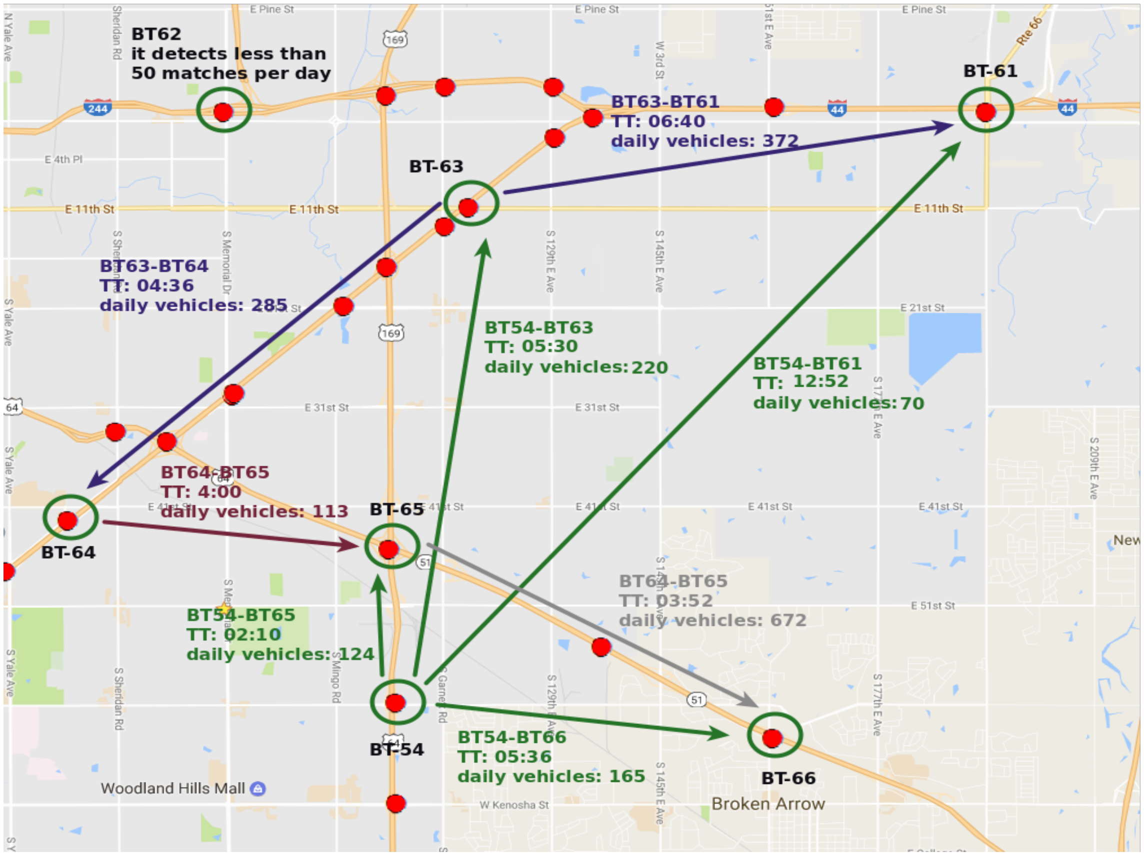


Figure 14. Average TT and average daily detected vehicles between two locations.

Table 3. Average TT and Average Daily Detected Vehicles

Units	Average TT	Average daily detected vehicles
BT-054 BT-061	**12:52**	**70**
BT-054 BT-062	07:35	34
BT-054 BT-063	**05:30**	**220**
BT-054 BT-064	06:21	67
BT-054 BT-065	**02:10**	**124**
BT-054 BT-066	**05:36**	**165**
BT-061 BT-062	08:00	45
BT-061 BT-063	**06:40**	**372**
BT-061 BT-064	10:30	25
BT-061 BT-065	11:05	4
BT-061 BT-066	13:20	6
BT-062 BT-063	07:44	8
BT-062 BT-064	07:38	10
BT-062 BT-065	06:40	38
BT-062 BT-066	10:00	11
BT-063 BT-064	**04:36**	**285**
BT-063 BT-065	05:10	22
BT-063 BT-066	08:50	9
BT-064 BT-065	**04:00**	**113**
BT-064 BT-066	07:34	55
BT-065 BT-066	**03:52**	**672**

Study of Travel Time between Two Units

In this section, the results of data collected between two different road segments will be discussed.

TT between Units BT-061 and BT-063

Distance between units BT-061 and BT-063 is 5.2 miles. BT sniffers detected an average of 372 vehicles/day and calculated TT of 06:40, which is one minute longer than Google estimates. However, road work may have had an effect on the difference between the two values. During rush hour (time between 2:00pm and 06:00 p.m.), the hourly rate of detected vehicles was approximately 40.

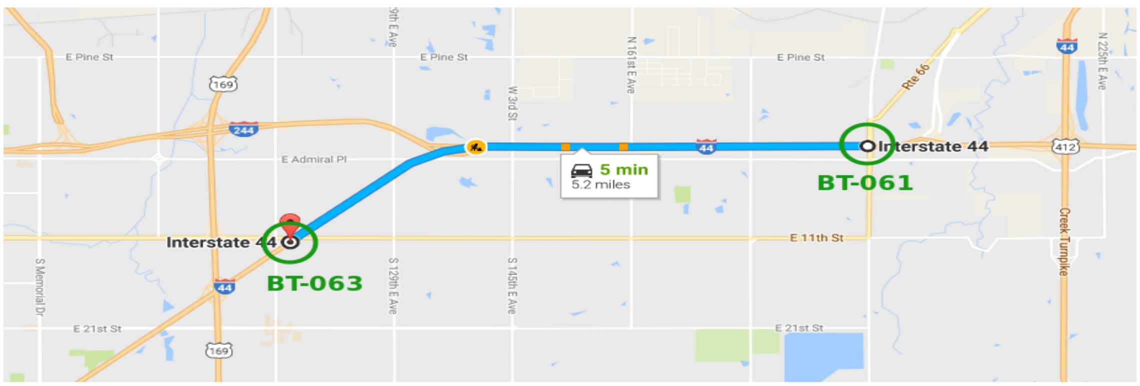


Figure 15. Bluetooth monitoring system for 5.2-mile distance on I-44 in Tulsa, OK.

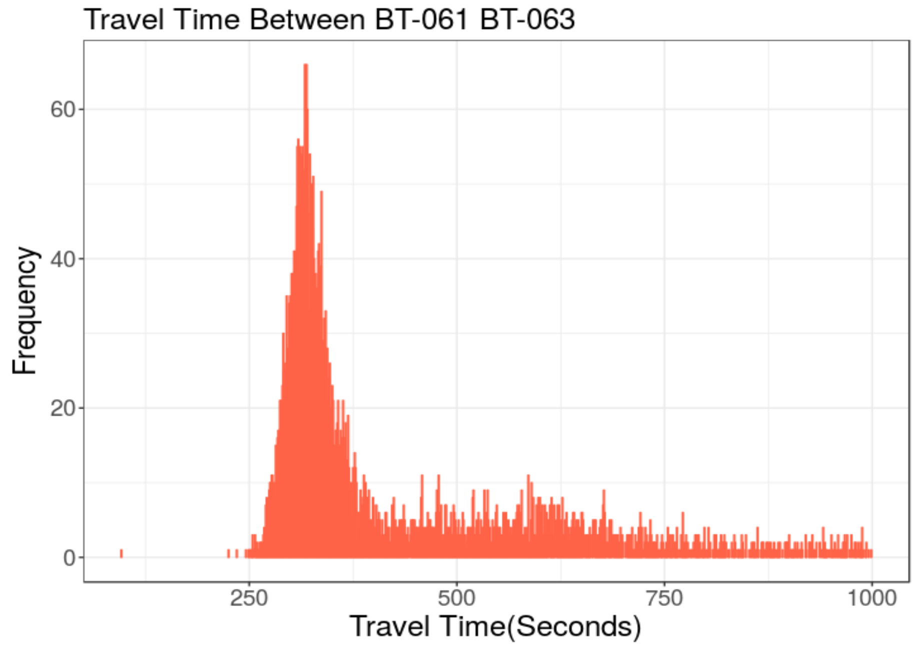


Figure 16 TT distribution

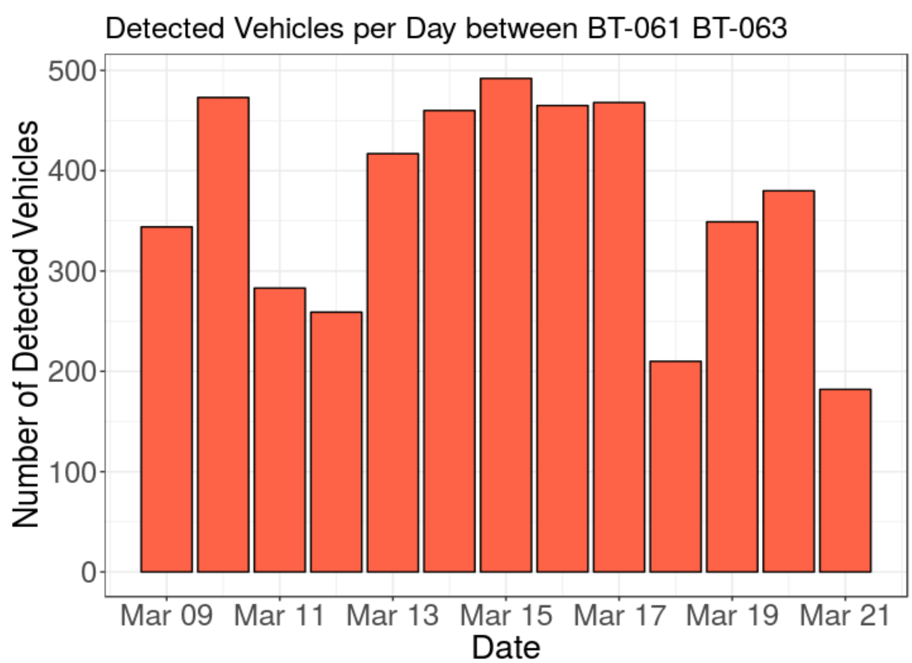


Figure 17 TT values based on Bluetooth on I-44 in Tulsa, OK

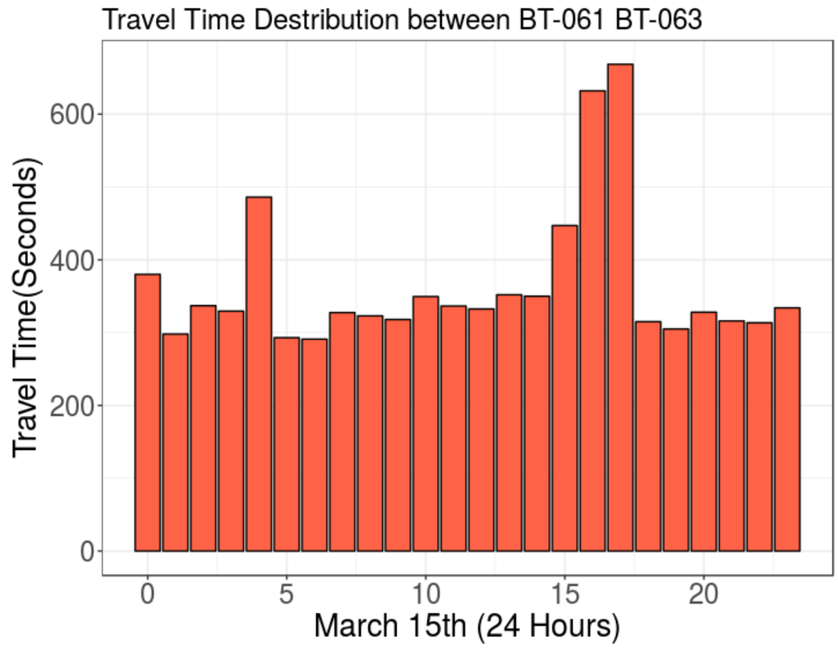


Figure 18 Hourly Average TT on A Single Day

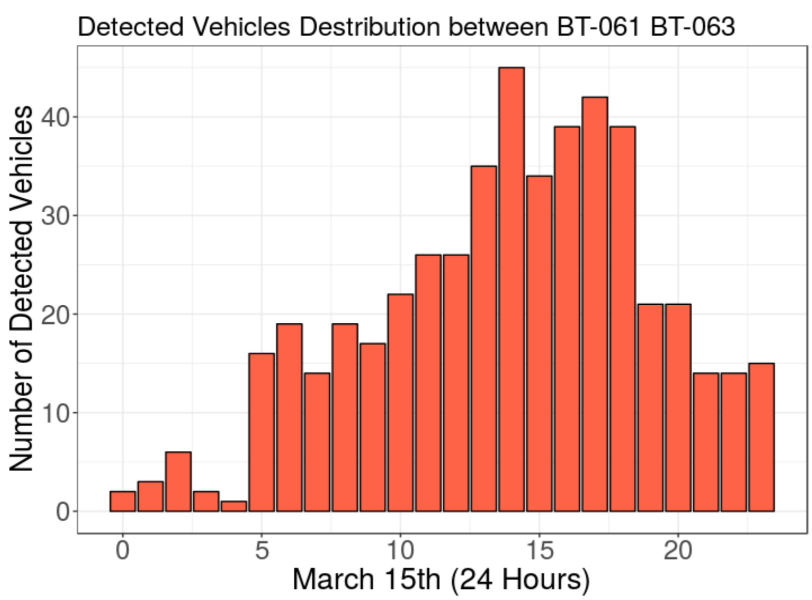


Figure 19 Hourly Number of Detected Vehicles on A Single Day

Two factors of rush hour traffic and the presence of road work increased TT by more than 200 seconds.

Units BT-065 and BT-066

Distance between unit BT-065 and BT-066 is 4.1 miles. BT sniffers were able to detect an average of 672 vehicles/day and calculate TT as 03:52, which is the same as the Google estimate. During rush hour (between 03:00 and 05:00 p.m.), the hourly rate of detected vehicles was approximately 70 vehicles. Figures [20, 21, 22, 23, 24] show the location of the segment, TT distribution, daily number of detected vehicles, hourly number of detected vehicles on a single day, hourly average TT on a single day.

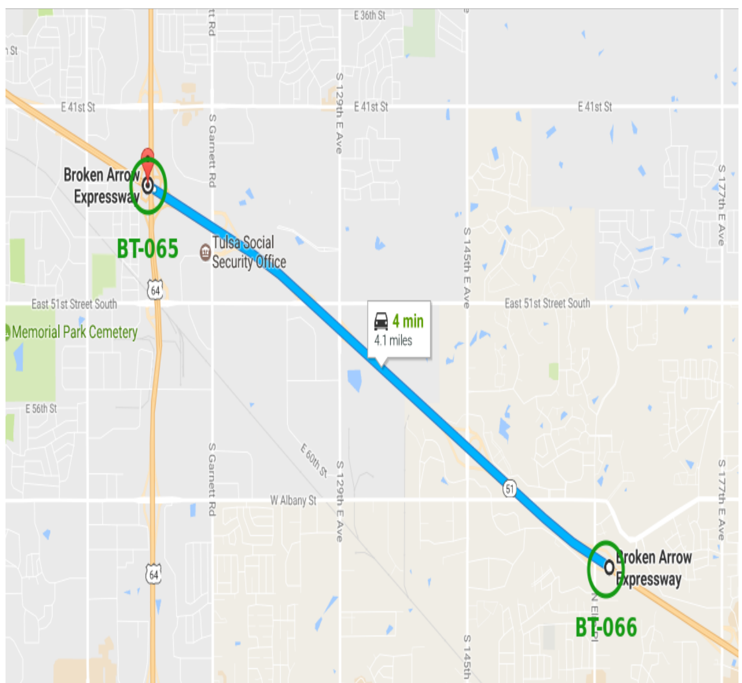


Figure 20. Bluetooth monitoring system for 4.1-mile distance, on BA-Expressway in Tulsa, OK.

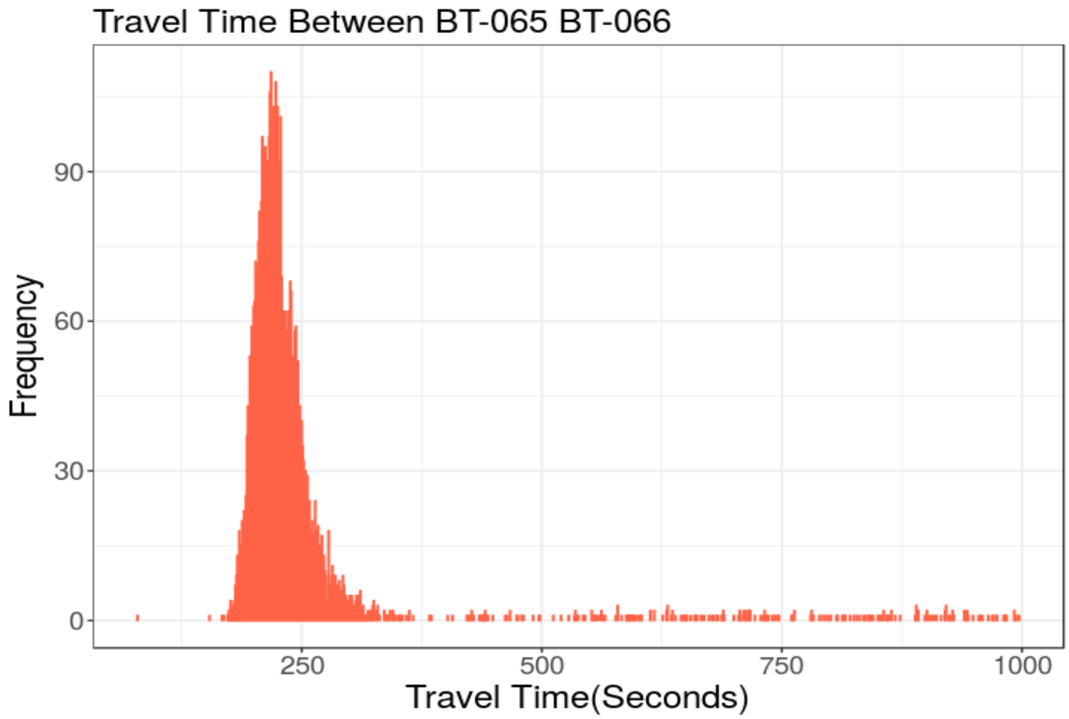


Figure 21. TT values based on Bluetooth on BA-Expressway in Tulsa, OK.

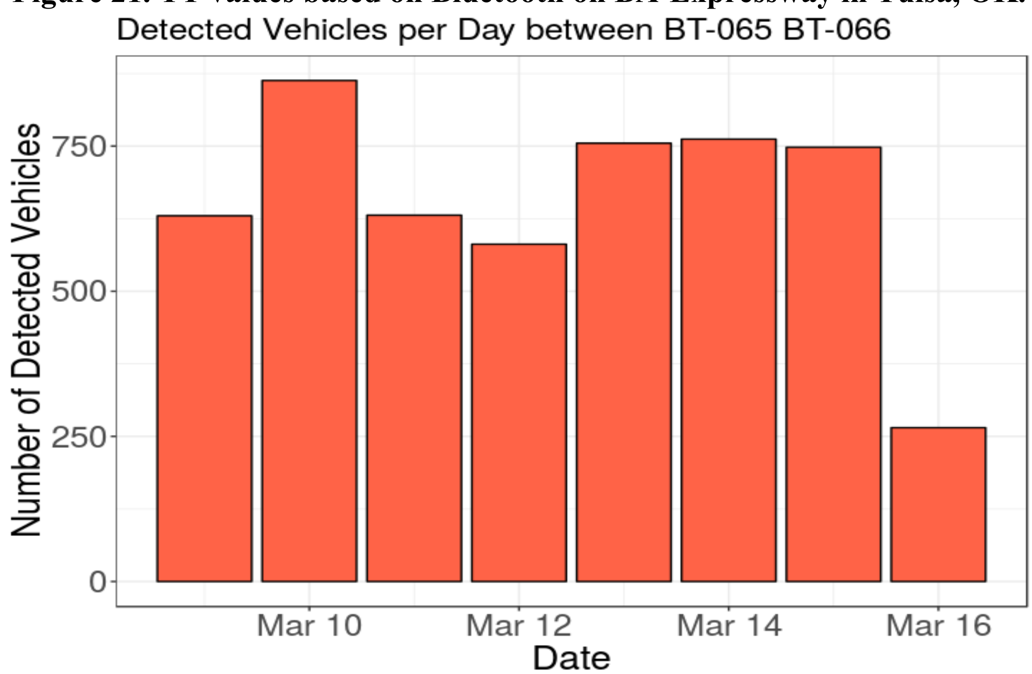


Figure 22. Number of detected vehicles on one segment 4.1-mile length, over 5-Day period.

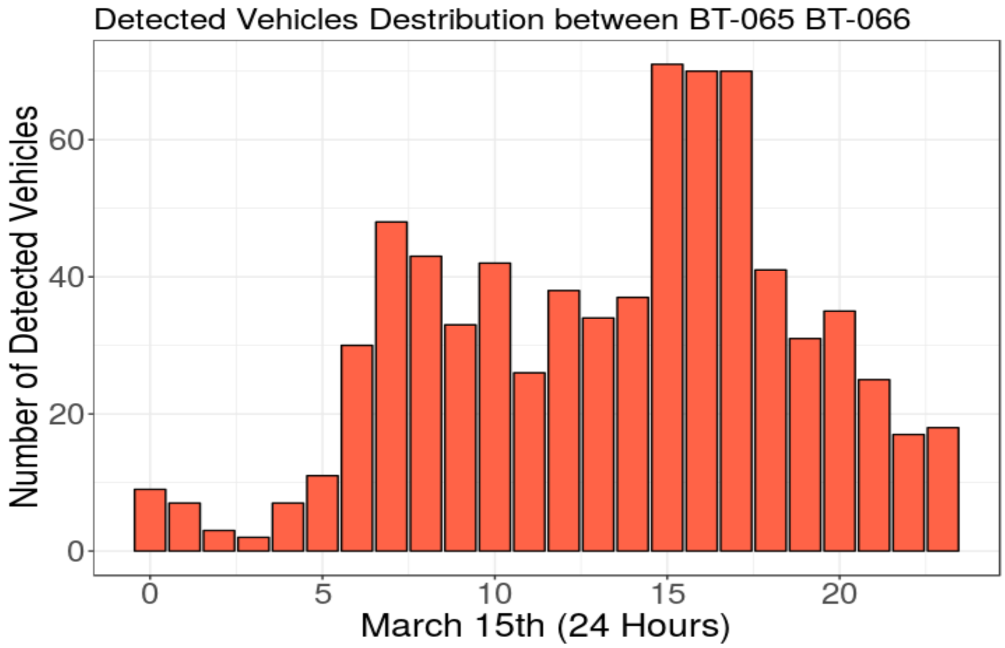


Figure 23. Hourly number of detected vehicles on a single day.

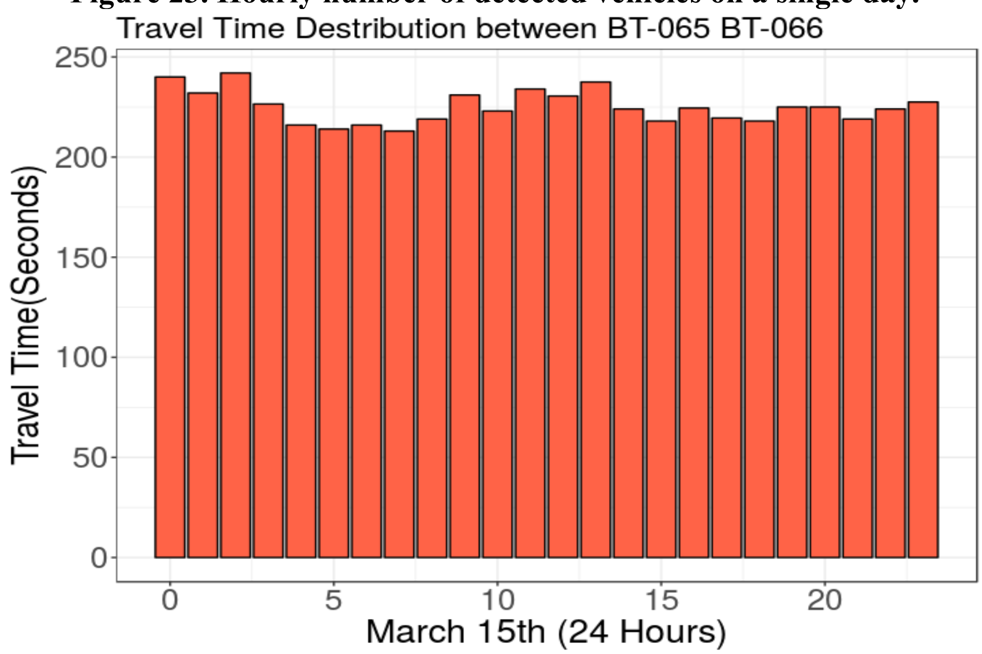


Figure 24. Hourly average TT on a single day.

No major changes were detected in TT between two locations over a period of 24 hours.

Improvements of The Bluetooth System:

Previous experiments show that the number of vehicles detected between two locations is extremely low. An alternate antenna was tested to improve traffic system penetration. Table [5], depicts two antennas considered for BT detection.

Table 5. Bluetooth Sniffer Antennas

	
The used antenna in the previous experiments is an internal omni-directional antenna with 5dBi gain and 50 Ω impedance.	Alternative antenna: TP-Link TL-ANT2414A) An External directional antenna with 14dBi and 50 Ω impedance.

Figures [25, 26] show a substantial change in number of detected vehicles when using an external directional 14dBi antenna instead of internal 5dBi antenna.

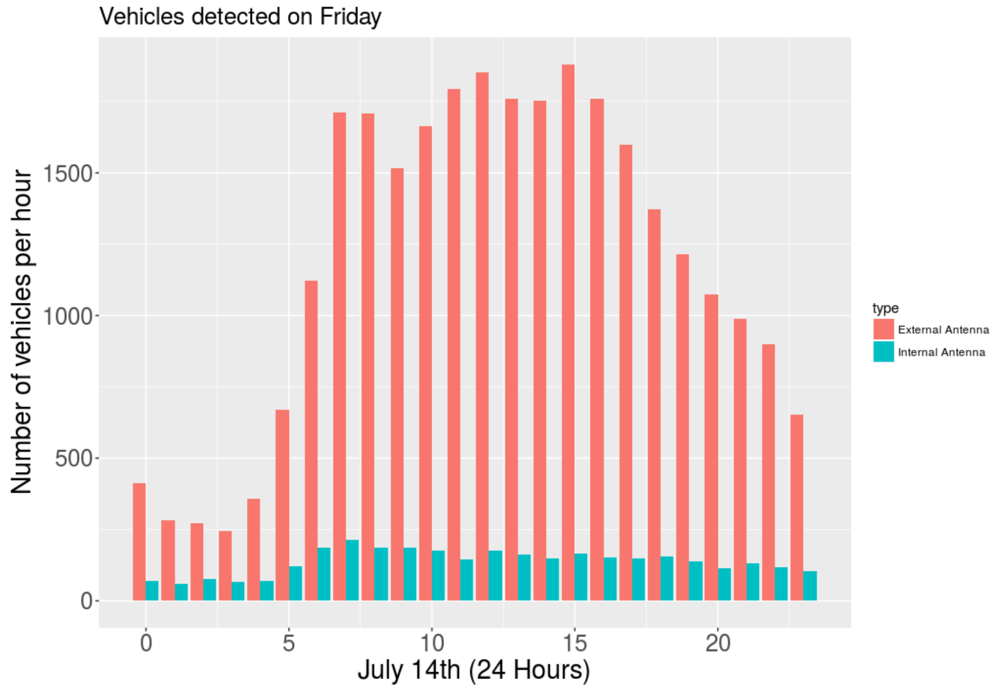


Figure 25. Hourly number of detected BT devices on a single site.

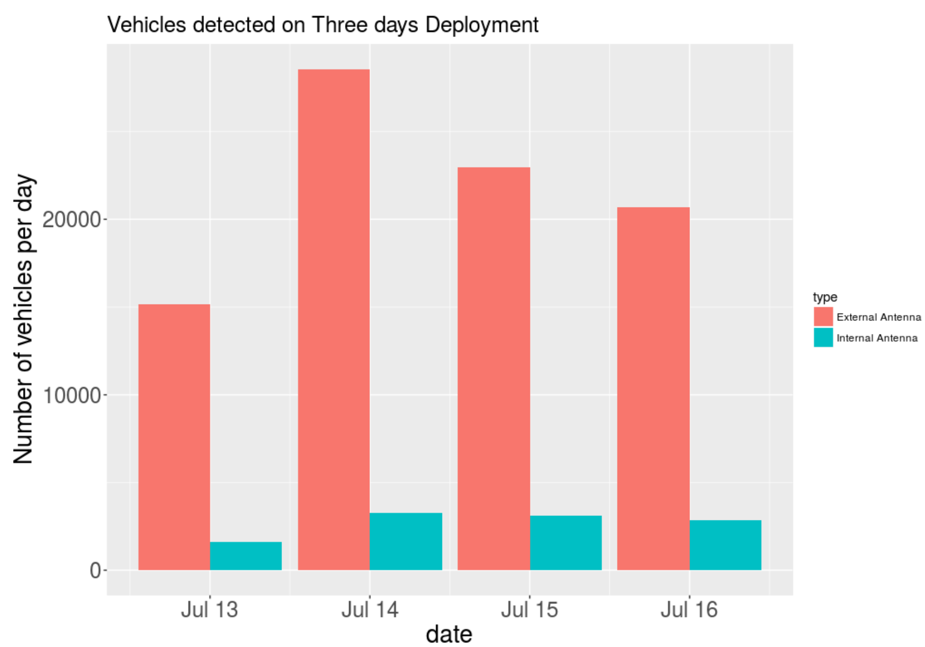


Figure 26. Daily number of detected BT devices on a single site.

Chapter 5 - Inductive Loops Detectors for Travel Time Estimation

Background

An inductive loop consists of wire "coiled" to form a loop shaped as a square, circle or rectangle that can be installed into or under the surface of the roadway. Inductive loops work like a metal detector, measuring the change in magnetic field when objects pass over them. Once a vehicle drives over a loop sensor, the loop field changes and the detection device detects the presence of an object (e.g., vehicle). Inductive loops are referred to as presence detectors. Traffic detection are often used in combination with axle sensors to collect classification data, such as speed and length.

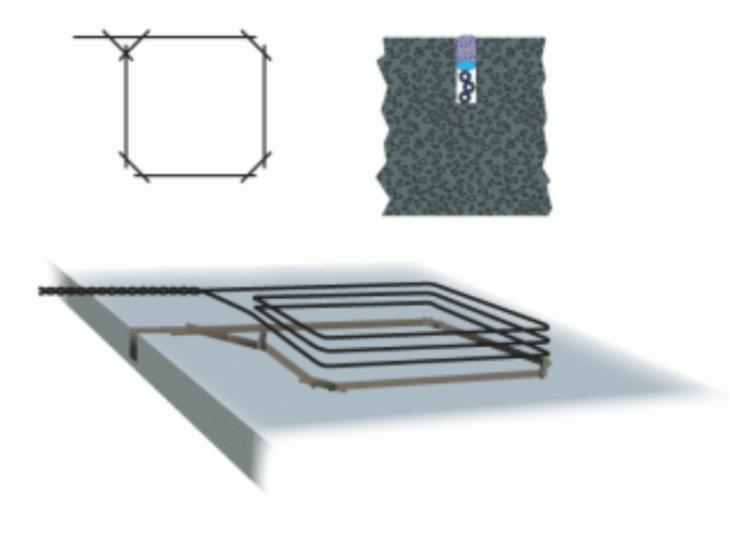


Figure 27. Square inductive loop.

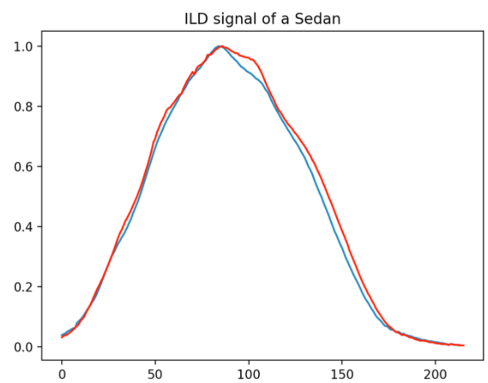
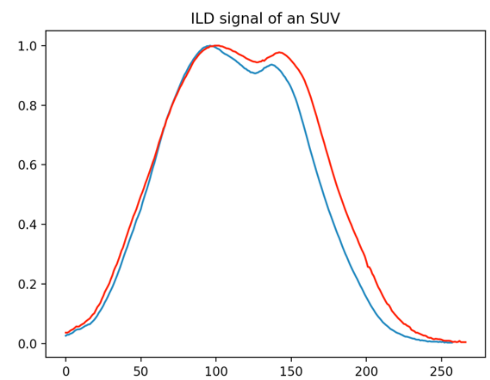
There are two commonly used shapes for inductive loops: rounded loops or square/ rectangular loops. Many loop designers have theorized that circular loops provide optimum detection because a uniform magnetic field is produced without dead spots. Proponents of the round loop argue that the circular design maximizes loop sensitivity for detection of motorcycles, as well as high-bed trucks while eliminating splash over from adjacent lanes. Other cited advantages include the elimination of sharp corners and the reduction in wire stress. Modern cutting techniques have mitigated difficulties associated with cutting a circular shape in the pavement.

The loop is a continuous length of wire that enters and exits from the same point. Both ends are connected to the loop extension cable, which in turn connects to the vehicle detector. The detector powers the loop, causing a magnetic field in the loop area. The loop resonates at a constant frequency monitored by the detector. A base frequency is established when there is no vehicle over the loop. When a large metal object, such as a vehicle, passes over the loop, the resonate frequency increases. The change is sensed and, depending on the design of the detector, forces a normally open relay to close. The relay will remain closed until the vehicle leaves the loop and the frequency returns to the base level.

Study of the Inductive Loop Signature

Generated ILD signals vary from one vehicle to another. An ILD signature depends on vehicle length, metal surface, speed, and the way in which a vehicle will pass over the loops. Table [6] shows the difference in signatures between SUV and sedan vehicles. Signature length and amplitude of the magnetic field vary. While the sedan has one peak point, an SUV has two. Also, signatures of the same vehicle will not be identical on another loop. The red and blue signatures for each vehicle represents the signature detected on the lead and lag loops, respectfully, with 8ft distance between them. Variations will increase for longer distances between two loops. ILD signals have many applications in length-based vehicle classifications due to accurate measure of vehicle length. Hence, magnetic field strength will vary among vehicles of the same class.

Table 4. ILD Signatures of Two Vehicles

Loop Shutdown

Two phenomena might cause the inductive loop to malfunction:

- Lightning Strike: Given that lightning strikes relatively close to a road-embedded loop, it is possible that a large static charge will be transmitted through the loops into the loop board circuitry. Although the loop board is equipped with a certain level of electronic protection against this type of event, lightning strikes might cause the loop board to lock up or shut down. A strike is unlikely to cause damage to a unit that is grounded [61].

- Other Electrical Noise: Similar to a lightning strike, other sources of strong electronic signals could cause the loop board to detune.

In some cases, inductive loops do not shutdown even though they add noise to the ILD signals and require retuning to fix the issue. Figure 28 show some noisy signatures.

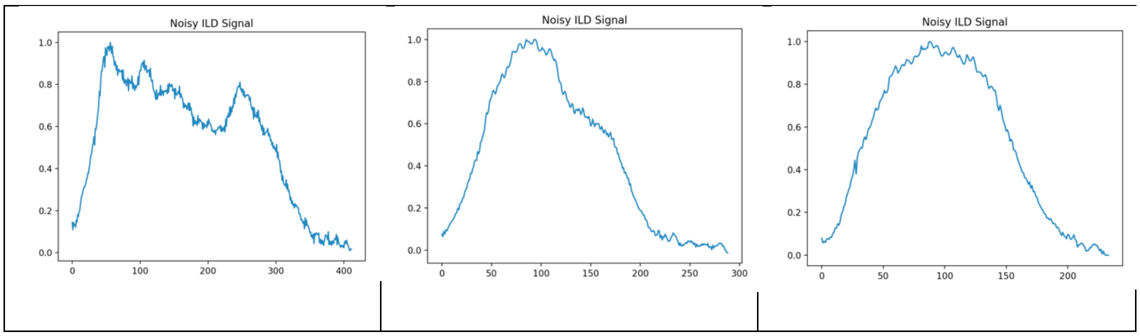


Figure 28. Noisy ILD signals.

Phoenix II Diamond Traffic

Phoenix II is a multi-lane time interval counter/classifier designed for permanent installs or large portable applications. The classifier can count from one to eight lanes using axle sensors; 16 lanes with loops; or one to eight lanes with gap, headway, and speed by axle type. The system can be fitted with four road tube sensors, two to eight remote inputs, four to 16 presence inductive loop sensors, and four to eight piezo or resistive sensor inputs.

When Phoenix II is activated in VO=CMA (vehicle output in comma delimited output), it generates the signatures as a string of a time series data sample and each vehicle signature will be separated by CR/LF.

Figure 29. Phoenix II Diamond Traffic unit.

Deployment setup

The following steps are necessary for deployment.

- Phoenix II Diamond Traffic with 1KB sampling rate is connected to a standard 6’ by 6’ rectangular inductive loop.
- REECE is connected over RS-232 with Phoenix II and over Ethernet to the cloud.

Figure 30 shows the ILD Traffic Monitoring System setup at Hefner parkway in Oklahoma City, OK.

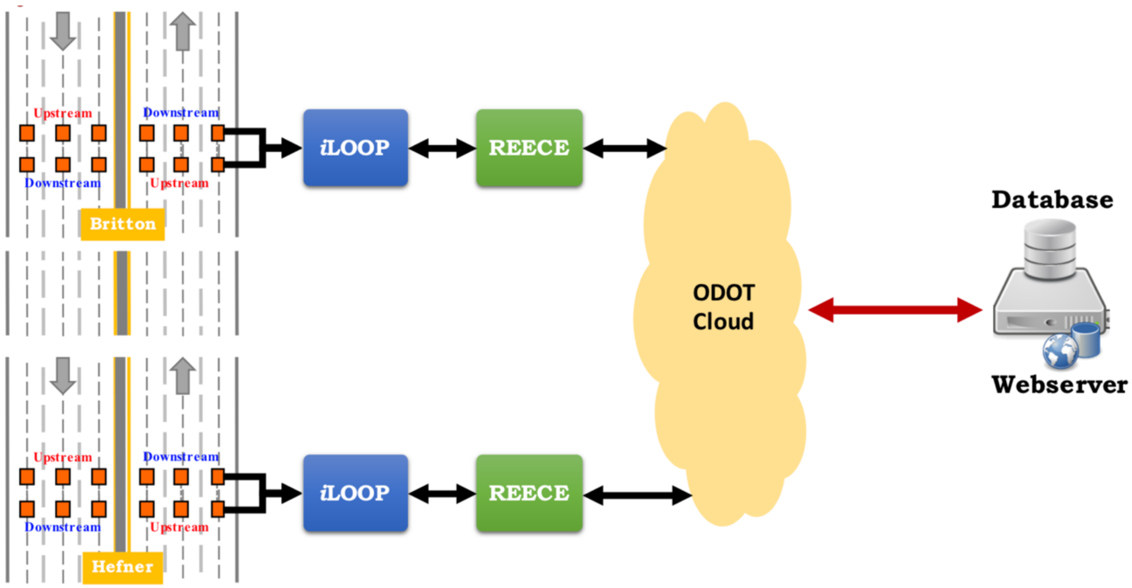


Figure 30. ILD traffic monitoring system.

Preprocessing the data

The process of signature re-identification is accomplished in the following approach.

1. Data Cleaning to ensure incomplete or distorted signatures are ignored.
2. Speed Based Normalization on the time domain.
3. Vehicles Length estimation.
4. Amplitude Normalization between 0 and 1.

Vehicle re-identification and Travel Time Estimation

1. Comparing signatures with time window less than 300 seconds.
2. Comparing signatures with vehicle length difference less than 40 cm.
3. Vehicle re-identification by way of matching signatures, accomplished through either Pearson Correlation or Relative Entropy.

5. TT calculation indicates the time difference between the best matched signatures.

6. Data spike detection.
7. TT estimation from the mean value of spike data.

Data Cleansing:

Some signatures were interrupted due to vehicle lane change or a hardware faults, or loop errors. All signatures contained interruptions that were deleted from the data. Figure 28 show a sample of a noisy signature.

Speed based Normalization:

Signature length were dependent upon vehicle length and speed. In the test setups, there were two loops: one at the upstream (lead) location and another at the downstream (lag) location. Distance between the two loops is 8ft, and detection time at each loop is provided by Phoenix II. Speed can be calculated using the following equation:

$$V = \frac{Distance}{T_2 - T_1}$$

where ***distance*** is the amount of separation between the two loops at one site; T1 is the detection time at the lead loop; and *T2* is the detection time at the lag loop.

The signature can be normalized by fixing the speed for all vehicles to 60 mph so that a new signature length can be calculated from the fixed speed.

$$l_{normalized} = \frac{l_{original} \times v_{60\ mph}}{V}$$

where *V* is vehicle speed; $l_{original}$ is signature length at speed (V); $v_{60\ mph}$ is the fixed speed for all vehicles at 60 mph; and $l_{normalized}$ is the speed normalized signature length.

Cubic Spline interpolation was applied to shrink or compress the signature to the normalized length. Cubic spline interpolation is an interpolation polynomial method that provides a smoother when compared to other interpolation polynomials [62].

The device suffered from hardware issues, causing signatures from four lanes to go undetectable on both sites. To mitigate this problem, one lane detected 90% of vehicles on both loops and was subsequently selected for the study. Vehicle speeds that was not reported by Phoenix II controller was estimated based on the speed mean of all detected speeds within the same minute. Figures [32, 33, 34] compare the number of detected vehicles between the lead and lag loops on a single site and number of detected vehicles between the two sites.

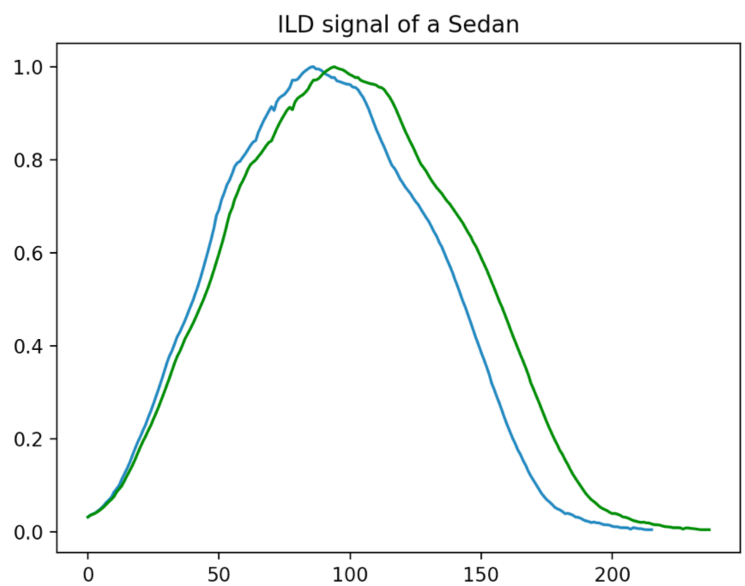


Figure 31. Normalized signature.

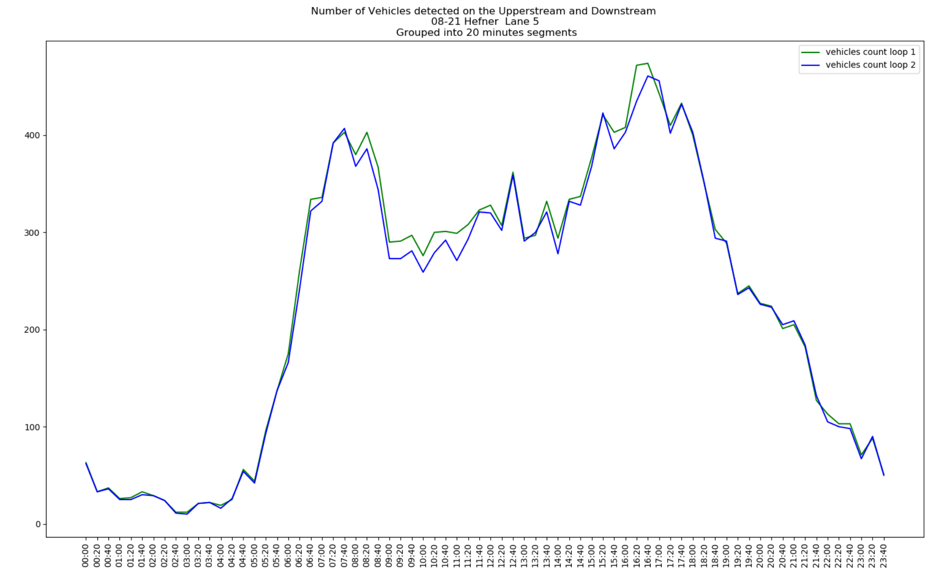


Figure 32. Number of vehicles detected on both loops at Britton site.

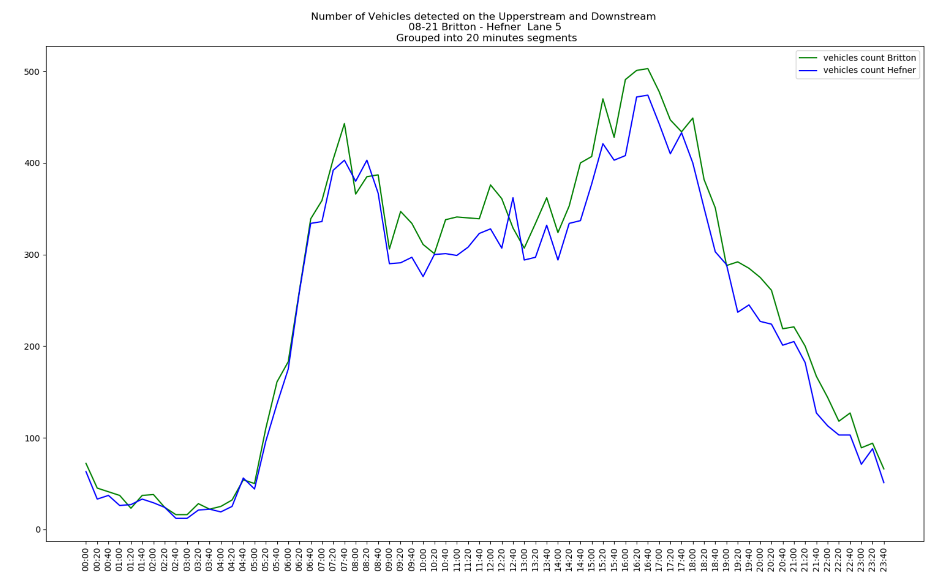


Figure 33. Number of vehicles detected on both loops at Hefner site.

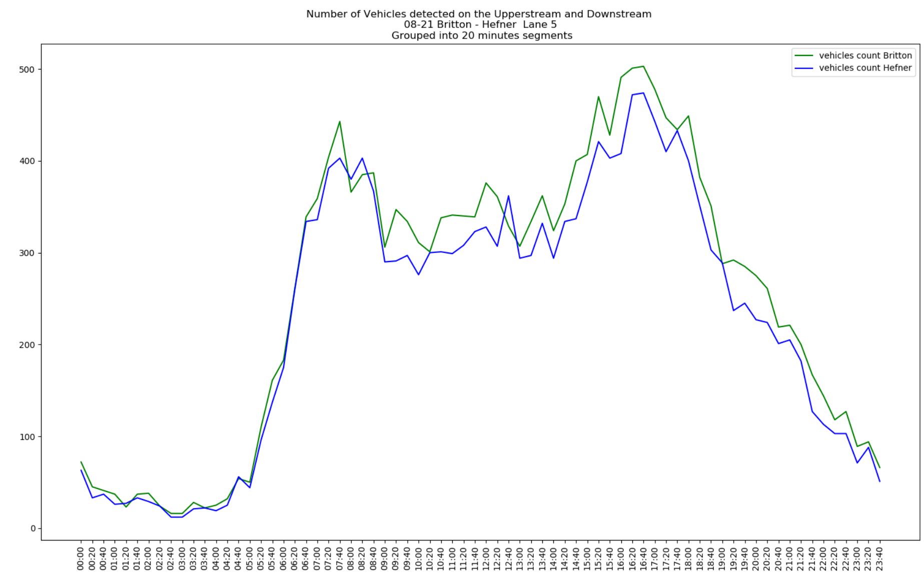


Figure 34. Number of detected vehicles on Britton and Hefner sites.

Vehicle Length Estimation:

After normalizing, the signature length based on speed, vehicle length can be estimated from the length of the signature. Phoenix II sampling rate was set to 1000/s. Therefore, the signature length can be considered the time required for a vehicle to cross over a loop. Based on this, we can estimate vehicle length from the following formula:

$$Vehicle\ Length = Vehicle\ Speed \times Signature\ Length$$

Vehicles re-identification:

Two methods were tested to determine the most correlated signatures (e.g., Pearson Correlation or Relative Entropy). Pearson Correlation was used to find linear dependences between the two signatures. Pearson Correlation depends on the shape of the signature. Two signatures are correlated if they are similar in their slope rates.

$$r = \frac{\boldsymbol{n}\sum(\boldsymbol{xy}) - (\sum \boldsymbol{x})(\sum \boldsymbol{y})}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

where n is the number of samples (signature length). When r =1, both signals are identical and correlated.

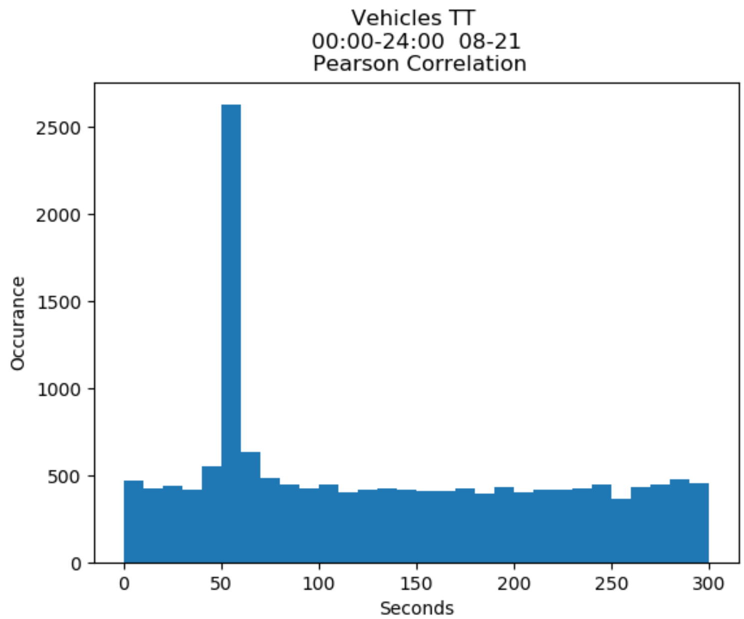


Figure 35. TT using Pearson's r Correlation.

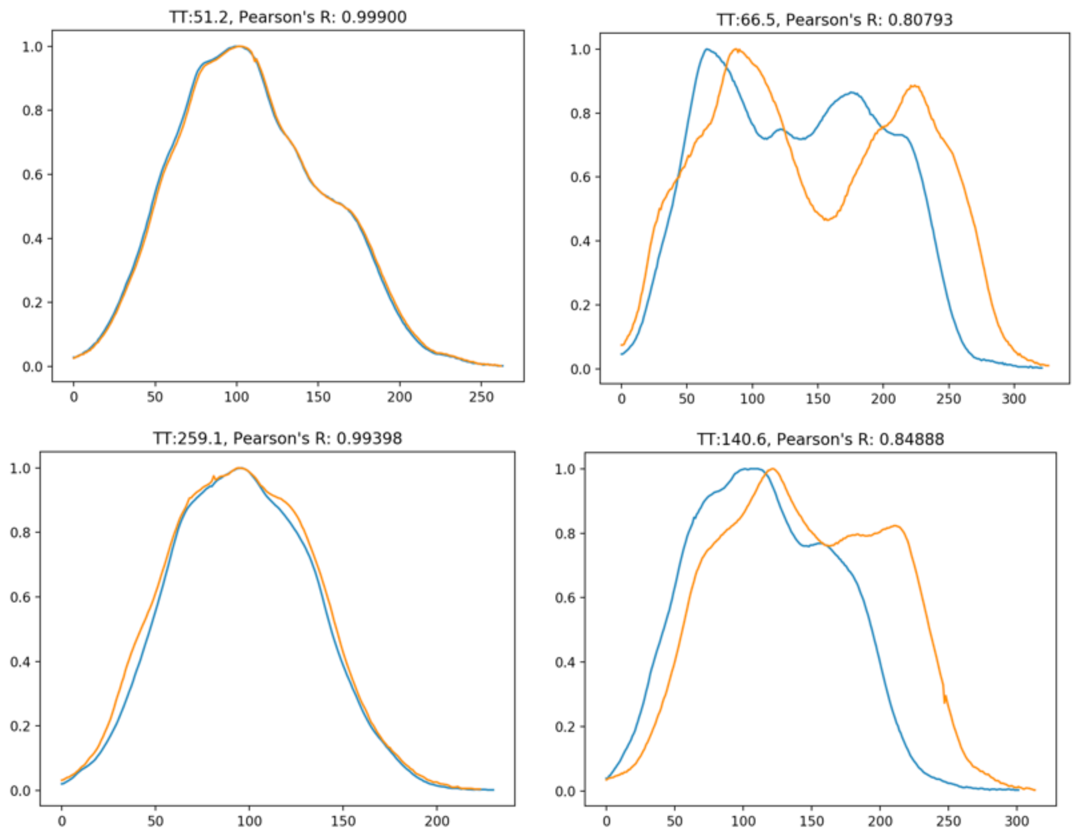


Figure 36. Correlated signatures using Pearson Correlation.

Relative Entropy assumes signatures of the same vehicle will have the same value of the area under the curve, since the amount metal in vehicles is invariant. However, slope rate and strength of the magnetic field can change based on the way a vehicle is passing over the loop. Relative Entropy was used to compute the relativity of the pdf value between two signatures.

$$d = \sum_{k}^{n} p_k \, log_2(\frac{p_k}{q_k})$$

where p_k , q_k = the probability functions for both signatures. When d = 0 =>, both signatures have the same P.D.F value and are correlated.

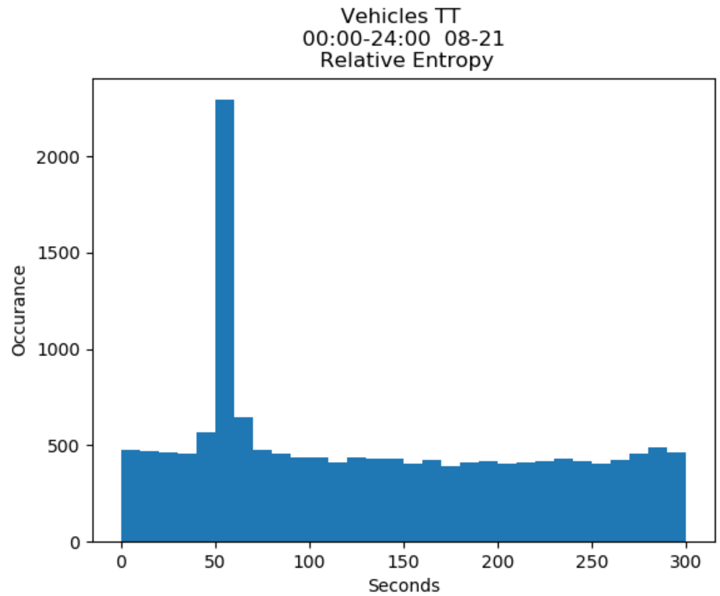


Figure 37. TT using Relative Entropy.

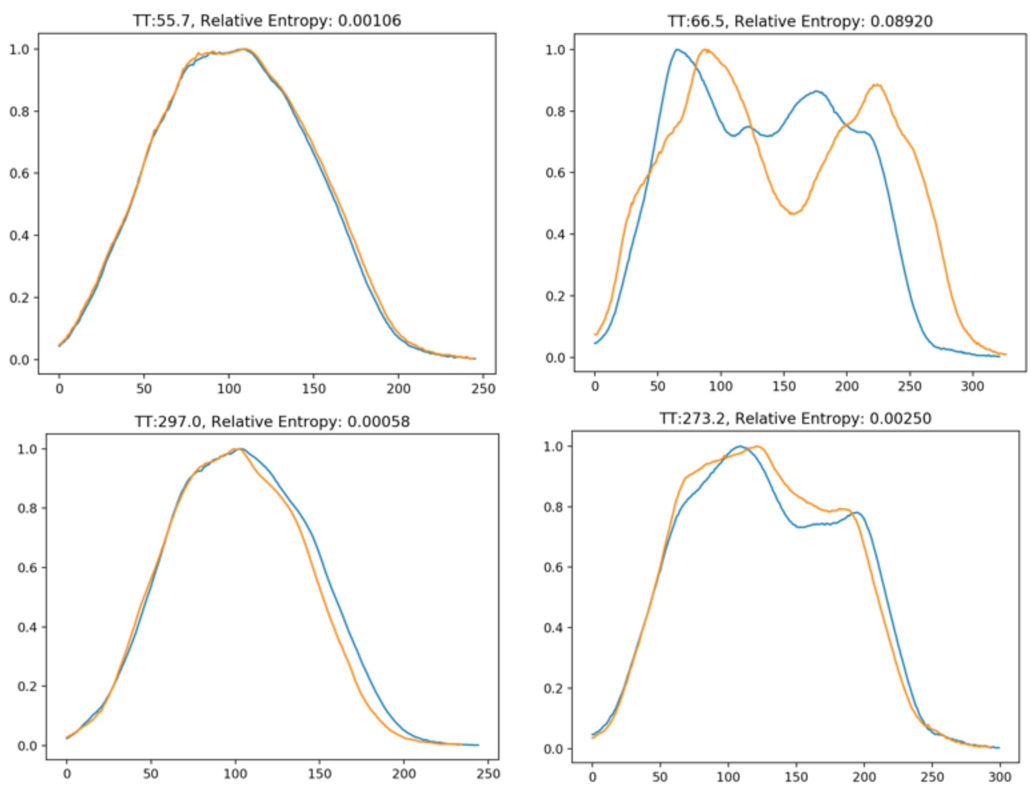


Figure 38. Correlated signatures using Relative Entropy.

Both methods were successfully used in re-identifying vehicle signatures. Relative Entropy was characterized with higher noise levels than Pearson Correlation. Both methods were applied to achieve an improvement in correlating vehicle signatures. Tables 6 and 7 show TT values of ILD traffic monitoring systems divided into five periods: 00:00-07:00, 07:00-10:00, 10:00-16:00, 16:00-20:00, 20:00-24:00. The percentage under each Figure represents total number of vehicle with TT between 45-70mph compared to total number of re-identified vehicles. Clearly, noise ratio increases during the rush hour periods from 7 to 10 a.m. and 4 to 8 p.m.

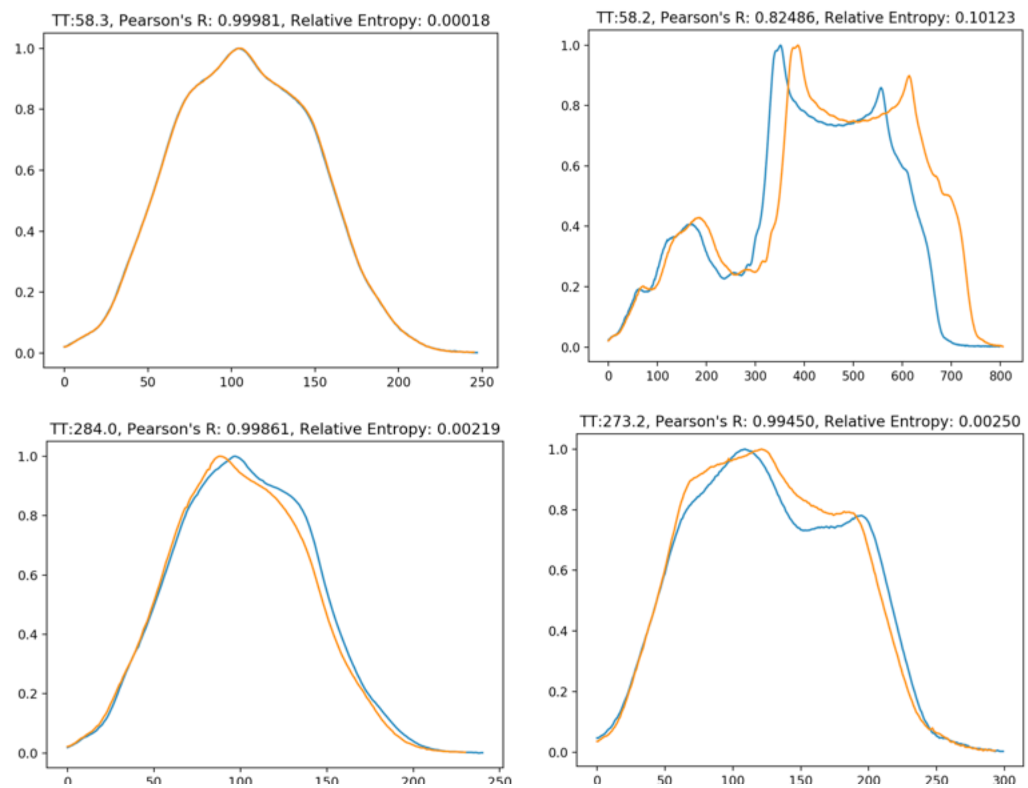


Figure 39. Correlated signatures using Pearson Correlation and Relative Entropy.

Table 5. TT Values of ILD Traffic Monitoring System on Monday

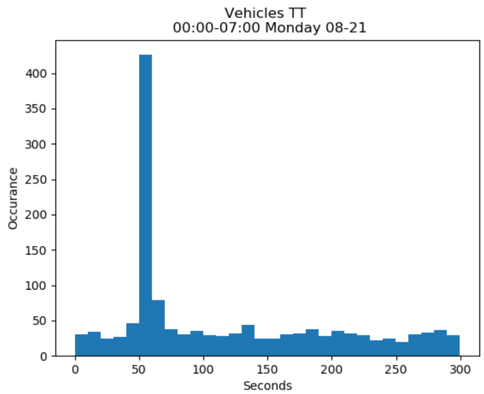
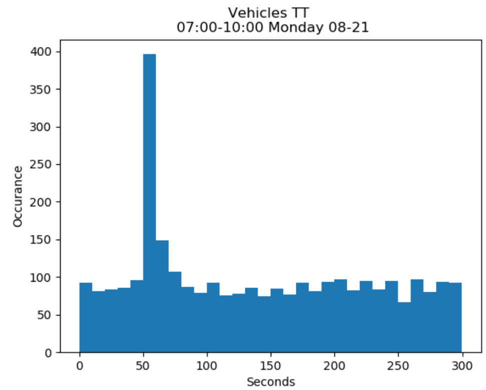
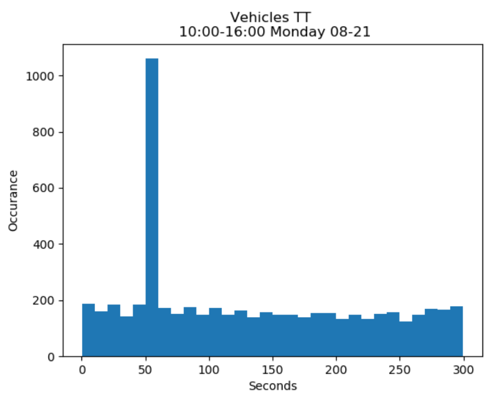
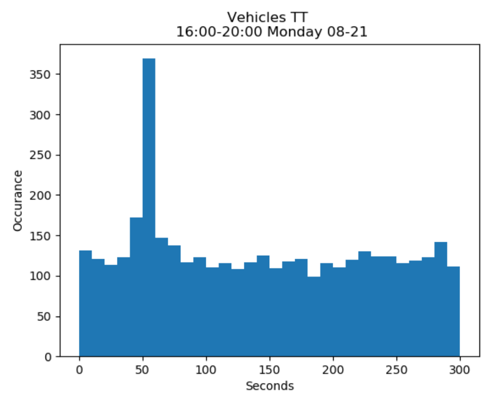
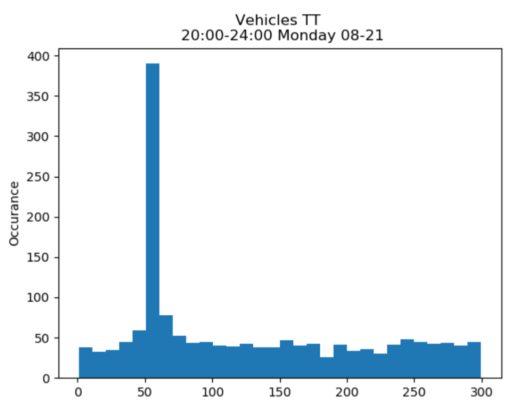
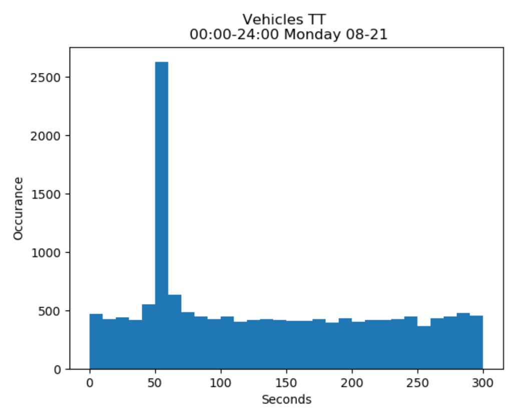
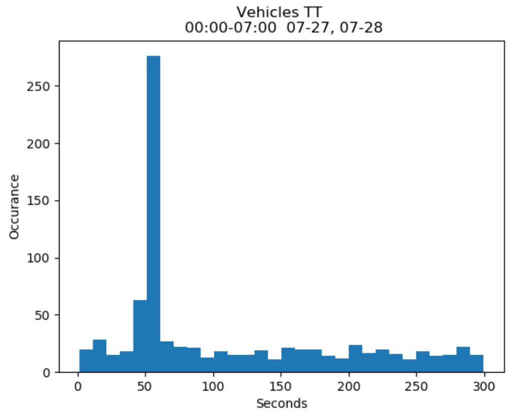
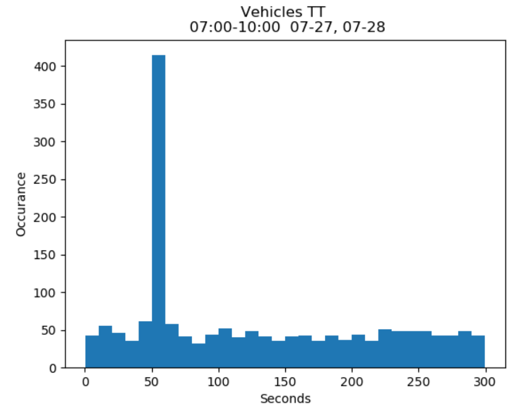
 505 / 1372 36%	 545 / 2976 18%
 1232 / 5594 22%	 516 / 3909 13%
 471 / 1612 29%	 3269 / 15464 21%

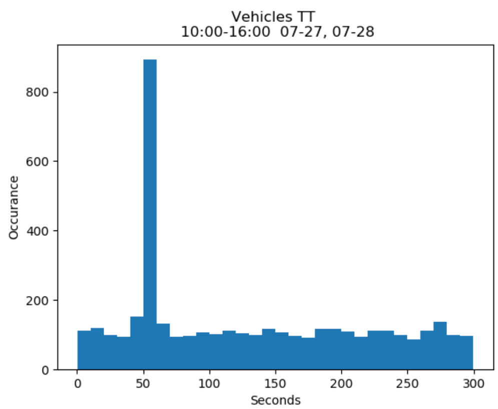
Table 6. TT Values of ILD Traffic Monitoring System During Weekend



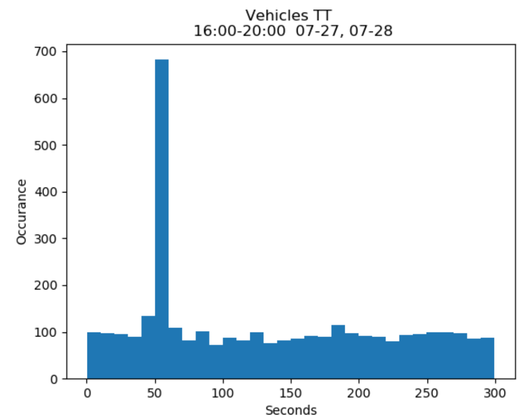
313 / 840
37%



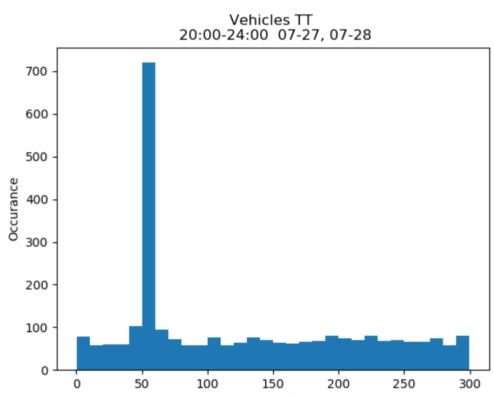
472 / 1701
27%



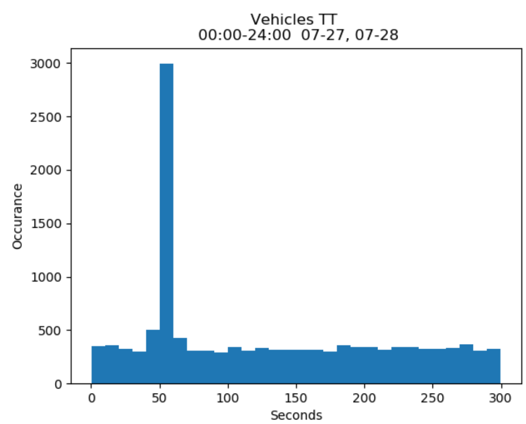
1026 / 4029
25%



794 / 3384
23%



815 / 2750
30%



3420 / 12704
27%

Travel Time Using Data Spike Detection Algorithm:

Correlated signatures contained a significant amount of noisy data, where 27% provided the correct TT and 73% experienced errors. Computing overall mean of correlated signatures resulted in a TT value longer than expected. To illuminate the noise, spike detection algorithms where applied. TT values were binned into 10-second groups, and a spike was selected for estimating TT given that half of the samples inside the spike were greater than number of samples in other binned groups. Subsequent to detecting spikes, TT can be estimated as the mean value of all samples within the spike.

Spike Detection: $\frac{Size_{group\ x}}{2} = Size_{\ group\ i}\ ; i\ \in [1, n]; i\ \neq x$

TT Estimation: $TT = mean\ (TT_{spike\ group})$

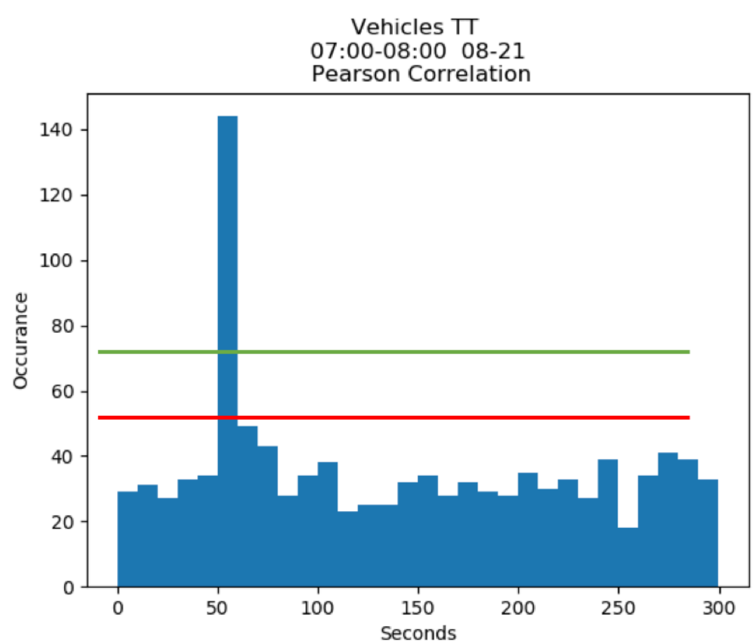


Figure 40. Travel Time data spike.

Figure 40 shows a sample of TT values for a single hour, where search window was 300 seconds. The data spike was considered to estimate TT value. Figure 41 illustrates estimated TT values for a single day where TT values were grouped into 20-minute segments. Total number of segments was 72. Unfortunately, the algorithm was not able to function well during morning rush hour (7:20 to 9:00 a.m.) and evening rush hour (15:20 to 18:00 p.m.). For 17 segments, the algorithm wasn't able to detect a data spike or estimate TT. Figure 42 shows examples for which the algorithm was unable to detect data spikes.

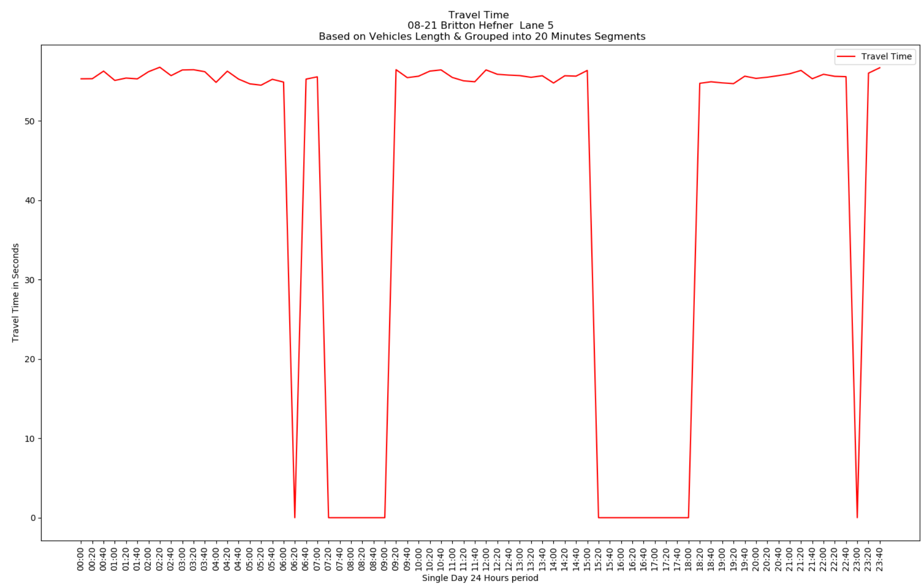


Figure 41. Estimated TT values for 1 mile distance.

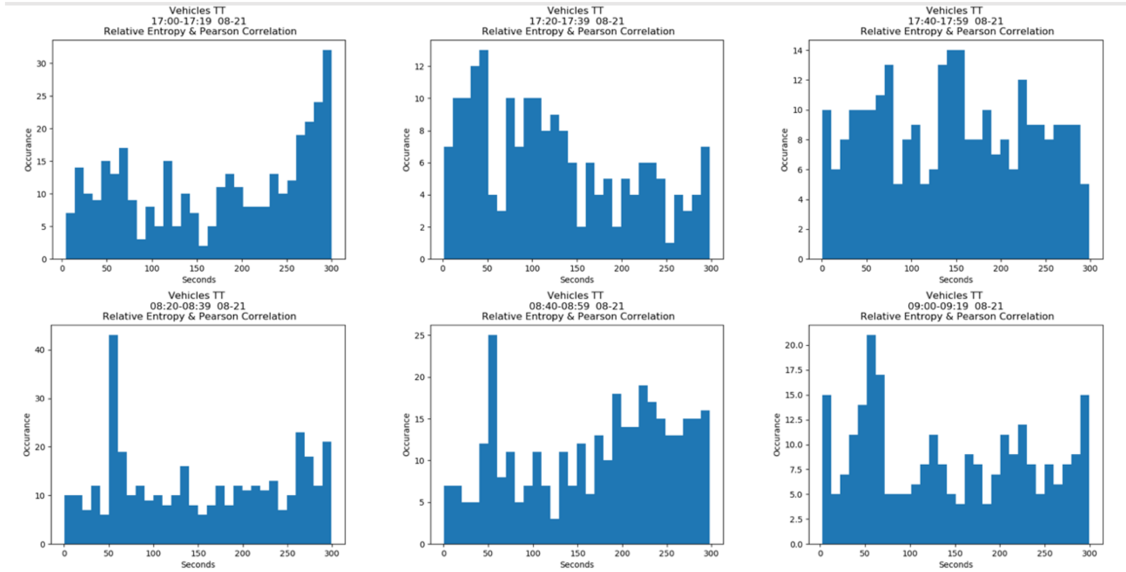


Figure 42. Data spikes during rush hour.

Several methods have been investigated to improve real-time TT accuracy (e.g., reducing correlation search window from 5 minutes to 2 and ½ minutes for 1 mile distance). The 5-minute search window proved more accurate than the shorter period, primarily because it aided in flattening error values. See Figure 43.

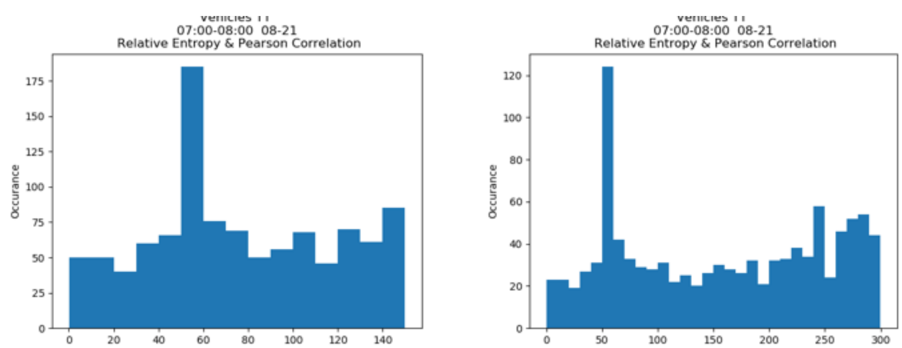


Figure 43. TT for a 2:30-minute vs. 5-minute search window.

Improving TT Accuracy Based on Vehicle Length

Longer vehicles have unique signatures, which provides a higher accuracy in signature correlation and TT estimation (See Figures [44, 45]). However, sedans are the most common vehicle type for the deployment site that was investigated. The majority of class 2 vehicles have the same signature shape; hence, this excessive number of this class of vehicles contributed to errors in estimating TT (See Figure 46). The optimized algorithm replaces missing TT values with estimates using long vehicles only. Figure 47 shows how number of segments missing TT estimation reduced from 17 to 13 after applying the length-based TT estimation enhancement.

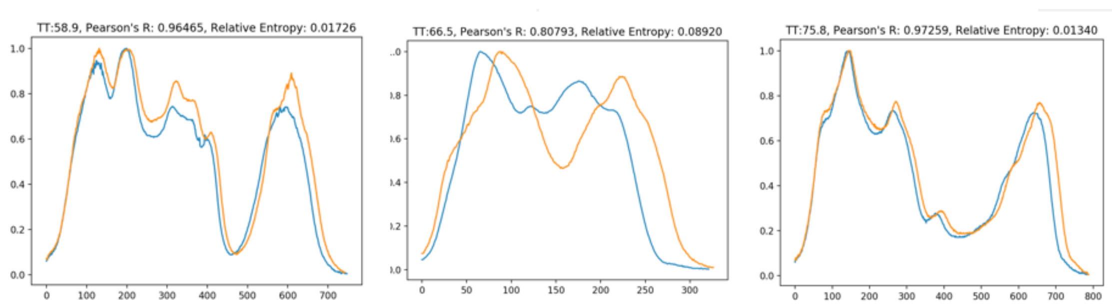


Figure 44. Correlated signatures for long vehicles.

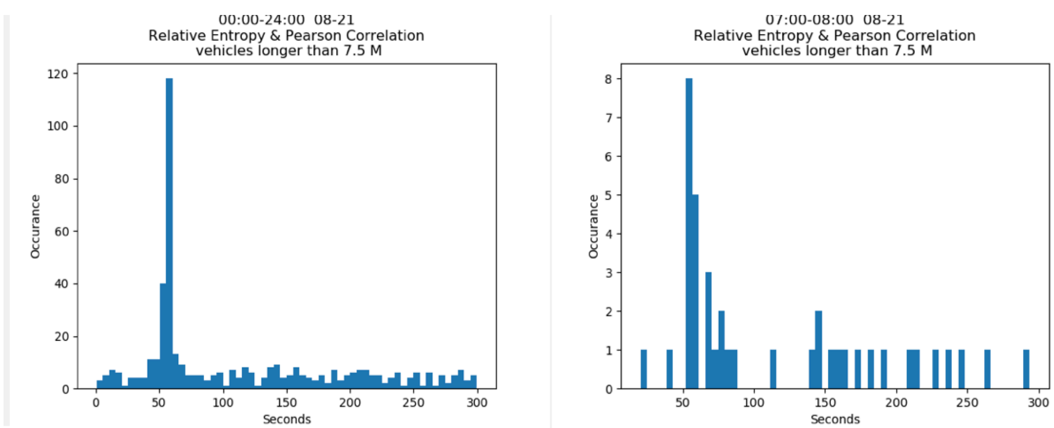


Figure 45. TT for long vehicles.

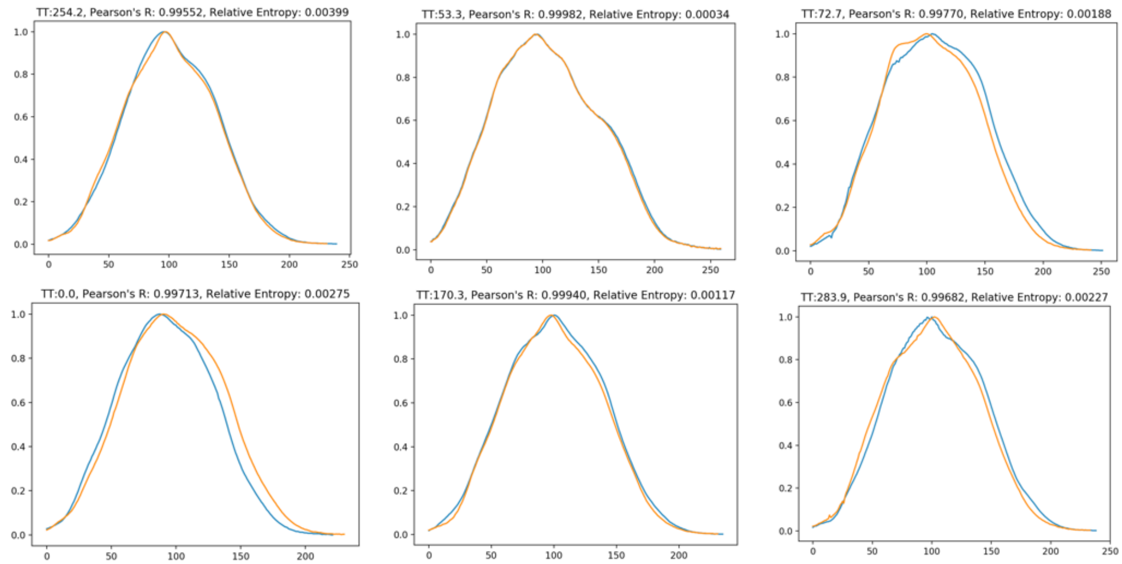


Figure 46. Correlated signatures for sedan vehicles.

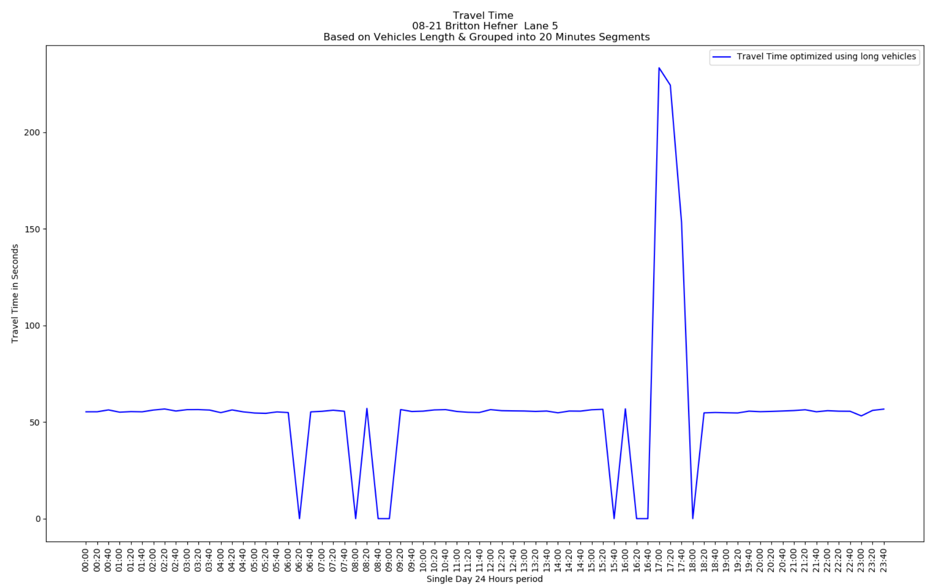


Figure 47. TT estimation using vehicles length enhancement.

Another enhancement to the algorithm aided in increasing system reliability. When searching for a data spike, the algorithm will compare neighboring data groups to the group with the maximum number of vehicles. If this neighboring group has number of vehicles more than half the number of the group with the maximum number of vehicles, both data groups will be combined as a single data spike, and then compared with all other groups to estimate TT based on average of all TT values inside this new data group. Figure 48 shows how this final optimization reduced the number of segments with missing TT from 13 to 4 out of 72 total number of segments. These optimizations made the system able to estimate TT up to 94% of the time and update TT estimation every 20 minutes.

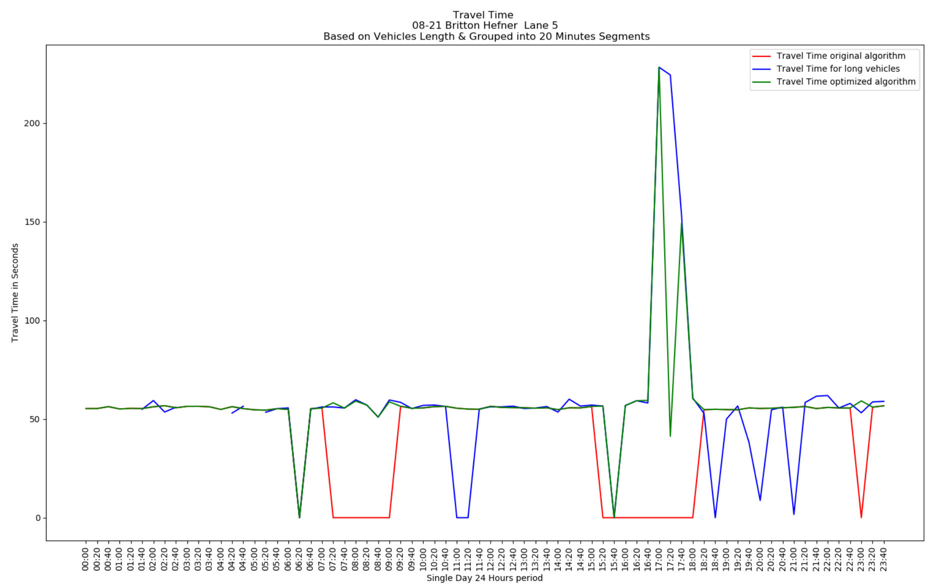
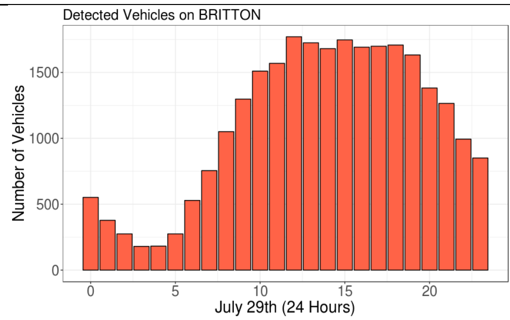
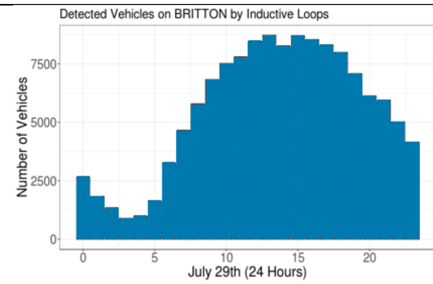
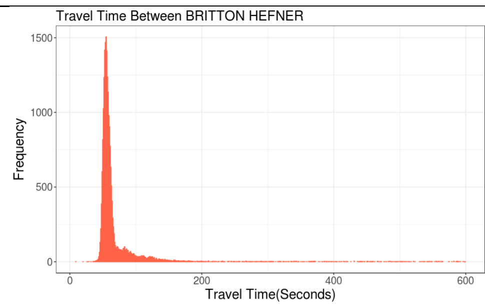
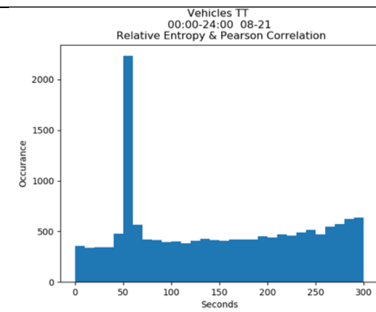


Figure 48. TT estimation using optimized spike detection algorithm.

Chapter 6 - Comparison between ILD and BT signatures

Table 7. Comparison Between ILD and BT Traffic Monitoring Systems

	Bluetooth System	**ILD System**
Number of detect vehicles		
Travel Time Estimation		

The following highlights the similarity and differences between TT calculations using BT sniffer versus ILD detectors.

- The ILD system demonstrated higher detection rate than the BT system, although both were characterized by the same distribution. Only 20% penetration was achieved using BT sniffer with external directional antenna.
- Both systems detect TT, permitting TT estimation. However, the ILD system experienced more noise—only 27% of the data within the correct TT. Thus, the ILD system requires advanced processing to filter out noise and estimate TT.

- Both systems are reliable for estimating TT. The BT system provides accurate TT estimation 99.9% of the time; while the ILD system provides correct TT estimation 94% of the time.
- The BT system provides a real-time TT estimations per site; while ILD-based system provides TT estimates per lane every 20 minutes.
- The ILD-based system provides additional traffic information: vehicle length, classification, and speed on a single site; while the BT system provides only TT.
- Many factors can affect the accuracy of ILD-based systems (e.g., vehicles changing lanes, vehicles changing speed while crossing the loops, congestion, distance between two sites, and other environmental factors causing electrical noise); while BT-based systems have fewer factors that affect performance (e.g., type of the antenna, distance between sniffer and the roadway).

Chapter 7 - Conclusion and Future Work

Conclusion

This thesis described the architecture of real-time IoT systems used in the Intelligent Transportation Systems (ITS). The systems improved traffic surveillance system performance, enabling traffic operators to detect rapid changes in traffic conditions and to provide commuters with information to help them avoid congestion and work zones.

The focus of the research was to leverage BT technology and ILD to re-identify vehicles for TT estimation. Both systems adequately estimated TT. Although BT sniffers proved to have lower traffic penetration, 95% of the calculated TT was accurate. Conversely, ILD had a higher detection rate, but TT estimation was prone to greater error. Only 27% of correlated signatures were able to estimate TT. TT based on BT is a straight forward method in which the transportation system searches for matching MAC addresses recorded at two checkpoints, and then reports TT of the re-identified vehicle MAC by subtracting detection time and computing the mean of TT values detected at both locations. BT systems can provide accurate TT measures 99.9% of the time.

ILD-based systems require a greater number of computational processes to normalize signatures and determine TT based on highly correlated signatures. A spike detection algorithm can be applied to filter out erroneous TT values and determine the mean of all TT values within the spike. ILD-based systems can detect a data spike and estimate TT only 94% of the time, as higher noise values occur during traffic congestion.

Both systems are characterized by advantages and disadvantages. While ILD-based systems are less accurate and require a longer period (e.g., 20 minutes) to report TT values, they provide rich information about road capacity, vehicle speed, and vehicle length. Such information is necessary for length-based vehicle classification. Although BT-based systems report more frequently and accurately TT estimates, they are unable to provide additional traffic information details.

Future work

Future work could combine BT sniffers and ILD for correlating both data to similar vehicles traveling on urban roads.

References

1. J. Rupnik, J. Davies, B. Fortuna, A. Duke and S. S. Clarke, "Travel Time Prediction on Highways," 2015 IEEE International Conference on Computer and Information Technology.

2. S. T. Jeng, Y. C. A. Tok and S. G. Ritchie, "Freeway Corridor Performance Measurement Based on Vehicle Reidentification," in IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 3, pp. 639-646, Sept. 2010.

3. S.-T. Jeng and L. Chu, "Vehicle reidentification with the inductive loop signature technology," J. of the Eastern Asia Society for Transp. Studies, vol. 10, 2013, pp. 1896-1915.

4. A. Bhaskar and E. Chung, “Fundamental understanding on the use of Bluetooth scanner as a complementary transport data,” Transp. Res. C, Emerging Technol., vol. 37, pp. 42–72, Dec 2013.

5. Q. He and L. Wang, "Reliability Research on Travel Time of Transportation Network," 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, 2010, pp. 1005-1007.

6. Hilmi Berk Celikoglu, “Flow-Based Freeway Travel-Time Estimation: A Comparative Evaluation Within Dynamic Path Loading” IEEE Transactions on Intelligent Transportation Systems (Volume: 14, Issue: 2, June 2013)

7. Danda B. Rawat, Danda B. Rawat , Danda B. Rawat “Towards Intelligent Transportation Cyber-Physical Systems: Real-Time Computing and Communications Perspectives” IEEE SoutheastCon 2015, April 9 - 12, 2015 - Fort Lauderdale, Florida

8. Arafat Al-Dweik, Radu Muresan, Matthew Mayhew, Mark Lieberman “IoT-Based Multifunctional Scalable Real-Time Enhanced Road Side Unit for Intelligent Transportation Systems ” 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)

9. C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey", IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 414-454, Jan. 2013.

10. C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things", Trans. Emerg. Telecommun. Technol., vol. 25, no. 1, pp. 81-93, 2014.

11. L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey", Comput. Netw., vol. 54, pp. 2787-2805, Oct. 2010

12. “FHWA Forecasts of Vehicle Miles Traveled (VMT): May 2014,” Office of Highway Policy Information

13. “WHO | Global status report on road safety 2015.”

14. “2015 Urban Mobility Scorecard and Appendices — Urban Mobility Information.”

15. R. Akçelik, “Travel time functions for transport planning purposes: Davidson’s function, its time dependent form and alternative travel time function,” Australian Road Res., vol. 21, no. 3, pp. 49–59, 1991.

16. K. Davidson, “A flow travel time relationship for use in transportation planning,” in Proc. 3rd ARRB Conf., Sydney, NSW, Australia, 1966, vol. 3, no. 1, pp. 183–194.

17. S. M. Turner, W. L. Eisele, R. J. Benz, and D. J. Holdener, “Travel time data collection Handbook,” Transp. Res. Board, Federal Highway Admin. (FHWA), Washington, DC, USA, Tech. Rep., 1998.

18. Pushkin Kachroo, Shankar Sastry “Travel Time Dynamics for Intelligent Transportation Systems: Theory and Applications” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 17, NO. 2, FEBRUARY 2016

19. CHARITH PERERA‚ CHI HAROLD LIU, SRIMAL JAYAWARDENA, AND MIN CHEN” A Survey on Internet of Things from Industrial Market Perspective” Received November 13, 2014, accepted December 7, 2014, date of publication January 8, 2015, date of current version January 26, 2015.

20. Sadao Takaba, Takeshi Morita, Takashi Hada, Tsutomu Usami and Morie Yamaguchi “Estimation and Measurement of Travel Time by Vehicle Detectors andLicense Plate Readers ” IEEE Vehicle Navigation and Information Systems Conference, 1991, Troy, MI, USA, USA

21. L. Dlagnekov, Video-based Car Surveillance: License Plate, Make, and Model Recognition, Masters Thesis, U.C. San Diego, 2005.

22. C. Anagnostopoulos, I. Anagnostopoulos, I. Psoroulas, E. Kayafas, V. Loumos, "License Plate Recognition from Still Images and Video Sequences: A Survey", IEEE Trans. on Intelligent Transportation Systems, vol. 9, No. 3, pp.377-391, 2008.

23. Aude Hofleitner, Ryan Herring, Pieter Abbeel, and Alexandre Bayen "Learning the Dynamics of Arterial Traffic From Probe Data Using a Dynamic Bayesian Network " IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 13, NO. 4, DECEMBER 2012

24. Wang Wei, Guo Xiucheng , Jin Jing, Ran Bin "GPS Probe Based Freeway Real-Time Travel Speed Estimation using Kalman Filter"2010 International Conference on Intelligent System Design and Engineering Application

25. Charith D. Chitraranjan, Amal S. Perera, and Anne M. Denton "Tracking Vehicle Trajectories by Local Dynamic Time Warping of Mobile Phone Signal Strengths and its Potential in Travel-time Estimation" The Fourth IEEE International Workshop on the Impact of Human Mobility in Pervasive Systems and Applications, 2015

26. ETSI, Reccommendation GSM 05.08: Radio Sub-system Link Control. European Telecommunications Standards Institute, 1995.

27. Kristina Taneva , Danco Davcev "Traffic jam detection and control by crowdsourcing and a social network"2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)

28. Daehan Kwak, Daeyoung Kim, Ruilin Liu, Liviu Iftode, Badri Nath, "Tweeting traffic image reports on the road" IEEE 6th International Conference on Mobile Computing, Applications and Services, 2014 Austin, TX, USA

29. Karoly Farkas ; Gabor Feher ; Andras Benczur ; Csaba Sidlo"Crowdsending Based Public Transport Information Service in Smart Cities" IEEE Communications Magazine (Volume: 53, Issue: 8, August 2015)

30. Dominik Zapletal, Adam Herout "Vehicle Re-Identification for Automatic Video Traffic Surveillance" 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops

31. Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma "PROVID: Progressive and Multi-modal Vehicle Re-identification for Large-scale Urban Surveillance" Citation information: DOI 10.1109/TMM.2017.2751966, IEEE Transactions on Multimedia

32. J.C.Haartsen,"The Bluetooth radio system,"IEEEPers.Commun.Mag., vol. 7, no. 1, pp. 28–36, Feb. 2000.

33. "Automotive market rapidly expanding beyond hands-free calling," Blue- tooth Special Interest Group (SIG), Kirkland, WA, USA, 2014.

34. “Mobile phone use: A growing problem of driver distraction,” World Health Org., Geneva, Switzerland, 2011.

35. Juan José Vinagre Díaz, Ana Belén Rodríguez González, and Mark Richard Wilby “Bluetooth Traffic Monitoring Systems for Travel Time Estimation on Freeways” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 17, NO. 1, JANUARY 2016

36. Ashish Bhaskar, Ming Qu, and Edward Chung “Bluetooth Vehicle Trajectory by Fusing Bluetooth and Loops: Motorway Travel Time Statistics” IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 16, NO. 1, FEBRUARY 2015

37. Isaac K Isukapati, George F List “Synthesizing Route Travel Time Distributions Considering Spatial Dependencies” 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

38. J. S. Wasson, J. R. Sturdevant, and D. M. Bullock, “Real-time travel time estimates using media access control address matching,” Inst. Transp. Eng. J., vol. 78, no. 6, pp. 20–23, Jun. 2008.

39. Gabriel Michau, Alfredo Nantes, Ashish Bhaskar, Edward Chung, Patrice Abry, and Pierre Borgnat “Bluetooth Data in an Urban Context: Retrieving Vehicle Trajectories” IEEE Transactions on Intelligent Transportation Systems (Volume: 18, Issue: 9, Sept. 2017)

40. Janusz Gajda, Piotr Burnos “Identification of the Spatial Impulse Response of Inductive Loop Detectors” Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International

41. Traffic Detector Handbook: Third Edition—Volume I October 2006

42. Shin-Ting (Cindy) Jeng and Lianyu Chu “A High-Definition Traffic Performance Monitoring System with the Inductive Loop Detector Signature Technology” 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC) October 8-11, 2014. Qingdao, China

43. C.Oh,A.Tok,andS.G.Ritchie,“Real-timefreewaylevelofserviceusing inductive-signature-based vehicle reidentification system,” IEEE Trans. Intell. Transp. Syst., vol. 6, no. 2, pp. 138–146, Jun. 2005.

44. R. D. Kühne and S. Immes, “Freeway control systems for using section- related traffic variable detection,” in Proc. Pacific Rim TransTech Conf., Seattle, WA, 1993, pp. 56–62.

45. Soner Meta ; Muhammed G. Cinsdikici “Vehicle-Classification Algorithm

Based on Component Analysis for Single-Loop Inductive Detector" IEEE Transactions on Vehicular Technology (Volume: 59, Issue: 6, July 2010)

46. Herivelton A. Oliveira ; Fabio R. Barbosa ; Otacilio M. Almeida ; Arthur P. S. Braga "A vehicle classification based on inductive loop detectors using artificial neural networks" 2010 9th IEEE/IAS International Conference on Industry Applications - INDUSCON 2010

47. Anahita Jamshidnejad ; Bart De Schutter "An iterative procedure for estimating the generalized average speed using microscopic point measurements" 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)

48. Mandoye Ndoye, Virgil F. Totten, James V. Krogmeier, Member, IEEE, and Darcy M. Bullock "Sensing and Signal Processing for Vehicle Reidentification and Travel Time Estimation" IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 12, NO. 1, MARCH 2011

49. Ashton, K.: That 'internet of things' thing, in the real world things matter more than ideas. RFID J. (2009). http://www.rfidjournal.com/articles/view?4986

50. Giancarlo Fortino • Paolo Trunfio "Internet of Things Based on Smart Objects: Technology, Middleware and Applications" Springer Publishing Company, Incorporated ©2014

51. Jie Lin ; Wei Yu ; Nan Zhang ; Xinyu Yang ; Hanlin Zhang ; Wei Zhao "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications" IEEE Internet of Things Journal (Volume: 4, Issue: 5, Oct. 2017)

52. Jason R.C. Nurse ; Sadie Creese ; David De Roure "Security Risk Assessment in Internet of Things Systems" IEEE IT Professional (Volume: 19, Issue: 5, 2017)

53. http://beagleboard.org/bone

54. BeagleBone Black System Reference Manual https://cdnshop.adafruit.com/datasheets/BBB_SRM.pdf

55. systemd System and Service Manager https://www.freedesktop.org/wiki/Software/systemd/

56. MySQL Rational Database Management System https://www.mysql.com

57. NodeJS Javascript runtime https://nodejs.org/en/

58. RFC 6455 - The WebSocket Protocol https://tools.ietf.org/html/rfc6455

59. RFC 7158 - The JavaScript Object Notation (JSON) Data Interchange Format https://tools.ietf.org/html/rfc7159
60. Ubertooth One http://ubertooth.sourceforge.net

61. Bruce A. Harvey “Long-Term Field Study of Lightning Surges Through Traffic Monitoring In-Pavement Sensors” IEEE Transactions on Industry Applications (Volume: 51, Issue: 4, July-Aug. 2015)
62. Malik Zawwar Hussain ; Misbah Irshad ; Muhammad Sarfraz ; Nousheen Zafar “Interpolation of Discrete Time Signals Using Cubic Spline Function” 2015 19th International Conference on Information Visualization.