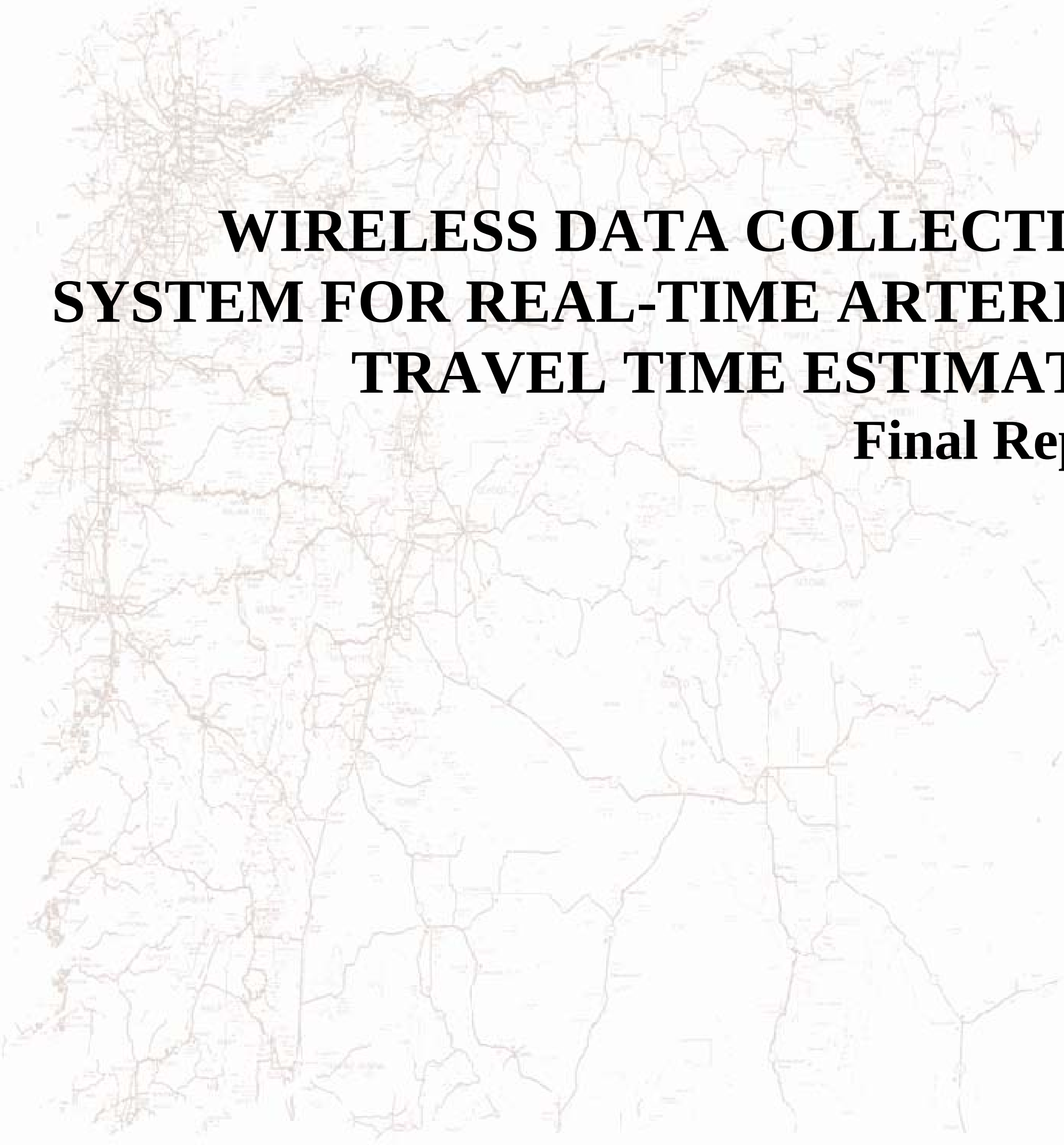
RESEARCH



WIRELESS DATA COLLECTION SYSTEM FOR REAL-TIME ARTERIAL TRAVEL TIME ESTIMATES

Final Report

RESEARCH ODOT/OTREC

This publication is a result of joint funding by
Oregon Department of Transportation
(ODOT) and Oregon Transportation Research
and Education Consortium (OTREC)

WIRELESS DATA COLLECTION SYSTEM FOR REAL-TIME ARTERIAL TRAVEL TIME ESTIMATES

RS 500-410
OTREC-RR-10-16

by

J. David Porter
David S. Kim
Mario E. Magaña
Oregon State University

for

Oregon Department of Transportation
Research Section
200 Hawthorne Avenue SE, Suite B-240
Salem OR 97301-5192

and

Oregon Transportation Research
and Education Consortium (OTREC)
P.O. Box 751
Portland, OR 97207

March 2011

1. Report No.

OR-RD-11-10
OTREC 10-16

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Wireless Data Collection System for Real-Time Arterial Travel Time Estimates

5. Report Date

March 2011

6. Performing Organization Code

7. Author(s)

J. David Porter and David S. Kim
School of Mechanical and Industrial Engineering
204 Rogers Hall
Oregon State University, Corvallis, OR 97331

Mario E. Magaña
School of Electrical Engineering and Computer Science
1148 Kelley Engineering Center
Oregon State University, Corvallis, OR 97331

8. Performing Organization Report No.

9. Performing Organization Name and Address

Oregon State University
Corvallis, OR 97331

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

OTREC 10-16 / SPR 500-410

12. Sponsoring Agency Name and Address

Oregon Department of Transportation
Research Section
200 Hawthorne Ave. SE, Suite B-240
Salem, Oregon 97301-5192

Oregon Transportation Research
& Education Consortium (OTREC)
P.O. Box 751
Portland, Oregon 97207

13. Type of Report and Period Covered

Final Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This project pursued several objectives conducive to the implementation and testing of a Bluetooth (BT) based system to collect travel time data, including the deployment of a BT-based travel time data collection system to perform comprehensive testing on all the components. Two different BT-based travel time data collection systems were installed. The first system, composed of two DCUs, was installed on a corridor located in Salem, OR. Extensive testing was done on this system, including the collection of travel time samples. A second system composed of five DCUs was installed along 99W in the city of Tigard, OR. Very limited data collection was done on 99W due to the lack of network connectivity.

Six different antenna types were characterized using the two DCU BT-based travel time data collection system. The result of the antenna characterization tests showed that vertically polarized antennas with gains between 9 and 12 dBi are good candidates to support a BT-based travel time data collection system. Antennas with circular polarization do not seem to improve the performance, despite the lack of control regarding the orientation of BT-enabled devices in most applications. Travel time samples were also collected with this system. The results indicate that a trade-off exist between the number of samples obtained and the accuracy of these travel time samples. This trade-off is most likely the result of differences in road coverage areas provided by the different antenna types.

17. Key Words

Antenna characterization, Bluetooth, Bluetooth-based travel time estimation

18. Distribution Statement

Copies available online at http://www.oregon.gov//ODOT/TD/TP_RES/ or www.otrec.us

19. Security Classification (of this report)

Unclassified

20. Security Classification (of this page)

Unclassified

21. No. of Pages

147

22. Price

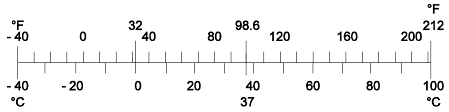
SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	millimeters squared	mm^2
ft^2	square feet	0.093	meters squared	m^2
yd^2	square yards	0.836	meters squared	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	kilometers squared	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	meters cubed	m^3
yd^3	cubic yards	0.765	meters cubed	m^3
NOTE: Volumes greater than 1000 L shall be shown in m^3.				
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams	Mg
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5(F-32)/9	Celsius temperature	°C

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	millimeters squared	0.0016	square inches	in^2
m^2	meters squared	10.764	square feet	ft^2
ha	hectares	2.47	acres	ac
km^2	kilometers squared	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	meters cubed	35.315	cubic feet	ft^3
m^3	meters cubed	1.308	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.205	pounds	lb
Mg	megagrams	1.102	short tons (2000 lb)	T
		TEMPERATURE (exact)		
°C	Celsius temperature	1.8 + 32	Fahrenheit	°F



***** SI is the symbol for the International System of Measurement

(4-7-94 jbp)

ACKNOWLEDGEMENTS

We would like to thank the Oregon Transportation Research and Education Consortium (OTREC) and the Oregon Department of Transportation (ODOT) Research Section for funding this work. We would also like to thank Galen McGill from the ODOT ITS Unit, June Ross from the ODOT Research Section, and the members of the Technical Advisory Committee for their guidance and support in this effort. Special thanks go out to Adam Bradford, Tony Owens and Gary Binning for assisting the project team in setting the antennas at the test site and for their invaluable technical support.

DISCLAIMER

The contents of this report reflect the views of the authors, who are solely responsible for the facts and the accuracy of the material and information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation University Transportation Centers Program and Oregon Department of Transportation in the interest of information exchange. The U.S. Government and the Oregon Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views of the U.S. Government or Oregon Department of Transportation. This report does not constitute a standard, specification, or regulation.

WIRELESS DATA COLLECTION SYSTEM FOR REAL-TIME ARTERIAL TRAVEL TIME ESTIMATES

TABLE OF CONTENTS

LIST OF TABLES

LIST OF PHOTOS/FIGURES

EXECUTIVE SUMMARY

This project has evolved from a new method for travel time estimation based on the collection of time-stamped media access control (MAC) addresses from Bluetooth-enabled devices (e.g., cell phones, personal digital assistants, GPS-based navigation systems, etc.). MAC addresses are unique to a particular Bluetooth device and are most commonly represented with a 48-bit hexadecimal (i.e., base 16) number, which consists of 12 characters. They are arranged in six pairs, each separated by a colon (e.g., 00:10:B5:C4:99:6A).

This project pursued several objectives conducive to the implementation and testing of a Bluetooth (BT) based system to collect travel time data. The first objective was to make improvements to an existing BT-based data collection unit (DCU) developed on an earlier project funded by ODOT's ITS Unit. The second objective was to deploy a BT-based travel time data collection system to perform comprehensive testing on all the components. The third objective was to develop and test software applications to collect, process, and store time-stamped MAC addresses. Finally, the fourth objective involved the development of functional requirements and technical specifications for the DCU and the preparation of a user's manual with instructions on how to assemble, configure, and troubleshoot the DCU.

The hardware platform of the original DCU was very reliable, so the focus in this area was to improve the efficiency of the Linux script that collects time-stamped media access control (MAC) addresses from BT-enabled devices in vehicles. The improvements made included the use of a more efficient command to collect MAC addresses, the development of an algorithm to check for duplicate MAC addresses, and the removal of unnecessary lines and variables in the code. Additionally, procedures for the collection, processing and storage of time-stamped MAC addresses were specifically defined to protect citizens' privacy.

Two different BT-based travel time data collection systems were installed. The first system, composed of two DCUs, was installed on a corridor located in Salem, OR. Extensive testing was done on this system, including the collection of travel time samples. A second system composed of five DCUs was installed along 99W in the city of Tigard, OR. Very limited data collection was done on 99W due to the lack of network connectivity.

A good match between the DCU and antenna is critical in a BT-based travel time data collection system. Since the orientation of the BT-enabled devices in vehicles cannot be controlled, intuition may dictate that antennas with circular polarization would perform better. However, the results of the antenna characterization tests revealed that vertically polarized antennas with gains between 9 and 12 dBi have very good performance. Antennas with circular polarization did not seem to improve performance, despite the lack of control regarding the orientation of BT-enabled devices in most applications.

The two-DCU system was used to collect travel time samples. The results indicated that the 180 degree antenna is clearly the best performing antenna with respect to the generating travel time samples. The Yagi also performed well, but showed inconsistencies over time periods which are not well understood. While a high sampling rate value is desirable, the accuracy of the collected travel time samples is perhaps a more critical performance measure. The most accurate method for computing travel time samples was to utilize the difference in *average* group times.

Three server-based applications were developed in collaboration with ODOT's ITS Unit. The first application collects a text file from DCU containing individual records of time-stamped MAC addresses collected from BT-enabled devices in vehicles. The second application stores each record in the text file into a Microsoft© Access database. The Microsoft© Access database was the third software component developed. These software components were tested on a limited basis during the antenna characterization tests conducted in Salem, OR.

A new phase of this project in underway and will focus on issues related to the processing and synthesis of data collected from Bluetooth-enabled devices to generate travel time performance measures. A number of issues related to data processing and synthesis need to be addressed, including:

- How data collected by the DCUs should be filtered. Some MAC addresses are read a large number of times on specific days indicating that most of these records may not be used for travel time calculations. Filtering this data before computing travel times will increase computational efficiency.
- Specific procedures for identifying travel time sample data outliers and non-vehicle related data. Under very congested conditions, very slow travel times must be distinguished from non-vehicle travel times.
- The amount, format, and length of time MAC-based address data are stored.
- Development of methods to estimate the precision of travel time performance measures.
- Utilization of MAC-based data to estimate other traffic performance measures at intersections and on highways (e.g., traffic volume).
- Understanding the impact of average vehicle speeds and different traffic states (stationary vs. moving) near an intersection or on a highway, on the MAC addresses recognized by a reader.
- Understanding the effect that antenna and reader design variables have on the precision and accuracy of travel time estimates.
- Evaluation of the impact of pushing data from readers to a central server vs. pulling data from the readers. Examine if pushing data provides significant benefits with respect to providing "real time" travel time information.
- Assessment of the information flow volume assuming a wide-scale implementation of Bluetooth readers.

1.0 INTRODUCTION

In this report, the findings and results of the project "Wireless Data Collection System for Real-Time Arterial Travel Time Estimates" are presented. This project has evolved from a new method for travel time estimation based on the collection of time-stamped media access control (MAC) addresses from Bluetooth-enabled devices (e.g., cell phones, personal digital assistants, GPS-based navigation systems, etc.). MAC addresses are unique to a particular Bluetooth device and are most commonly represented with a 48-bit hexadecimal (i.e., base 16) number, which consists of 12 characters. They are arranged in six pairs, each separated by a colon (e.g., 00:10:B5:C4:99:6A).

The project documented in this report expands on an earlier ODOT ITS Unit-funded project that focused on the initial development of inexpensive data collection units (DCUs) (Porter et al., 2009). Each DCU platform developed in that project utilized different sets of commercially available electronic boards, antennae, power supplies, and enclosures. The objectives of the current project, as stated in the project work plan, were to:

1. Improve the environmental durability of the existing prototype DCUs by applying conformal coating and/or utilizing different enclosures.
2. Develop and test different antennae designs in order to optimize the performance of the DCUs.
3. Conduct comprehensive field testing of the prototype DCUs in various locations representing a variety of conditions.
4. Develop centralized travel time estimation procedures for processing data obtained from DCUs installed at various locations.
5. Develop procedures for data collection, data processing and data storage to protect citizens' privacy.

To accomplish these objectives, a set of research tasks was defined. These tasks were jointly defined and agreed to by the research team and the project Technical Advisory Committee (TAC). The following research tasks were included:

Task # 1: Literature review and communication with other DOTs.
Task # 2: Define potential system architectures and installation configurations.
Task # 3: Development of procedures for data collection, data processing and data storage to protect citizens' privacy.
Task # 4: Modifications and enhancements of the DCU platforms.
Task # 5: Comprehensive performance evaluation of the travel time data collection system.
Task # 6: Travel time data collection and analysis.
Task # 7: Update the functional and technical requirements document and development of a users' manual.
Task # 8: Documentation of system components and preparation of the final report.

These tasks were defined as a research *plan* and therefore there were deviations with respect to their original scope as the research efforts progressed, additional information and/or constraints were discovered. Deviations were approved by the TAC and are described in the section documenting the particular task.

1.1 TERMINOLOGY, DEFINITIONS, AND ACRONYMS

Table 1.1 defines various terms and concepts that are used throughout this document; Table 1.2 defines important acronyms.

Table 1.1: Relevant terms and concepts used in this document

TERMINOLOGY	DEFINITION
Bluetooth Wireless Technology	Bluetooth wireless technology is a wireless communication link, operating in the unlicensed Industrial, Scientific and Medical (ISM) band at 2.4 GHz using a frequency hopping transceiver. It allows real-time audio and video (A/V) and data communications between Bluetooth enabled hosts.
Data Collection Unit	A data collection unit (DCU) is the component of the travel time data collection system responsible for collecting time-stamped MAC addresses from Bluetooth-enabled devices.
Commercial Off-The-Shelf	Technology with specific functionality that exists as a current commercial product.
Functional Requirement	A function that a particular system component in a technology configuration must execute to successfully complete a task or a portion of a task.
Media Access Control Address	Media Access Control (MAC) addresses are unique identifiers for every node on a local area network (LAN) or other network and are most commonly represented with a 48-bit hexadecimal (i.e., base 16) number, which consists of 12 characters. They are arranged in six pairs, each separated by a colon (e.g., `00:10:B5:C4:99:6A`).
Technical Specification	Specific metrics and values that precisely describe how a functional requirement must be performed.

Table 1.2: Relevant acronyms

ACRONYM	DEFINITION
COTS	Commercial-Off-Shelf
CSV	Comma Separated Value
DBMS	Database Management System
LAN	Local Area Network
DCU	Data Collection Unit
MAC	Media Access Control
RF	Radio Frequency

1.2 ORGANIZATION OF THE DOCUMENT

Sections 2.0, 3.0, and 4.0 of this report present the results and findings of research task #1. Section 2.0 presents an overview of travel time data collection methods. Section 3.0 discusses privacy issues related to automatic travel time data collection methods. Section 4.0 summarizes the information gathered via interviews with the different DOTs currently working on travel time data collection using Bluetooth technology.

Sections 5.0 and 6.0 present the results of research task #2. Section 5.0 includes a discussion on several potential system architecture options and installation configurations, whereas section 6.0 describes the procedures defined for data collection, data processing and data storage to protect citizen's privacy.

Section 7.0 presents the results of research tasks #5 and #6. First, a description of the Bluetooth (BT) based travel time data collection system that was tested is presented. Subsection 7.2 presents the results of characterizing six different antenna types to assess their suitability to support a BT-based travel time data collection system. Subsection 7.3 presents the results of the collection and analysis of travel time samples. Finally, Section 8.0 presents the conclusions of this study.

Appendix A and Appendix B include the deliverables of research task #7. Appendix A includes the updated version of the functional requirements and technical specifications. Appendix B is a user's manual covering the different aspects related to the assembling, configuration and troubleshooting of the DCU.

2.0 OVERVIEW OF TRAVEL TIME DATA COLLECTION METHODS

In this section, a review of existing methods and on-going developments in the area of automated travel time data collection (ATTDC) is presented. This review demonstrates where travel time estimation based on MAC address identification from Bluetooth-enabled devices falls within the general spectrum of ATTDC methods. ATTDC methods can be classified into three main categories:

- Indirect methods
- Vehicle re-identification methods; and
- Vehicle tracking methods (using GPS technology).

Indirect ATTDC methods estimate travel times from other collected data, and do not directly collect travel times for individual vehicles. Vehicle re-identification techniques collect individual vehicle travel times by identifying a “signature” and time associated with a vehicle at a specific location (e.g., point 1), and then re-identifying the same signature at another location and time that is a known distance from point 1. Vehicle re-identification methods differ mainly in the signature used to identify a vehicle. GPS-based vehicle tracking systems are used to obtain individual vehicle travel times from a subset of the location and time data collected as a vehicle travels.

In general, as one moves from indirect travel time estimation methods to vehicle tracking methods, the level of detail available through collected data increases as do concerns about privacy invasion. Costs of the various ATTDC methodologies, however, do not vary in the same way. For example, some indirect methods are more costly despite the fact that they provide less detailed travel time information. The different ATTDC methodologies reviewed in this section are:

- **Indirect methods**
  - Loop detectors

- **Vehicle re-identification methods**
  - Magnetic signatures
  - Inductive signature sensors
  - Laser-based scanners
  - License plate recognition systems
  - Radio Frequency Identification (RFID)

- **Vehicle tracking**
  - GPS tracking systems
  - Crowd sourcing

2.1 LOOP DETECTORS

The predominant ATTDC method currently in use throughout the United States is inductive loop detectors. An inductive loop detector is a large coil of electrically charged wire embedded beneath the pavement surface. Large metal objects (e.g., vehicles) passing over the loop detector affect the coil's electrical inductance causing a change in the flow of electricity running through it and triggering the detector (Gibson, 2009). Inductive loop detectors are deployed as single detectors (i.e., one loop per lane), or as dual-loop detectors formed by two consecutive single-loop detectors placed a set distance apart. Single-loop detectors are used to measure volume and lane occupancy; dual-loop detectors are also capable of measuring speed and vehicle length (Zhang et al., 2005).

Since travel times are estimated from loop detector data and this data is not individual vehicle travel time data, this ATTDC method is considered an indirect method of travel time estimation. There has been a considerable amount of past research that has focused on estimating travel times from inductive loop detector data (Angelo et al., 1999; Palen et al., 2000; Chen et al., 2001; Bremmer et al., 2004; Coifman, 2007). Many of these studies have noted that estimated travel times from single loop data are flawed because the estimated speed is based on the assumption of a common vehicle length. Despite the fact that double loop detectors do a better job at estimating speeds on road segments, the travel times estimated from these data may also be flawed under congested traffic conditions, which is when accurate travel time estimation is most important (Oh et al., 2002).

Loop detector-based travel time estimation is mostly used on freeways. However, there has been research on its application on signalized roads. In Nikolas & Alexander (2006), an analytical model was developed and tested to estimate travel times on signalized arterials in real time using flow and occupancy data from loop detectors, and signal information (cycle length, green times and offsets). This method was tested (via a simulation) on a section of road in Los Angeles with seven signalized intersections and resulted in moderate accuracy.

Loop detectors are already present in many roads and highways (many used for signal control), which would reduce wide-scale implementation costs. The data collected cannot be associated with any specific vehicles so there are no privacy invasion concerns. The main drawback is the accuracy of the travel times estimates generated.

2.2 MAGNETIC SIGNATURES

Travel time estimation using magnetic sensors is a vehicle re-identification method based on a technology patented by the Sensys Corporation. Sensys's system employs sensors referred to as *magnetometers* that generate a magnetic signature of an automobile as it drives over the sensor. With sensors located at two different points on a road, signatures are matched and the time the signatures were generated can be used to estimate the travel time of a specific vehicle (Kwong & Kavaler, 2009).

In this ATTDC system, each lane in a road must have a sensor and each sensor has seven nodes (each with its own microprocessor, antenna, transceiver, memory and battery). The magnetic field is distorted as a vehicle drives over a node and each node records a measurement of the magnetic field. Theses seven measurements make up a vehicle signature. The sensors transmit the time when the vehicle drove over them along with the magnetic signature to an access point on the side of the road via radio frequency. The access point then transmits this data to a server through a cellular service.

Matching magnetic signatures read at two different points is accomplished by identifying signatures that differ the least. The matching algorithm assigns each pair of signatures a distance measure calculated from the signature differences. The smaller this distance, the more likely both signatures correspond to the same vehicle. Unmatched vehicles indicate a vehicle has turned off the road.

Like loop detectors, this ATTDC method requires that sensors be installed in the road, which would be very costly and disruptive to implement on a wide-scale basis. Accuracy of travel time estimates from testing are good, but have not been compared to other vehicle re-identification methods. Privacy concerns related to the collection of magnetic signatures are minimal.

2.3 INDUCTIVE SIGNATURE SENSORS

In Andre et al. (2008) a vehicle re-identification method for the collection of travel times using loop detector inductive signatures is presented. In this method, each detector station consists of a single round or double square conventional inductive loop embedded in each lane of the freeway. The inductive loops are connected to advanced loop detector cards located in field units. These detector cards are then connected via USB interface to a field PC housed within the field unit. These detector cards process inductance signals generated by vehicles driving over the loops to generate vehicle signatures. Field computers are equipped with a cellular wireless modem for real time communication and data transfer with a centrally located server.

This ATTDC method can utilize existing loop detectors supplemented with additional hardware and computing power to process data and generate inductive signatures. It is similar in concept to the use of magnetic signatures.

2.4 LASER-BASED SCANNERS

In this vehicle re-identification ATTDC method, vehicle length and/or height are used as a vehicle signature, and are measured using laser-based vehicle detectors. A system for measuring vehicle length can be configured as depicted in Figure 2.1.

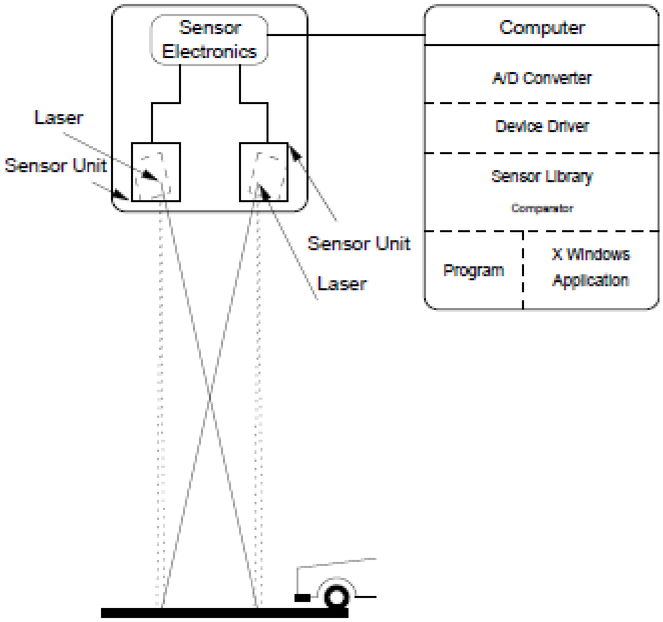


Figure 2.1: Laser based scanner system for measuring vehicle length (Cheng 2001)

The system consists of a laser and a spatially offset photo detector positioned above the plane of detection. The detector consists of imaging optics and a linear photodiode array. The offset photodiode array receives the laser light that is reflected back from the region of detection. The signal from the photodiode is amplified and sent to a computer for processing.

In Cheng (2001), a field prototype was developed and tested; however, no performance data is presented. The working prototype is shown in Figure 2.2.

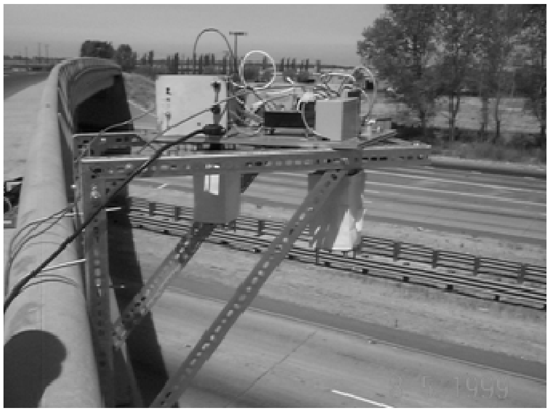


Figure 2.2: Laser system prototype (Cheng 2001)

2.5 LICENSE PLATE RECOGNITION SYSTEMS

License plate recognition systems can collect individual vehicle travel times between two points by taking and processing pictures of vehicles' license plates. Since this vehicle re-identification method uses license plate information as a signature, it can also provide other transportation data associated with a license plate such as vehicle classification. A license plate recognition system consists of a camera to take pictures and software to extract license plate information from the pictures. Figure 2.3 shows camera hardware installed over a particular point on a highway.

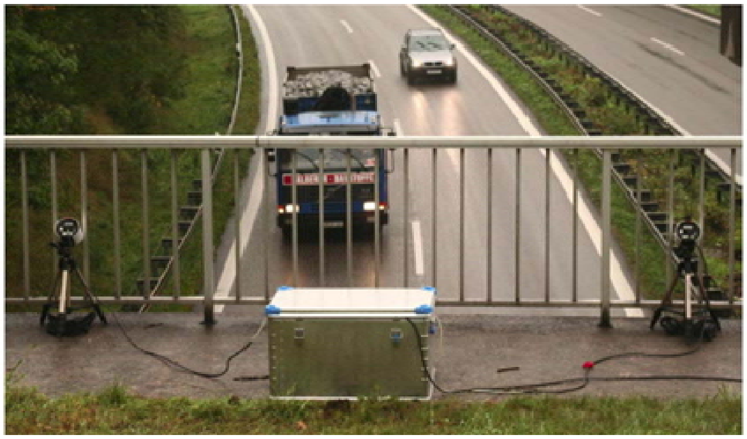


Figure 2.3: License plate recognition system hardware (Friedrich et al., 2008)

The camera hardware consists of an infrared camera, an optical color-detecting camera, and an array of LEDs that emit infrared light. The LED array directs beams of infrared light in the direction of the infrared camera, which then captures the light reflected by the white background of the license plates. Numbers appear white on the image and non-reflecting color characters and everything else appear black. Images are sent every 300 milliseconds to a computer. Software on the computer identifies the license plate number from the picture using an optical character recognition algorithm. The license plate number is then saved in a database with a timestamp and color image. The overall process is shown in Figure 2.4.

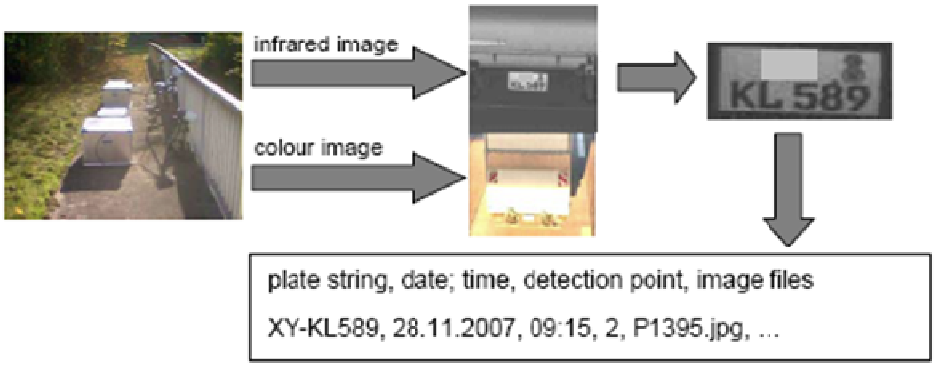


plate string, date; time, detection point, image files
XY-KL589, 28.11.2007, 09:15, 2, P1395.jpg, ...

Figure 2.4: License plate recognition system data processing (Friedrich et al., 2008)

The performance of license plate recognition systems depends on numerous factors:

- Uncontrollable factors
  - Precipitation
  - Angle and intensity of the sun
  - Shading on the pavement
  - Dirty or deformed plates
  - Vehicles changing lanes when picture is being taken
- Controllable factors
  - Angle of the camera relative to the horizontal
  - Distance between the camera and number plates

License plate recognition systems can be quite expensive and can generate large volumes of data. The accuracy of travel times is good, but privacy concerns with this method exist since the collected license plate information can be associated with specific vehicles and vehicle owners.

2.6 RADIO FREQUENCY IDENTIFICATION (RFID)

RFID is currently being employed in many electronic toll collection (ETC) systems in the United States. For example, in the San Francisco Bay area the FastTrak ETC system is operating on seven state-operated toll bridges and the Golden Gate Bridge. Because this technology can be used to identify a vehicle at a specific point while it is still moving, it can also be used to collect travel times for individual vehicles. In electronic tolling applications, the different locations where a vehicle is identified represent electronic "tolling booths," and the travel time between two tolling booths can be computed. It is also possible to deploy RFID technology for the main purpose of collecting travel time data. An RFID system used in toll collection or for collecting travel times has four main components:

- A RFID tag present on vehicles,
- Antennas,
- RFID Readers,
- A central computing and communication facility.

As a vehicle approaches a reader site, an overhead antenna emits a signal which is reflected back by the tag which provides the system with a unique code to identify the vehicle. The on-site reader stamps this data with the time and location and this information is sent to a central facility where the travel time can be calculated. An example of a roadway RFID system architecture is shown in Figure 2.5. Several reader-antenna installations would be deployed along the corridor being tolled to enable the collection of travel time data.

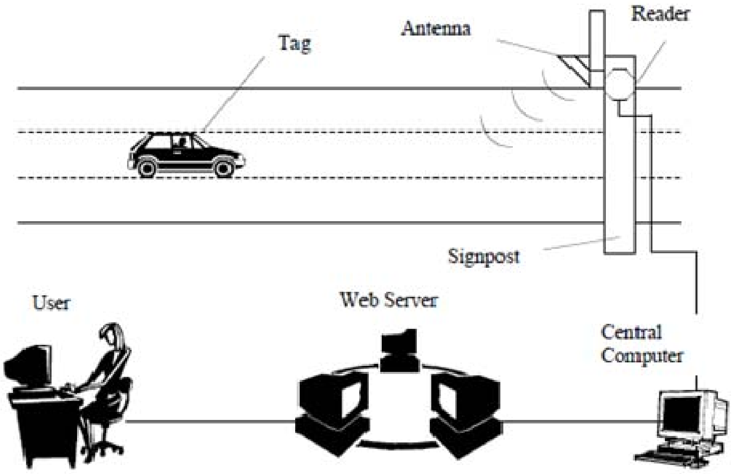


Figure 2.5: Roadway RFID system architecture (Wright and Joy, 2001)

According to Wright and Joy (2001), the costs of a roadway RFID system are significantly lower than loop detectors and video image-based systems. The travel time data collected is accurate; however, there are privacy issues since the RFID tags are unique identification tags that are issued to specific vehicles.

2.7 GPS TRACKING SYSTEMS

At the most detailed level of data collection, GPS-based vehicle tracking systems are used to collect travel times, which are derived from detailed location and time obtained from a GPS receiver as a vehicle travels. The vehicles equipped with this technology for the purposes of collecting travel data are often referred to as “probe vehicles,” and they may also use cellular communications for real time transfer of GPS receiver data (i.e., time and location data). Herrera et al. (2009) conducted a study utilizing 100 vehicles carrying GPS-enabled Nokia phones on a 10-mile stretch of a freeway near Union City, California. There was a single 8-hour test period and data was collected using virtual trip lines (VTL), which are geographical markers stored in the phones that trigger position and speed updates when the phone crosses them. For each of these VTL segments, a travel time is calculated and this information is broadcast in real time over the Internet. The information technology architecture of the system is shown in Figure 2.6.

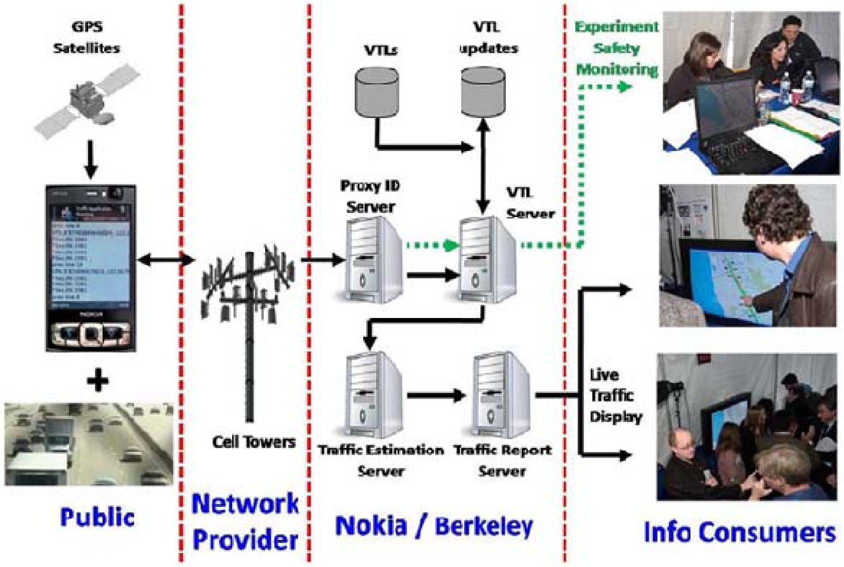


Figure 2.6: GPS travel time collections system architecture used in Herrera et al. (2009)

When a vehicle crosses a VTL, an update is generated with the following data: mobile device identification information, speed, timestamp, virtual trip line identification, and the direction of virtual trip line crossing. These updates are sent to the proxy server where, for privacy purposes, the mobile device identification information is removed from the data.

It is known that GPS tracking systems can generate accurate travel time data. It has also been shown (Hoh et al., 2006; Hoh et al., 2007) that the anonymity of time and location data collected by GPS-based tracking systems is not enough to guarantee privacy since data clustering techniques can often be used to identify a vehicle's home location. Aside from privacy concerns, implementing such GPS-based tracking systems in enough probe vehicles to reliably cover a metropolitan area would be cost prohibitive.

2.8 CROWD SOURCING

The most recent advancement in the collection of travel data is currently being implemented by companies providing web-based services (best exemplified by Google) in the form of a GPS-based location and speed data collection system. While it is not a travel time data collection system, it is included because it produces information that can be used to infer travel times. Google Maps, a web-based mapping application, provides an overlay to its maps that displays general travel speed categories for many main signalized roads and highways. To collect location and speed data, Google is applying a method they call *crowd sourcing*. As users with web-enabled, GPS-equipped mobile devices access Google Maps and then enable a feature that shows their location on the map, speed and location data are sent to Google.

As more users access Google Maps in this fashion while driving, more up-to-date and more widespread speed and location data is provided. If the mobile device is not equipped with a GPS receiver, it may be possible to use cell tower triangulation to obtain approximate location and speed data. This method of collecting speed and location data does have privacy concerns, and it is not clear how many users currently understand how this data is obtained. The public announcement from Google explaining this method was made on August 28, 2009. Other concerns are that mobile devices are not restricted to vehicles (i.e., pedestrians and/or cyclists could also provide data) and thus some accuracy issues may arise. Furthermore, the data collected is also owned by a private enterprise, potentially limiting its use and increasing the cost of outside applications that could use the data.

2.9 COMPARISON OF BLUETOOTH TECHNOLOGY FOR TRAVEL TIME DATA COLLECTION WITH OTHER METHODS

Applications that estimate travel time based on the collection of time-stamped media access control (MAC) addresses from Bluetooth-enabled devices have recently appeared in the literature. The road-side Bluetooth reader first broadcasts an inquiry to other Bluetooth devices in the surrounding area and waits for a response. When a Bluetooth device receives the inquiry, it replies back to the road-side Bluetooth reader. The reply message contains the MAC address of the Bluetooth device which is a unique identifier of the device. By recording both the time stamp (i.e., time when it receives the message) and the MAC address from two non-co-located stations along a road segment, average travel time can be calculated.

Figure 2.7 shows the general placement of various travel time collection technologies on the cost vs. accuracy vs. privacy spectrum. The use of enabled Bluetooth technology present in vehicles as a vehicle signature provides an advantageous combination of cost, accuracy, and privacy protection relative to competing methods.

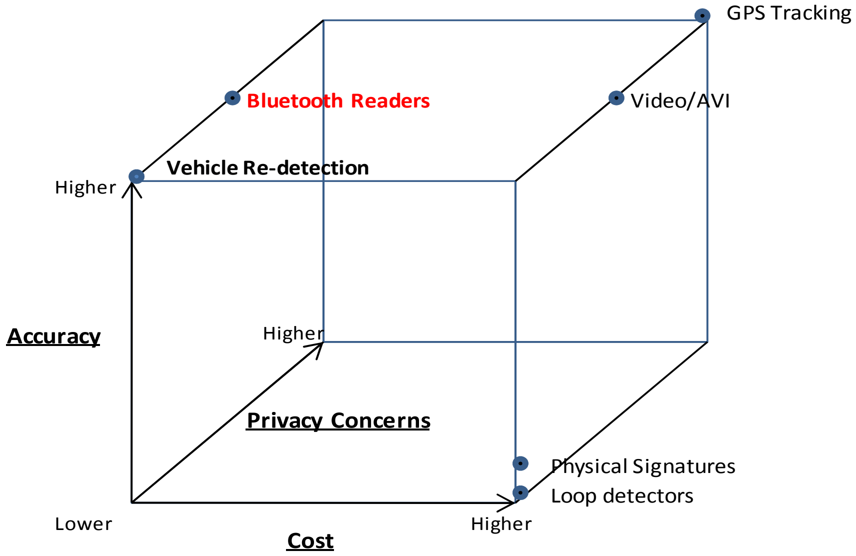


Figure 2.7: Bluetooth Readers relative to other ATTDC methods for travel time data collection with respect to cost, accuracy and privacy concerns

3.0 PRIVACY ISSUES RELATED TO AUTOMATIC TRAVEL TIME DATA COLLECTION METHODS

Travel time information for a road and highway system is very important to transportation engineers and planners interested in improving transportation system efficiency. However, some of the enabling automated travel time data collection (ATTDC) technologies have the potential to raise public concern over privacy. Due to the nature of the data collected in some ATTDC methods (e.g., location/time and vehicle identification), there exists the potential to use this data in ways that could prevent public acceptance of the ATTDC. This potential needs to be recognized and addressed before an ATTDC implementation is deployed. In this section, the general issue of privacy related to the use of ATTDC is explored in more depth.

Since ATTDC is an ITS application, information concerning public attitudes towards privacy with respect to ITS applications in general is presented. This is followed by a privacy categorization scheme for ITS technologies. Issues directly tied to specific applications or technologies that can be (or are) used for ATTDC are presented for electronic toll collection systems (RFID technology), GPS tracking systems, location-aware mobile devices, and Bluetooth readers. Guidelines and information that could facilitate the wide acceptance by the general public of new technologies with privacy concerns are discussed at the end of this section.

3.1 PUBLIC ATTITUDES RELATED TO PRIVACY AND ITS

According to the results of a study conducted by ITS America (Ogden, 2001), people can be categorized in three different groups based on their sensitivity to privacy invasion. These categories are *privacy insensitive*, *privacy fundamentalist*, and *privacy pragmatic*. People considered insensitive to privacy concerns are not threatened by the types of technologies employed in ITS applications, but are aware that those more sensitive to privacy concerns may affect technology development. Privacy fundamentalists are concerned about all forms of information gathering. They believe that just the absolute minimum amount of information must be collected and stored. Finally, those in the privacy pragmatic group (even though they are aware about the potential misuse of the information) still desire the benefits provided by the gathered information.

As shown in Figure 3.1, 20% of the population sampled in Ogden's study labeled themselves as privacy insensitive, 25% as privacy fundamentalists, and 55% as privacy pragmatists. If the results of this study still hold true today, it is clear that there is a significant portion of the population that is concerned with the potential misuse of personal information (i.e., 80%). However, 55% (i.e., privacy pragmatists) are willing to provide this data if they will benefit from the technology. Nevertheless, a quarter of the population feels that actions are needed to minimize privacy concerns.

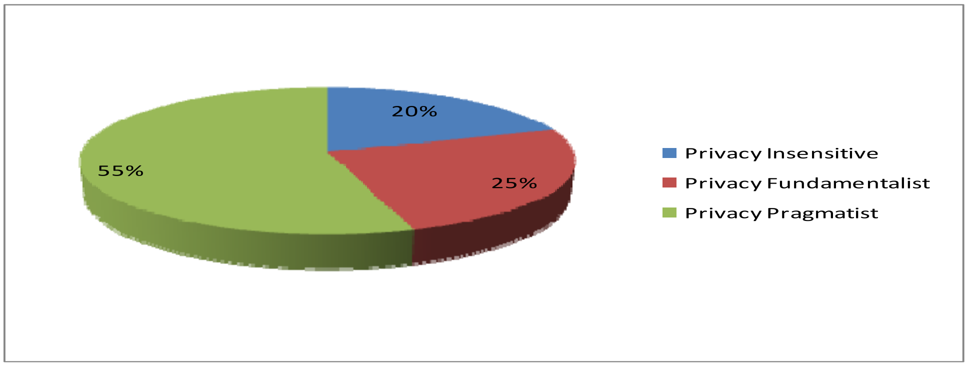


Figure 3.1: Attitudes of the public towards privacy and ITS applications (Ogden, 2001)

3.2 CATEGORIES OF ITS TECHNOLOGIES

Douma (2009) categorized technologies used for travel time data collection into the following three groups based on their potential impact on privacy:

- **Technologies with no impact on privacy**. Only system-level data are collected. No information is gathered that identifies either the vehicle or the driver. An example of technology in this category is loop detector systems.
- **Technologies with moderate impact on privacy**. Some information on the vehicle needs to be provided in order to operate the system or provide a service. However, the user has the option to accept or reject the service. Example technologies include license plate recognition and electronic toll collection systems.
- **Technologies with high level of impact on privacy**. This category includes technologies in which the driver and the passengers of a vehicle can be observed and identified, e.g., breathalyzer readers connected to ignition interlock systems.

The level of impact that a specific technology has on privacy also depends on the implementation. Some technologies may or may not expose the identity of the driver or the vehicle. For example, license plate recognition cannot reveal the identity of a driver without his or her picture. Similarly, an electronic toll collection tag purchased from a third party may not reveal the identity of the driver when using an electronic toll collection system.

It is clear that many of the vehicle re-identification ATTDC methods previously discussed may be considered to fall in the “moderate impact” category. However, establishing a wide-scale ATTDC system removes the option to reject the service. This will have the effect of increasing the potential perceived impact on privacy.

3.3 SPECIFIC PRIVACY AND SECURITY ISSUES OF AUTOMATED TRAVEL TIME DATA COLLECTION SYSTEMS

In this section, more detailed privacy and security concerns related to the automatic collection of travel time data are summarized. These privacy and security concerns are presented for particular ATTDC methodologies being employed. Vehicle re-identification ATTDC methods that used physics-based signatures (e.g., magnetic, inductive, and laser-based) are not discussed since they have not been widely utilized.

3.3.1 Electronic Toll Collection (ETC) Systems - RFID Technology

Information collected by ETC systems can potentially be used in surveillance and telemarketing applications. Thus, it raises concerns associated with loss of anonymity, and other parties having access to the personal information collected from the users of the system. It is important to note, however, that in the context of data collected by ETC systems, different categories of users may have different levels of privacy concerns. For example, operators of commercial vehicles (i.e., buses, taxis, and couriers) seem to have fewer concerns about privacy and may in fact benefit from the data being collected (Ogden 2001). For example, a passenger waiting for a bus or a taxi may be able to obtain an estimated time of arrival. The specific privacy concerns raised by data collection in ETC systems are as follows (Ogden, 2001):

- **Electronic tags are linked to user account identity**. ETC usually requires the creation of an account for each tag issued. Tags are typically linked to a credit card. A potential solution to this problem is to utilize a “smart card” so that payment transactions can be made anonymously between the user and the ETC system. Thus, it becomes impossible for the toll road operator to identify the user.

- **The collection of positioning data and the compilation of records related to individual travel behavior and patterns of use**. This is considered the most sensitive issue. Examples of travel behaviors are driving habits, regular destinations, location of family members and friends, time of driving, etc. This data can also be matched with other personal information, such as insurance, credit, buying habits, marital status, health data, etc.

- **Enforcement rules**. If the vehicle violates the regulations by not paying tolls or has insufficient credit, a photograph of the car will be taken in order to create a link to the vehicle registration record. By having no direct link between the tag and registration record, the identity of the driver is not revealed by the ETC system but by the photograph taken by the enforcement system.

- **Tag movement is monitored and recorded**. As previously mentioned, certain tag users (such as commercial road users) may benefit from being monitored by the ETC system for security purposes or to avoid hazardous conditions. In general, vehicle surveillance in the system should be prohibited unless a prior agreement exists between the tag user and the ETC system operator.

- **Access to information by other parties**. Since the information collected by the ETC system may have value to other parties, many ETC operations in the United States receive requests to disclose user account information (Ogden, 2001). Parties interested in this information may include market analysts, curious spouses, and employers tracking employees. However, none of the records have ever been released. When a user sets up an account with an ETC system, this event is considered as informal consent that the information provided will be used by the operator to enable the transaction. However, the user should also be informed and require his or her consent to use the information for any other purposes.

3.3.2 License Plate Recognition Systems

License plate recognition (LPR) systems are used for both law enforcement and in transportation applications. Examples of license plate recognition systems used to estimate travel time include those reported by Bertini et al. (2005) and Friedrich et al. (2008). Due to the ability of LPR technology to directly associate a license plate to a vehicle owner, LPR systems generate privacy concerns related to how the data collected by the system is used and also whether or not people are identified and/or monitored. Since LPR, like RFID used in electronic toll collection, is a vehicle re-identification method, the privacy concerns are directly connected to the information contained in the signature (i.e., the picture of the license plate number). For LPR systems, the amount of information that is linked to a license plate is detailed and personal. Thus, all of the privacy concerns documented above for RFID (used in ETC systems) are the same as for LPR systems, although the level of concern may be higher.

License plate information may be encrypted when transmitted to the central data storage location to alleviate privacy concerns and to prevent unwanted parties from accessing the information collected. Examples of this approach are found in the studies of Bertini et al. (2005) and Friedrich et al. (2008) where the license plate information was encrypted and sent via telephone communications to a central server.

3.3.3 Location-Aware Mobile Devices – GPS-Based Tracking Systems

GPS-based tracking systems are a subset of a larger category of devices called location-aware mobile devices. In addition to GPS-based tracking units, mobile devices such as cell phones and personal digital assistants (PDAs) that use either the surrounding cellular network or GPS receivers built into the device to determine location are examples of location-aware mobile devices. Location-aware mobile devices have been utilized extensively in intelligent transportation systems (Minch, 2004; Herrera et al., 2009). Minch (2004) separates the privacy concerns of location-aware mobile devices into four categories.

- **Collection of location information**. The main concern is how the location determination function in the device is activated (i.e., automatically or by request).

- **Location information retention.** Fundamental questions in this category include who decides what data is stored, where it is stored, and how secure and how long the data is kept. The potential of future use and misuse of information depends on the type of information stored and the location where the data is kept. Limiting the time that information is stored can also avoid future access.

- **Use of location information.** How collected information is utilized will directly impact the level of privacy concerns. For example, time and location information may be used to only compute specific road segment travel times, or it may be used to find the complete travel path of a specific vehicle.

- **Location information disclosure.** Revealing location information to other parties will generate concerns over privacy. If travel time data is collected by a private company these concerns will need to be addressed specifically. There are examples where service providers were sending users' telephone numbers to web sites visited from their internet-enabled mobile phones (Minch, 2004). This increases the potential of a user being tracked, especially with the availability of location information available with GPS enabled cell phones. The method of *crowd sourcing* obtains location and time information when specific web sites are accessed utilizing a GPS-enabled mobile device.

3.3.4 Privacy Concerns with Bluetooth Technology

Bluetooth technology, like RFID and LPR, is a vehicle re-identification method and would share similar privacy concerns, but the signature utilized (i.e., a MAC address) is much less directly tied to potentially sensitive information than a license plate or RFID number. However, even though it is quite difficult to derive the identity of the device's owner via a MAC address, there is evidence to suggest that by having such data, i.e., partial knowledge of the MAC address combined with other information such as user behavior, the system can be compromised and made vulnerable to malicious attempts to track the user.

In principle, if the MAC address of the device is set by the manufacturer, there is a possibility that a link can be established between the product part number and the owner via the product registration database or product warranty. In practice, however, there are several reasons to believe that the real chance of tracking back a MAC address to a user is low due the following reasons:

- The owner and the user may be a different person.
- The device may be purchased without revealing the owner's identity, e.g. using cash to buy the product.
- The product may not have been registered.

3.3.5 The Security of a Bluetooth Device

In addition to privacy concerns, Bluetooth technology also has some security concerns (Munro, 2008; Wong and Stajano, 2005; Cross et al., 2007). Wong and Stajano (2005) reported that the MAC address in Bluetooth is closely tied into the access procedure of the device. Cross et al. (2007) showed that even if a Bluetooth device is not in discovery mode, algorithms can be designed to gain access into the device in less than 24 hours.

It should be noted that the studies referenced above mostly address proof-of-concept security concerns. In a travel time measurement scenario, a Bluetooth equipped device is located inside a moving vehicle. This makes it less likely to gain access through the Bluetooth security protocol within a short period of time. However, if the driving route is a pattern, then the system would be more susceptible to an eavesdropping attack because Bluetooth devices would be scanned always at the same time by the Bluetooth readers installed along the road.

3.4 GUIDELINES FOR MINIMIZING PRIVACY/SECURITY CONCERNS

It has been shown that specific ATTDC methods utilizing signatures such as RFID tags, license plate numbers, MAC addresses, and GPS-based tracking systems all have the *potential* to be utilized in a way that will cause significant privacy concerns. There are many ways that data may be truncated and encrypted; however, the acceptance by the public of ATTDC methods primarily relies on trust. In this section, an overview of some work addressing similar issues for telematics (i.e., information obtained from vehicles using telecommunications and computing technology) is presented.

3.4.1 Keys to Success on Creating Trust

One of the most important and difficult challenges facing security and privacy in automotive telematics is *trust* (Duri et al., 2002). Trust must be established by both the users and service providers to ensure that the end-to-end system is doing the "right thing" at all times. Users need to trust service providers in that the privacy of their personal information is being protected and service providers need to trust users to protect the integrity of the data.

In general, telematics applications will be successful if providers know that the data they receive is accurate and if end users know that their privacy is assured. Therefore, the following issues must be addressed.

- **Integrity of telematics information.** That is, the generated/stored in/ transmitted data (i.e., user data, vehicle data, time and location information, and even executable software) must be protected.

- **Users must be assured that their privacy is respected**. It also means providing access to users to all logs and repositories concerning user data.

- **Security is in place to protect data while enabling the sharing of data.** To satisfy these key concepts, a defense-in-depth approach needs to be implemented to build a secure platform from the ground up. This needs to be done in both the hardware and software that make up the in-vehicle client and service provider platforms themselves. The development of the architecture should be based on open standards and accepted practices where they exist, by insisting on openness where new innovations are necessary, and by subjecting the architecture and its components to appropriate review and security evaluations. Finally, users should be able to define their privacy policies by providing consent before data is collected and used.

3.4.2 Providing Alternatives and Value

Duri et al. (2002) compared the growth of future ITS applications that employ vehicle telematics with the early growth of applications on the Internet (Hoffman et al., 1999). The results of their study revealed that in order to utilize the full potential of e-commerce over the World Wide Web (WWW), gaining trust from web-based users was a critical factor. It was found that a large number of online users were discouraged to shop online due to a lack of confidence in how the personal data they provided was used and also reported that there was a high demand from users to have the ability to control the information they provided on a website. Without a system that properly manages users' privacy concerns, it was estimated that system usage of ITS applications that employ vehicle telematics could drop as much as a fifty percent (similar to what was observed with WWW users).

Riley (2008) found that privacy concerns are greater in systems where money is collected due to the uncertainty of how the personal information is used and the connection that is established with the money transfer system. A case study of Fastrak (the electronic toll collection system employed in California) showed that the consumer's value of perceived privacy was more important than the convenience provided by the toll system and that this was the main cause for the slow adoption of the system.

For a wide-scale deployment of an ATTDC system, privacy concerns can be minimized if the public is allowed to choose when to participate, which may be difficult for some ATTDC systems. Additionally, the value of participating in the system should be clear and it should be readily accessible to the public.

3.5 SUMMARY

It is evident that different ATTDC technologies generate differing levels of public concern over privacy. In the specific case of the use of Bluetooth technology to collect MAC address data from Bluetooth-enabled devices (e.g., cell phones, navigation systems, hands-free car kits, etc.) to estimate travel times, these concerns are also present. Therefore, it is very important that the organization responsible for designing, installing and operating such a system fully discloses the location of readers and the measures taken to ensure that personal information collected (if any) will be used exclusively for the purpose it was intended. It is important to mention that, in contrast with other ATTDC based systems, a Bluetooth-based travel time data collection system does offer the public the opportunity to opt-out by turning off their Bluetooth compatible devices. Once these systems become more widespread, the value they provide will likely offset privacy concerns.

In this project, the specific approach followed to protect citizen's privacy is explained in chapter 6.0. Essentially, the first five leading characters and the last trailing character were removed from each individual MAC address collected making tracking of individual drivers extremely difficult (if not impossible).

4.0 INTERVIEWS WITH DEPARTMENTS OF TRANSPORTATION

An important component of task #1 involved contacting departments of transportation (DOTs) known to have experience with Bluetooth-based travel time data collection. The main objective pursued in contacting these agencies was to better understand the state of development of their Bluetooth travel time data collection projects, particularly with regards to antenna placement for data collection, and to identify potential improvements to the DCU platforms developed in Phase I.

This process began by contacting a person associated with the project at a state department of transportation (DOT) via email to inquire about the possibility of conducting an interview. A positive response was obtained from all five DOTs contacted. In three cases (i.e., Maryland DOT, Virginia DOT and Washington DOT[1]), the interview was conducted with a faculty member leading a university project co-funded by these DOTs. The interviews with the Indiana DOT and California DOT were conducted with the Director of the Traffic Management Centers Division and with a staff engineer in the Office of Traffic Operations Research in the Division of Research and Innovation, respectively. The research team is not aware of any other DOTs working in this area besides the five DOTs presented in this section.

4.1 MARYLAND DOT

This interview was conducted with Dr. Stanley E. Young and Nick Ganig of the Center for Advanced Transportation Technology (CATT) at the University of Maryland (UMD). In late 2007, UMD started a project titled *Bluetooth Traffic Monitoring Technology* with support from the Maryland State Highway Administration. By mid-2008, the commercialization of the Bluetooth reader units started through a startup company called Traffax, Inc. Dr. Young and Mr. Ganig serve as the CEO and CTO of Traffax, respectively.

4.1.1 Data Collection Unit

The data collection unit developed by Traffax is about the size of a briefcase, as shown in Figure 4.1. This portable unit (referred to as BluFax) is powered by a battery and lasts approximately six or seven days without recharging. BluFax units have a GPS module used to keep a record of the location where they are installed along the road and also to synchronize the internal clock of the unit.

BluFax units utilize a Class 1 power transmission omnidirectional antenna. This antenna allows for sufficient coverage of both directions of a divided freeway. To optimize the detection rate of the BluFax units, Traffax recommends that the antenna be placed three meters above the ground.

[1] Most of the information included in this report for the Washington DOT was obtained from an article presented at the Transportation Research Board (TRB) meeting on January, 2010.

The Debian distribution of Linux is the software platform used with the BluFax units. Linux was selected due to the cost benefit that it offers for a relatively small production of units. Traffax has developed its own proprietary software code to scan MAC addresses from Bluetooth-enabled devices via the BluFax units. To increase privacy protection, Traffax is planning to incorporate a truncation encryption at the unit level. (No further details were available due to intellectual property issues).

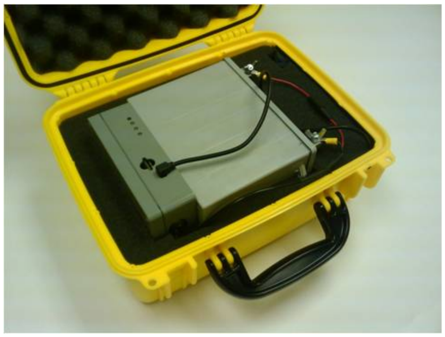


Figure 4.1: Traffax's BluFax unit (http://www.traffaxinc.com, 2009)

4.1.2 Data Storage and Analysis

The travel time data collected is stored in the BluFax unit in a removable 2 GB micro SD card. A prototype unit that reports the data collected once a day through a cellular network connection is currently undergoing testing.

Traffax has developed its own data analysis software called BluSTATS, which is based on the analytical research and development performed at the University of Maryland's Center for Advanced Transportation Technology and optimized for commercial use. BluSTATS is capable of matching, filtering and displaying travel times and speeds derived from the Traffax Bluetooth data collection units. BluSTATS also filters the data from outliers which are defined as any data points over 3 standard deviations away from the mean. As an optional feature, BluSTATS can work concurrently with Google earth.

4.1.3 Current Implementations

BluFax units have been used to collect travel time data at intersections as well as highways. Traffax indicated that if 2 to 3% of the total volume of traffic on a road is collected, that should be sufficient to obtain meaningful travel time estimates. However, a higher percentage might be required on signalized arterials. These percentages translate roughly to about three samples (i.e., MAC addresses) every five minutes.

4.1.4 Future Direction

Traffax suggested that interesting issues for future research may include investigating how travel time data could be collected in multi-use lanes such as local, express high occupancy vehicle lanes (HOV) and high occupancy/toll (HOT) lanes. This presents a challenge especially from the perspective of antenna design.

Another interesting issue is the analysis of origin and destination (O/D) data in urban areas. O/D data can now be collected faster, easier, and in larger volumes thus enabling more detailed analyses that could result in infrastructure modifications that can minimize traffic-related issues (e.g., congestion).

4.2 VIRGINIA DOT

This interview was conducted with Dr. Ramkumar (Ram) Venkatanarayana and Mr. Teang (Tim) Ngov. Dr. Venkatanarayana and Mr. Ngov are a research scientist and a system administrator, respectively, with the Center of Transportation Studies in the Department of Civil and Environmental Engineering at the University of Virginia. The University of Virginia team started working on travel time data collection with Bluetooth technology in April 2009 and is currently in the initial stages of proof of concept testing.

4.2.1 Data Collection Unit

The University of Virginia team has built only one data collection unit with commercial off-the-shelf (COTS) hardware. It includes a board with 256 MB of RAM and a Vortex chip manufactured by DMT (a Taiwanese company). The unit uses a battery that lasts up to 36 hours and uses a flash card to store the data collected in the field. Data collection period is two days. The use of different antenna types or locations has not been investigated so far but it is scheduled to occur in the near future.

Initially, the software in the data collection unit was embedded Windows XP; however, this was later replaced with Windows XP Pro on an 8 GB flash card. Currently, the unit uses Ubuntu Linux since it is easier to download software for it. Dr. Venkatanarayana mentioned that the Windows versions lacked some processes that they needed. Code written in C language is used to scan Bluetooth devices to collect MAC addresses.

4.2.2 Data Storage and Analysis

The University of Virginia plans to store travel time data collected with the unit in a database. The specific database management system to be used is Oracle 9.

Despite the fact that they have collected travel time data with the unit, no algorithm has yet been developed to actually calculate the travel time. Based on their experience, travel time data collected in the field represents approximately 4% of the total traffic observed with loop detectors. This data was collected only on highways; they have not used the technology at intersections.

4.3 INDIANA DOT

This interview was conducted with Jayson S. Wasson, Director of the Division of Traffic Management in the Indiana Department of Transportation.

4.3.1 Data Collection Unit

The travel time data collection units currently in use by the Indiana DOT were developed in cooperation with the University of Maryland. There are two main types of units:

- Suitcase style (see Figure 4.1).
- Semi-permanent.

The suitcase style unit is powered by a battery and is more suitable for rural areas where the use of electrical service is not cost effective. Thus, it provides more flexibility with regards to deployment and installation. The semi-permanent unit is typically installed on portable dynamic message signs and is powered by solar panels.

The Ubuntu distribution of Linux is the operating system used in these units. Memory capacity is 1 Gbyte. Currently, the last 3 octets (1 octet = 8 bits) of the MAC address are deleted to protect privacy. This is done at the server side of the system architecture (at the database). However, it is envisioned that this process will take place at the data collection unit level on a permanent installation maybe through hashing.

The antennas used by the data collection units are omnidirectional. However, high directional antennas are used sometimes perpendicular to the roadway. To maximize performance, antennas are installed five to seven feet above the ground, which is roughly the same height as a vehicle's window.

4.3.2 Data Storage and Analysis

Indiana DOT has implemented permanent installations of the data collection units since April 2009 and the system has since been integrated within Indiana DOT's ITS architecture. In a previous deployment of the system in a work zone on I-65 in North Western Indiana, a total of 1.4 million travel time records were obtained over a 12-week period.

Data collected is transmitted via modem in real time to the central database. For example, when semi-permanent units are utilized, the central database queries the unit for the median of the collected data, updates the previous estimate and displays the newest value on the message board.

4.3.3 Future Direction

This project is still in a research phase. Indiana DOT has not investigated use at intersections but has concentrated on rural and urban freeways as well as work zones. Their travel time algorithms are also in a development phase.

4.4 CALIFORNIA DOT

This interview was conducted with Joe Palen, Senior Research Engineer at Caltrans.

The data collection units developed so far by CalTrans are very simple and inexpensive. They consist of an antenna, electronics to read the MAC address data, an embedded controller that interprets and stores the data, and a communications module (e.g., cellular router, WiFi or Ethernet port). Stand alone units use battery power and units for more permanent installations utilize 120V AC. All the electronics are contained in a COTS NEMA 4x enclosure. No information was available as to the operating system currently installed in the units. Figure 4.2 depicts two example units developed by CalTrans.

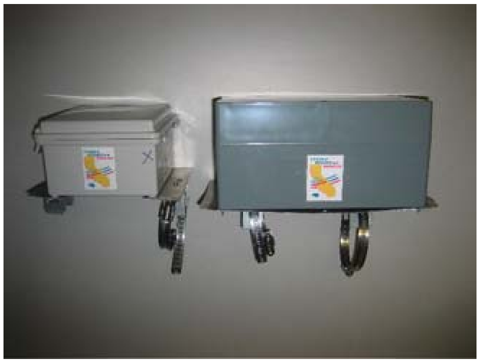


Figure 4.2: CalTrans' Bluetooth data collection units[2]

CalTrans has stopped working on this concept due to budgetary constraints. Once they are able to secure additional funding, the plan is to improve the performance of the data collection units by adding a GPS module for precise time synchronization and investigating configurations that minimize power consumption to possibly run the units entirely off of solar power.

Additional information about CalTrans Bluetooth project can be found on the following URL: http://www.dot.ca.gov/newtech/operations/bluetooth_web_page/intro.html.

4.5 WASHINGTON DOT

The project at Washington DOT is a collaborative effort with the Smart Transportation Applications and Research Laboratory (STAR lab) at the University of Washington. The details of this project can be summarized as follows.

4.5.1 Data Collection Unit

The first prototype was released in July, 2009, and is referred to as the *MAC Address Reader* (MACAD). The architecture of the initial product was quite similar to the prototype developed at OSU since both use an off-the-shelf electronic board from Gumstix, Inc. Different versions of the MACAD have been developed at the STAR lab, eventually resulting in a custom solution:

[2] http://www.dot.ca.gov/newtech/operations/bluetooth_web_page/intro.html, 2009.

- **Version 1.** The first prototype of the MACAD used a Gumstix Overo^{TM} Air Com 600MHz board with a transmission power of 3dBm. Power was supplied by 8 D-size batteries, which allow the unit to operate for 40 hours.
- **Version 2.** Version 2 of the MACAD was released in September, 2009. This version uses a Sparkfun 60MHz platform, which allows power control on the antenna, less heat generation, less power consumption, and can operate for five days using six D-size batteries.
- **Versions 2.1 and 2.2.** Minor revisions were made to the Sparkfun platform late in 2009 to include a weather proof antenna, Global System for Mobile Communications (GSM) capabilities to enable online data retrieval, an a GPS receiver module for automatic clock synchronization. This version of the MACAD is powered by a solar panel for continuous operation.
- **Version 2.3.** This version included a switch to a custom board with a low-power ARM processor and low-gain Bluetooth. This allowed the device to function for nearly two weeks using two lithium ion packs.

4.5.2 Current Implementations

The University of Washington performed multiple field tests with a range of antennae, including a 8 dBi omnidirectional and a 12 dBi directional antennae in a variety of configurations. These antennas had gains of 8dBi and 12dBi, respectively. The data collected with the MACAD unit using both types of antennas was compared to data collected with automatic license plate reader (ALPR) technology with respect to detection rate, matching rate, and average travel time error. Some of the fundamental questions behind this study were (1) whether Bluetooth sensors can act as an acceptable substitute for ALPR readers, (2) what error rates are associated with a particular configuration type and (3) how well Bluetooth measurements work on a short corridor..

The conclusions reached from the field test were that (1) the sample size detected by the Bluetooth readers was significantly smaller than the amount of vehicles detected by the ALPR system, (2) the error rates vary depending on the type of antennae setup, with directional antennae providing less accurate results due to a reduced sample size and (3) short corridors are subject to higher errors (about 10% for a mile-long corridor). Overall, the results were still representative of the actual traffic conditions. The Bluetooth reader using omnidirectional antenna showed to provide a better detection rate than the when the directional antenna was used. It was concluded that the reason for this was the smaller detection area produced by the directional antenna. Moreover, due to the smaller detection area, the detection rate was biased toward slower moving vehicles and faster moving vehicles were missed. Washington DOT reported that the current implementation of the Bluetooth reader costs less than $2,000 to cover all lanes while an ALPR system costs $15,000 per lane.

4.5.3 Future Direction

Future efforts in this project will continue to focus on redesigning the hardware device to make it more energy-efficient and increasing the detection range of the device.

4.6 SUMMARY OF DEVELOPMENTS AT DEPARTMENTS OF TRANSPORTATION

Table 4.1 presents a summary of the salient features of the Bluetooth-based travel time data collection projects at different DOTs. The majority of the DOTs contacted have DCUs that can operate on battery power or solar power; only California reported using 120V AC to power DCUs. Also, most of the DCUs use Linux as the operating system. Only two DOTs (Maryland and Indiana) have collected travel time data at both intersections and arterials. Maryland DOT and Indiana DOT were also the only ones post-processing MAC addresses to protect privacy.

Table 4.1: Salient features of Bluetooth-based travel time data collection projects at Departments of Transportation

				Data Storage Type		**Application**		**Communications**		
DOT (Collaborators)	**DCU Power Source(s)**	**GPS Module**	**Software Platform**	**Local**	**Remote**	**Inter**	**Arterial**	**Wired**	**W-less**	**MAC address processing to protect privacy**
MARYLAND **(University of Maryland, Traffax, Inc.)**	• Battery-powered (last 6 to 7 days) • Solar power	Yes	Linux (Debian)	2 GB Micro SD card	Yes	Yes	Yes	No	Yes	Yes
VIRGINIA **(University of Virginia)**	• Battery-powered (last up to 36 hours)	No	Linux (Ubuntu)	8 GB flash card	No	No	Yes	No	No	No
INDIANA **(Purdue University)**	• Battery-powered (last 6 to 7 days) • Solar power	Yes	Linux (Ubuntu)	1 GB	Yes	Yes	Yes	No	Yes	Yes
CALIFORNIA	• Battery-powered (one day to a week) • 120V AC	No	N/A	N/A	Yes	No	Yes	No	Yes	No
WASHINGTON **(University of Washington)**	• Solar power	Yes	Flash-ROM based Embedded System	Micro SD Card	Yes	No	Yes	No	Yes	No

5.0 SYSTEM ARCHITECTURES AND INSTALLATION CONFIGURATIONS

In this section, several system architecture options are presented. These options clearly identify the events that take place in each step of the process of collecting, processing and storing MAC address data from an information systems perspective. Also, alternative installation configurations for the Bluetooth readers will be discussed. These configurations are mainly defined by the location (e.g., traffic intersection versus arterial road), and the infrastructure available where readers will be deployed (i.e., power and communications options).

System architectures and installation configurations are not independent. Decisions made with respect to system architecture may impose constraints on the installation configuration, and vice versa.

5.1 SYSTEM ARCHITECTURES

In the context of this project, *system architecture* could be defined as “the organizational structure of a system, identifying its components, their interfaces, and a concept of execution among them.” (FHWA, 2006). Figure 5.1 depicts the three main components of the Bluetooth-based travel data collection system. These components are:

1. Data collection unit (DCU).
2. Communications network.
3. Applications/Database server.

The system components should perform (at a minimum) the set of functions described in the next three sections to allow for the collection, processing, and storage of travel time data.

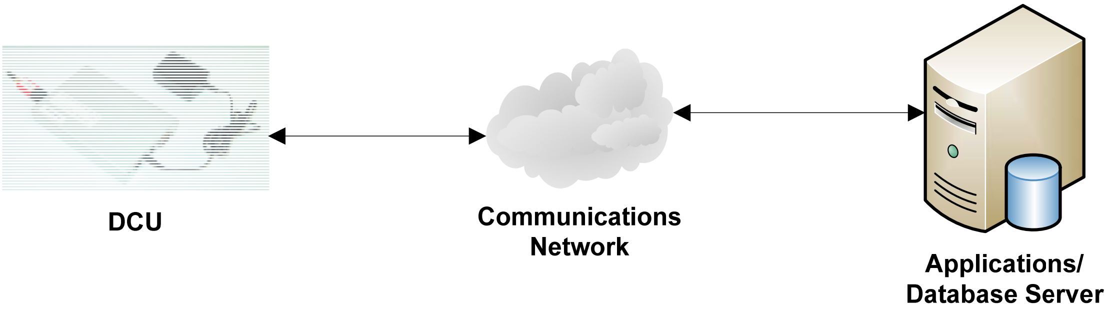


Figure 5.1: Main components of the Bluetooth-based travel time data collection system

5.1.1 Data Collection Unit Functionality

The data collection unit (DCU) is a fundamental component of the Bluetooth-based travel time data collection system. The DCU should be capable of performing at least the following functions:

- Accurately collect MAC addresses wirelessly from enabled portable devices (e.g., cell phones) under a full spectrum of environmental and traffic conditions.
- Add a time-stamp to MAC addresses collected so that origin and destination (O-D) data or travel time samples can be calculated.
- Store a large quantity of MAC addresses in internal memory before this information is extracted from the unit.

5.1.2 Communications Network Functionality

The communications network should enable the transmission of time stamped MAC address data from the DCU to a remote data storage location with minimum latency (i.e., the time delay experienced in a particular network) and with acceptable limits of bit error rate (i.e., number of received binary bits that have been altered due to noise and interference, divided by the total number of transferred bits during a studied time interval). Also, it should permit remote monitoring/troubleshooting of the DCUs.

5.1.3 Applications/Database Server Functionality

The applications/database server should house software applications and database procedures (i.e., queries and macros) to:

- Retrieve MAC address data from DCUs on a regular schedule or when needed.
- Sort MAC addresses based on their timestamp.
- Match MAC addresses that appear at distinct points of travel on the travel corridor of interest to compute a travel time sample.
- Store MAC addresses on a well-designed, maintainable database structure.
- Dispose of MAC addresses on a regular schedule.

5.1.4 Definition of Potential System Architectures

In defining potential system architectures that could be used for the collection of travel time data via Bluetooth on Oregon roadways, it is important to note that there are fundamental assumptions made in this project that will limit these possibilities. The first assumption is that the DCUs should use 120V AC as the source of power; no consideration has been made in this project for DCUs that draw power from a battery or a solar panel. The second assumption is that network connectivity between the DCUs and the remote server enabled by a cellular router is possible, but will be limited to small scale implementations due to the cost of hardware as well as the cost of contracting with a cellular service provider.

Given these assumptions, the following are the different “modes” in which the main architecture components could function individually (combinations of these individual functions will result in different architecture options):

1. **Data collection unit**.
   a. Data storage
      i. Local (in the DCU)
         - Requires calculation of required storage space and it is dependent of the frequency in which data is extracted from the unit.
      ii. Remote
         - Data can be retrieved on a set schedule to minimize loss of data.
   b. Data retrieval
      i. Push
         - A software application residing in the DCU manages the schedule for transmitting the data to an applications/database server.
         - Requires more coordination and perhaps tighter software security measures at the applications/database server (asynchronous transmission mode).
      ii. Pull
         - A software application residing in the applications/database server manages the schedule for retrieving the data from the DCUs.
         - Requires programming a well defined data retrieval schedule at the applications/database server (synchronous transmission mode).
         - Less software security measures needed at the at the applications/database server.
   c. Data filtering (i.e., removing duplicate MAC addresses).
      i. Local (in the DCU)
         - Requires more processing power in the DCU.
         - Data files transmitted to the applications/database server are smaller.
      ii. Remote.
         - Requires higher data bandwidth (larger files will be transmitted).
         - Provides more information (especially at intersections).
   d. Processing of data to protect privacy (i.e., MAC address digit removal).
      i. Local (in the DCU)
         - Requires more processing power in the DCU.
         - Data files transmitted to the applications/database server are smaller.
         - Increases protection of privacy.
      ii. Remote.
         - Requires higher data bandwidth (larger files will be transmitted).
         - Provides more information (especially at intersections).
         - Compromises protection of privacy.

2. **Communications Networks**.
   The type of communications network utilized will mainly depend on availability (i.e., options may not be available at every installation location). Three options will be considered:

a. Ethernet
b. Fiber optic
c. Cellular router

3. **Applications/database server**.
The modes in which the applications/database server will operate will be defined by the decisions that are made at the DCU level (and vice versa).

Table 5.1 presents potential system architectures, which combine the functions of the main system architecture components.

Table 5.1: Potential system architecture options

		Architecture Options		
		1	**2**	**3**
DCU	Data Storage	√		
	Data Retrieval	√	√	
	Data Filtering	√	√	√
	MAC address digit removal	√		√
Network Type	Ethernet		√	√
	Fiber Optic		(√)	(√)
	Cellular Router	√		
Applications/Database Server	Data Storage		√	√
	Data Retrieval			√
	Data Filtering			
	MAC address digit removal		√	

(√) – Indicates alternative communications network

The advantages and disadvantages of the potential system architectures proposed in Table 5.1 are as follows:

1. **ARCHITECTURE #1**
   a. **Advantages:**
      - Emphasis of the design is on the DCU as all the main functions related to data collection and data storage are performed there.

   b. **Disadvantages:**
      - More expensive DCU.
      - Data analysis still has to be performed at a central location (most likely the computer used to access the DCUs via the cellular router).
      - Not truly real-time system (depends on how often data is retrieved from DCUs).
      - Expensive to maintain (cellular router hardware cost and cellular provider service cost).
      - Single point of failure (i.e., DCU). Data will be lost if device malfunctions and memory cannot be read.

2. **ARCHITECTURE #2**
    a. **Advantages:**
        - Emphasis of the design is on the applications/database server.
        - Less expensive DCU.
        - More balanced design. Some of the data collection and data storage tasks are performed in the DCU and others in the applications/database server.
        - Could be real-time system if DCUs report data collected often.

    b. **Disadvantages:**
        - Could overload communications network (depending on the type of network selected) if DCUs transmit data too often.
        - Privacy protection may be compromised by performing MAC address digit removal in the applications/database server.
3. **ARCHITECTURE #3**
    a. **Advantages:**
        - More balanced design. Key data collection and data processing functions are evenly distributed between the DCU and the applications/database server.
        - Reasonably priced DCU.
        - Required data storage capacity in DCU is driven by how often data is retrieved from the device.
        - Privacy protection is enhanced by performing MAC address digit removal in the DCU.
        - Less data needs to be transmitted by performing data filtering in the DCU.

    b. **Disadvantages:**
        - Not truly real-time system (depends on how often data is retrieved from DCUs).

5.1.5 System Architectures: Design Recommendation

Several additional alternative architectures were defined by varying the distribution of functions among the main system components. However, the architectures described above fit best within the constraints and assumptions that are particular to the application of travel time data collection on Oregon roadways. The OSU research team implemented architecture #3; however, implementing either architecture #1 or architecture #2 would require minimal hardware and software changes.

5.2 INSTALLATION CONFIGURATIONS

The installation requirements for a DCU are straightforward. Each DCU requires the following to function properly:

1. Physical structure for securing the DCU.
2. Physical structure for mounting the DCU antenna and establishing a connection between the antenna and DCU.
3. Protection from the environment.
4. A power supply.
5. Remote connectivity for accessing the DCU, and transferring data.

For the purposes of this project, an installation configuration is defined as a specific combination of solutions for each of the previously stated requirements. The solution alternatives for meeting requirements 1 through 4 listed above will be restricted to infrastructure and hardware that exists and is currently in use. A prime example of this is a traffic signal control cabinet that would meet requirements 1, 3, and 4. The various existing infrastructure alternatives considered are:

- Signal control cabinet and variable message sign (VMS) cabinets (see Figure 5.2).
- Signal support poles (see Figure 5.3).
- Signal mast arms (see Figure 5.3).
- Variable message signs (see Figure 5.4).
- Overpasses.

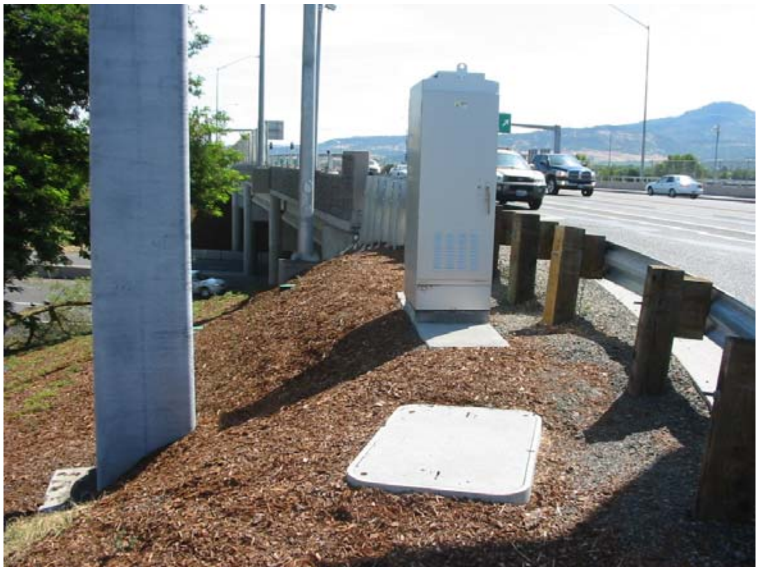


Figure 5.2: Signal control cabinet in Medford, OR

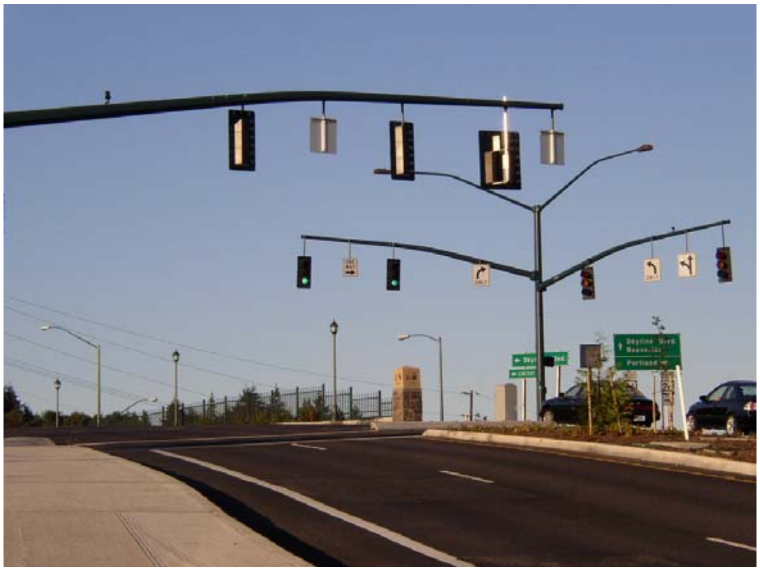


Figure 5.3: Signal mast arm with support pole in Portland, OR

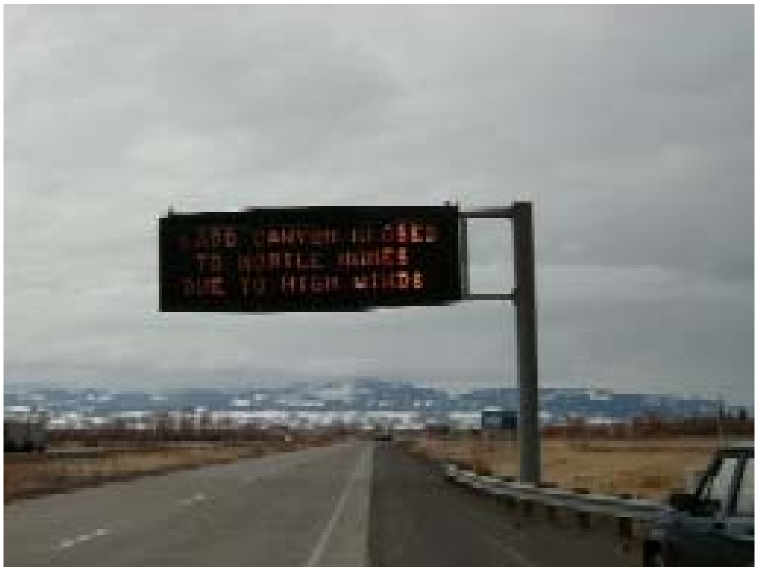


Figure 5.4: Variable message sign

A matrix that considers each of these alternatives with respect to the five installation requirements is presented in Table **5.2**. In Table **5.2**, cells at the intersection of specific physical installation requirements and existing infrastructure needed for installation are coded to indicate minimal or no action required (i.e., no background fill) versus action, design, and/or hardware required (i.e., gray background fill).

Table 5.2: Existing infrastructure mapped to installation requirements

	EXISTING INFRASTRUCTURE FOR INSTALLATION					
Physical Installation Requirement	**Signal Support Pole**	**Signal Cabinet**	**VMS Cabinet**	**Signal Arm**	**VMS**	**Overpasses**
Physical structure for securing the DCU	Metal straps are commonly used for mounting items similar in size	Provided by cabinet	Provided by cabinet	Mounting system will need to be designed	Mounting system will need to be designed	Mounting system will need to be designed
Physical structure for mounting the DCU antenna and establishing a connection between the antenna and DCU	Antenna should be very close if not integrated with the protective enclosure	A weather-proof means to connect antenna to DCU is required	A weather-proof means to connect antenna to DCU is required	Antenna should be very close if not integrated with the protective enclosure	Antenna should be very close if not integrated with the protective enclosure	Antenna should be very close if not integrated with the protective enclosure
Protection from the environment	Will require separate enclosure	Provided by cabinet	Provided by cabinet	Will require separate enclosure	Will require separate enclosure	Will require separate enclosure
Power supply	Available at some	Available	Available	Accessible?	Accessible?	Accessible?
Remote network connectivity	Requires cellular router	Network access available	Network access available	Requires cellular router	Requires cellular router	Requires cellular router
Comments	Enclosures need to be designed/ acquired	Primarily for arterial installation	Primarily for highway installation	Enclosures need to be designed/ acquired	Primarily for highway installation	Enclosures need to be designed/ acquired

A particular installation configuration and installation location will define an *installation environment*, which is defined as a combination of the following *installation environment variables*, some of which are controllable and others that are not changeable at a particular installation location. It will be assumed that the installation on a road/highway is primarily used to identify vehicles moving in a north-south travel direction. Distances from the DCU to locations in a lane are assumed to be measured on the plane observed from a top down view of the intersection. Within this plane these distances will be measured from a straight line originating from the DCU, and crossing a lane perpendicular to the direction of travel.

- The distance of the DCU antenna to the center of each northbound travel lane.
- The distance of the DCU antenna to the center of each southbound travel lane.
- The distance of the DCU antenna to the center of each westbound travel lane if applicable (e.g., at an intersection).
- The distance of the DCU antenna to the center of each eastbound travel lane if applicable (e.g., at an intersection).
- The height of the DCU antenna.
- The distance of the DCU antenna from the DCU.
- The location of the DCU relative to traffic flow (e.g., northbound or southbound side) with respect to the location of adjacent DCUs installed on the same road/highway.
- The presence of obstructions possible reducing antenna coverage.
- The distance of nearby residences, businesses, and pedestrian traffic if within maximum antenna coverage area.
- Other?

The values of these specific variables will be determined by the specific locations and installation configuration selected for DCU installation. In some installations there may be control over some variables. For example, if a signal support pole is used in the installation, and the DCU enclosure and antenna are integrated, the antenna height can be controlled. Also, in this example the antenna will be integrated with the enclosure used for environmental protection, so the distance of the DCU antenna from the DCU is not an issue.

A sample of “extremes” of different installation environments will identify requirements that will influence the DCU and/or antenna design. It will help to categorize different general types of installation environments. It is possible that having multiple interchangeable antenna designs is a more effective solution to accommodate various installation environments than a single general antenna design. A possible categorization scheme is:

- Intersections with less than or equal to *x* total lanes across in either north-south or east-west travel directions.
- Intersections with greater than *x* total lanes across in either north-south or east-west travel directions.
- Interstate highways with less than or equal to *x* total lanes across, and a wide center median.
- Interstate highways with less than or equal to *x* total lanes across, and a narrow center median.
- All other.

As an example, the specific values of the installation environment variables at two different intersections in Corvallis, Oregon, were measured and the results are presented in the next two sections.

5.2.1 Example 1: Monroe Street and 14th

The installation configuration assumed is an installation in the traffic signal control cabinet located on the northwest corner of the intersection.

- The distance of the DCU antenna to the center of each northbound travel lane
  - *34.5 feet.*
- The distance of the DCU antenna to the center of each southbound travel lane
  - *19.5 feet.*
- The distance of the DCU antenna to the center of each westbound travel lane if applicable (e.g., at an intersection)
  - *15 feet.*
- The distance of the DCU antenna to the center of each eastbound travel lane if applicable (e.g., at an intersection)
  - *26 feet, 36 feet.*
- The height of the DCU antenna
  - *Height of the signal control cabinet = 6 feet.*
- The distance of the DCU antenna from the DCU
  - *Less than 3 feet.*
- The location of the DCU relative to traffic flow (e.g., northbound or southbound side) with respect to the location of adjacent DCUs installed on the same road/highway
  - TBD
- The presence of obstructions possible reducing antenna coverage
  - *No significant obstructions.*
- The distance of nearby residences, businesses, and pedestrian traffic if within maximum antenna coverage area
  - *Adjacent heavy pedestrian traffic, and heavy pedestrian traffic on all sides of the intersection*
  - *Business on the same corner as the DCU, 20 feet from the DCU.*
  - *A single residence 55 feet from the DCU.*

5.2.2 Example 2: Circle Blvd. and Highway 99W

The installation configuration assumed is an installation in the traffic signal control cabinet located on the southwest corner of the intersection.

- The distance of the DCU antenna to the center of each northbound travel lane
  - *58 feet, 70 feet, 82 feet.*
- The distance of the DCU antenna to the center of each southbound travel lane
  - *34 feet, 46 feet.*

- The distance of the DCU antenna to the center of each westbound travel lane if applicable (e.g., at an intersection)
  - *106 feet, 121 feet.*
- The distance of the DCU antenna to the center of each eastbound travel lane if applicable (e.g., at an intersection)
  - *35 feet, 64 feet, 76 feet, 88 feet.*
- The height of the DCU antenna
  - *Height of the signal control cabinet = 6 feet.*
- The distance of the DCU antenna from the DCU
  - *Less than 3 feet.*
- The location of the DCU relative to traffic flow (e.g., northbound or southbound side) with respect to the location of adjacent DCUs installed on the same road/highway
  - TBD
- The presence of obstructions possible reducing antenna coverage
  - *No significant obstructions.*
- The distance of nearby residences, businesses, and pedestrian traffic if within maximum antenna coverage area
  - *Adjacent light pedestrian traffic, and light pedestrian traffic on all sides of the intersection*
  - *Parking lot on the same corner as the DCU, 10 feet from the DCU.*

5.2.3 Installation Configurations: Design Recommendation

These two installation configurations show large differences in distances, traffic volumes, and surrounding environment (e.g., pedestrian traffic, volume and proximity). The requirements characterized by these two examples will dictate different “optimal” designs. Since variety presented by potential installation locations is so large, the general strategy for this project will be to first select specific installation locations and design the DCUs for the requirements at the selected location.

6.0 PROCEDURES FOR DATA COLLECTION, DATA PROCESSING AND DATA STORAGE TO PROTECT CITIZENS' PRIVACY

In this task, the development and implementation of software tools to ensure that the collection, processing and storage of travel time data does not infringe on a citizen's right to privacy is discussed. Software was developed for both the DCUs as well as the database management system that will be used to manipulate the data collected from Bluetooth enabled devices.

Additionally, some specifics regarding the collection, processing and storage of travel time data from an information systems perspective are discussed. These specifics include the definition of the data elements of the travel time data records to be collected (i.e., semantics and syntax); whether data matching will be performed at the reader or at the central server; the format of the data that will be stored (i.e., origin and destination reads versus travel time samples) and later used by ODOT and other potential users.

6.1 COLLECTION, STORAGE AND DISPOSAL OF MAC ADDRESS INFORMATION

The steps implemented through software in the DCU to collect data from Bluetooth enabled devices are as follows:

1. The inquiring device (i.e., DCU) retrieves its own MAC address and stores it in the file bluetoothDeviceINFO.txt. Any unnecessary information in the file is then removed and the file *bluetoothDeviceJustMAC.txt* is created containing only the MAC address of the reader.
2. The inquiring device sends data packets in an attempt to discover Bluetooth devices. If any devices are found, their MAC addresses are stored in the file *rawScannedMacs.txt*.
3. The first five leading characters and the last one trailing characters are removed from the MAC addresses saved in the file *rawScannedMacs.txt*. A new file is created that contains the new MAC addressed named *EditedMacs.txt*.
4. Finally, the file is renamed as *macs.txt*

Step 3 of this process is very important since it is here where the first five leading characters and the last trailing character are removed from the MAC addresses to increase privacy protection. Figure 6.1 shows these steps as a flowchart.

The Linux shell script contains code that would allow for removal of duplicated MAC addresses. This functionality exists in the code, but it was not implemented in this phase of the project. That is the reason for the change in the name of the MAC address file from *EditedMacs.txt* to *macs.txt*.

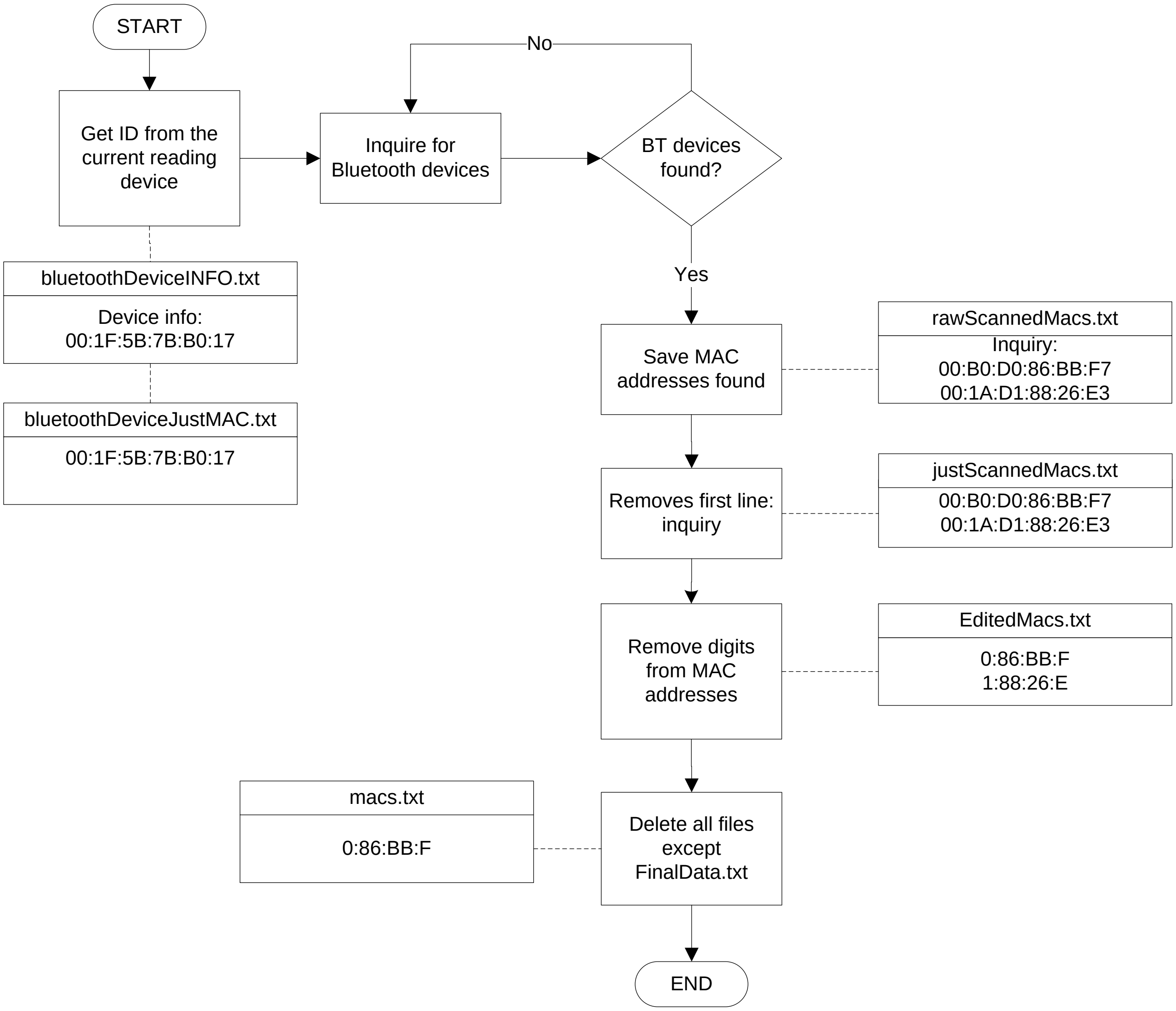


Figure 6.1: Steps taken by a DCU to collect data from Bluetooth enabled devices

6.2 DATA SEMANTICS AND DATA SYNTAX OF FILE *MACS.TXT*

As depicted in Figure 6.1, the file *macs.txt* contains unique (i.e., non-duplicate) six-character MAC addresses. This is the file that will be extracted (or transmitted) from the DCU and stored in a database management system (DBMS) to produce travel time samples.

The semantics (i.e., the meaning) of the individual data elements that will be either read or generated by the DCU is shown in Table **6.1**.

Table 6.1: Data elements of a travel time data record in macs.txt

Data Element	Variable or Fixed	Type	Data Format	Example Data
Bluetooth enabled device six-character MAC address	Fixed	Character	A:AA:AA:A	F:DA:7C:E
Date	Fixed	Date	MM/DD/YYYY	01/27/2010
Time	Fixed	Time	HH:mm:SS	17:42:25
DCU MAC address	Fixed	Character	AA:AA:AA:AA:AA:AA	00:01:95:09:5C:67

A: Alphanumeric character	MM: Month DD: Day YYYY: Year	HH: Hour mm: Minutes SS: Seconds

These data elements will be then be assembled into a record and saved in the file *macs.txt* following the syntax shown in Figure 6.2.

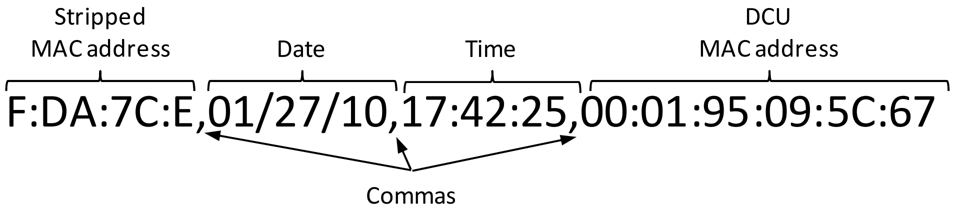


Figure 6.2: Syntax of the data elements for an individual record

6.3 SIZE OF FILE *MACS.TXT*

The estimated size of the file *macs.txt* in bytes will be function of the number of lines (i.e., data records) it contains. The following equation was found through the use of a script that obtains the size of the file every time a MAC address is added:

$$\textit{File Size (bytes)} = 46L \qquad (1)$$

- where L is the number of lines (i.e., data records) contained in the file.

Since one byte[3] equals 9.5367 x 10^{-7} Megabytes (Mbytes), the size of the file in Mbytes is

$$\textit{File Size (Mbytes)} = L * 4.3869 \times 10^{-5} \qquad (2)$$

Table 6.2 shows approximate file sizes based on a specific amount of lines contained in the file.

[3] One byte equals 1,048,576 bytes (i.e., 1024^2) for computer memory.

Table 6.2: Approximate size of file FileData.txt in mbytes based on number of data records

Number of Lines (rounded down to the nearest integer value)	File Size(in Mbytes)
22,795	1
1,139,800	50
2,279,500	100

6.4 DATABASE MODEL

A database model has been defined to store the MAC address data collected with the DCUs. The database model consists of two entities (i.e., tables): TBL_RAW_DATA and TBL_EQUIPMENT. The individual data elements and their data types are shown in Table 6.3 and Table 6.4, respectively.

Table 6.3: Data elements and data types of entity TBL_RAW_DATA

Data Element	Type	Description
Record_ID	Numeric	Primary key
Equip_MAC_Address	Alphanumeric	MAC address of the DCU
BT_MAC_Address	Alphanumeric	Six-character MAC address collected by the DCU
Input_Time	Date/Time	Timestamp of the collected MAC address

Table 6.4: Data elements and data types of entity TBL_EQUIPMENT

Data Element	Type	Description
Equip_MAC_Address	Alphanumeric	Primary key
ODOT_Device_ID	Numeric	Optional identifier
Equip_Type	Text	DCU model
Equip_Manufacturer	Text	DCU board manufacturer
Location	Numeric	Physical location of DCU

The two tables are related according to the entity relationship (E-R) diagram depicted in Figure 6.3.

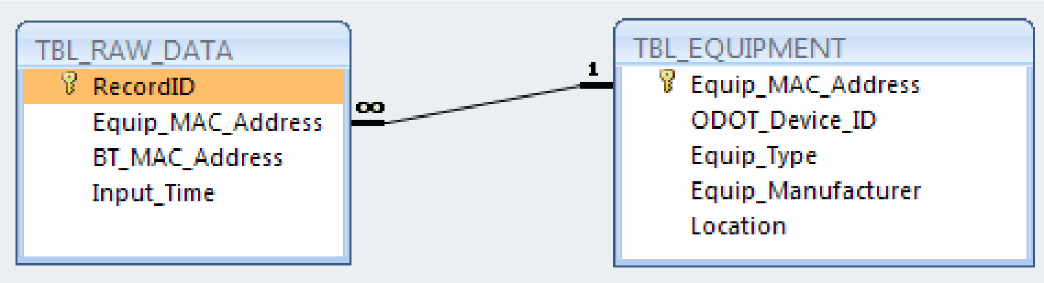


Figure 6.3: Entity-relationship diagram of database model

6.5 CHANGES MADE TO THE LINUX SHELL SCRIPT

Three main changes were made to the Linux shell script, as follows:

1. The main command that is issued to collect the data from Bluetooth (BT) enabled devices.
2. The development of an algorithm to check for duplicated MAC addresses.
3. Organizing the information collected.

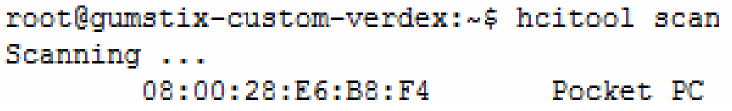
Additionally, the code was optimized by removing unnecessary lines and variables. A complete listing of the final code is included in Appendix C.

6.5.1 Changes to the Main Command in the Linux Shell Script

The format of the original main command used in the Linux shell script to collect the data (i.e., to read MAC addresses from BT enabled devices) was:

hcitool scan --flush --length=3 > rawScannedMacs.txt

In this command, the method *scan* from the class *hcitool* is used to discover BT enabled devices. It is known that the *scan* method command takes more time to process because the inquiring device (i.e., the DCU) will look for the user-friendly names of the BT enabled devices it is trying to communicate with (Albert & Larry, 2007). An example of the output produced when the *scan* method is used as part of the *hcitool* command is depicted in Figure 6.4, where “Pocket_PC” is the user friendly name of the BT device read.



```
root@gumstix-custom-verdex:~$ hcitool scan
Scanning ...
        08:00:28:E6:B8:F4       Pocket PC
```

Figure 6.4: Output obtained by using the scan method a part of the hcitool command

In an effort to improve the discovery of BT enabled devices, the *scan* method has been replaced with the *inq* method, which does not look for the user-friendly name of BT devices. The format of the new main command used in the Linux shell script is:

hcitool inq --flush --length=3 > rawScannedMacs.txt

7.0 INSTALLATION AND TESTING OF THE TRAVEL TIME DATA COLLECTION SYSTEM

The information presented in this section pertains to research tasks #5 (*Comprehensive performance evaluation of the travel time data collection system*) and #6 (*Travel time data collection and analysis*) of the work plan.

The original intent of task #5 was to evaluate the performance of a Bluetooth (BT) based travel time data collection system composed of several DCUs under more realistic operational conditions (i.e., not in a laboratory setting). Additionally, the impact on performance of other factors such as antenna design and placement and central software reliability were to be assessed. The objective of task #6 was to conduct comprehensive testing of the prototype DCUs with real traffic. Thus, DCUs would be installed at intersections, and/or other sections of highway where power and cover was available. Travel time data collected during the testing period would be analyzed to identify key performance measures, identify outliers and estimate travel time between two data collection points.

Unforeseen delays in the availability of the needed infrastructure to test the BT-based travel time data collection system required a reduction of the original scope of tasks #5 and #6. Nevertheless, several of the original objectives were completed successfully. First, two different BT-based travel time data collection systems were deployed. The first system, composed of two DCUs, was installed on a corridor located in Salem, OR, and was used extensively to characterize the performance of six different antennas using a single DCU. Later, the same system was used to collect travel time data samples using two DCUs. The two-DCU system also allowed for limited testing of the central software. Late in the project, a second BT-based travel time data collection system (composed of five DCUs) was installed along 99W in the city of Tigard, OR. Very limited data collection was done with these DCUs due to the lack of network connectivity.

The remainder of this section is organized as follows. First, the main components of the Bluetooth (BT) based travel time data collection system are presented. The experimental approach and the results of characterizing the performance of six different antennas with a single DCU are presented next. Finally, the experimental approach and the results of collecting travel time samples with two DCUs are presented.

7.1 BLUETOOTH BASED TRAVEL TIME DATA COLLECTION SYSTEM

Several approaches for travel time estimation based on the collection of time-stamped media access control (MAC) addresses from BT-enabled devices have been reported in the literature in recent years (Wasson, et al. 2008; California DOT, 2010; Traffax, Inc., 2010). This new approach to estimate travel times offers a number of advantages over more conventional methods, including lower costs of hardware and software, the volume of data that can be collected over time, and ease of implementation. The latter advantage makes this data collection method suitable for quick temporary or permanent deployment along different types of travel corridors, including interstate highways, freeways, and other principal and minor arterial systems.

A basic setup to collect travel time data via BT includes a DCU and an antenna. The DCUs that have been reported in the literature vary from commercial-off-the-shelf (COTS) laptop computers to custom-built units which may include additional capabilities, such as global positioning system (GPS) receivers and/or cellular communications modules for remote data transfer (Haghani et al., 2010; Haseman et al., 2010; Quayle et al., 2010).

7.1.1 Prototype Bluetooth-based Data Collection Unit

The prototype BT-based DCU unit used to collect travel time data is depicted in Figure 7.1. The DCU is a fundamental component of the BT-based travel time data collection system and is capable of performing several functions, including:

- Collecting MAC addresses wirelessly from enabled portable devices (e.g., cell phones) under a full spectrum of environmental and traffic conditions. To protect citizens' privacy, the reader unit's software removes the first five characters and the last character from the MAC addresses collected.
- Adding a time-stamp to partial MAC addresses collected so that origin and destination (O-D) data or travel time samples can be calculated.
- Storing a large quantity of partial MAC addresses in internal memory before this information is extracted from the unit.

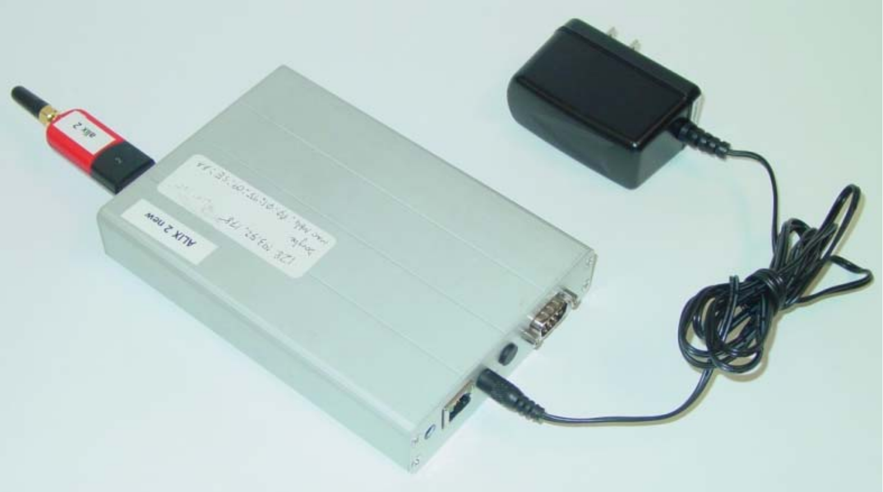


Figure 7.1: Bluetooth-based data collection unit

The DCU was assembled from commercially available components. Internally, the DCU is equipped with an ALIX WIFI board which provides several interfaces, including a 10/100 Base

T Ethernet port for network connectivity, an external memory slot for a compact flash card, a DB9 serial port, and dual USB ports (Ituner Networks Corporation, 2010). An external BT adapter (Sena Technologies, Inc., 2010) is connected to the unit via one of the USB ports. The DCU may be powered with a direct current (DC) jack or by using a power over Ethernet (POE) injector. The ALIX WIFI board is protected by an extruded aluminum enclosure to minimize the potential of damage resulting from the accumulation of dust and/or moisture. The operating system employed by the DCU is the Linux distribution Voyage Linux (kernel 2.6.26).

7.1.2 Data Collection Unit Antenna

There are several factors that may affect the quantity and the quality of the travel time data collected with a BT-based system. These factors may be associated with the firmware/software implementation or with the hardware components used (e.g., the type of BT chipset). An important decision that needs to be made in the hardware category is the type of antenna to use. Antenna characteristics, such as polarization and gain, must be matched to specific application environments to optimize the performance of a DCU.

There were two main factors that drove the process to select an antenna to be used with the DCU. These factors were *polarization* and *gain*. Polarization refers to the orientation of the electric field vector in the radiated wave. For example, Figure 7.2a depicts the orientation of the electric field (E-field) vector for an antenna with horizontal polarization (e.g., a Yagi antenna for TV reception). This antenna will not radiate and will not receive a wave polarized perpendicular to its electrical vector (Vizmuller, 1995).

Gain is a measure of the directionality of the antenna. Antenna gain is defined as the power output, in a particular direction, compared to that produced in any direction by a perfect omnidirectional antenna (i.e., isotropic antenna). For example, if an antenna has a gain of 3 decibels (dB), that antenna improves upon the isotropic antenna in that direction by 3 dB, or a factor of 2 (Stallings, 2005). Figure 7.2b depicts a comparison between the ideal concept of an isotropic antenna (Gain = 1) and a directional antenna with a value of gain greater than 1 along the horizontal axis.

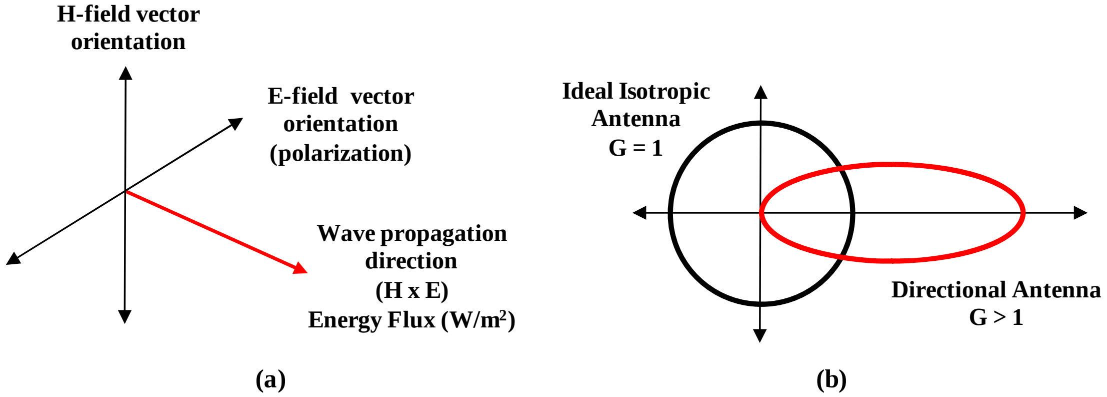

H-field vector orientation
E-field vector orientation (polarization)
Wave propagation direction (H x E) Energy Flux (W/m²)
(a)
Ideal Isotropic Antenna G = 1
Directional Antenna G > 1
(b)

Figure 7.2: Concepts of polarization and gain

7.2 ANTENNA CHARACTERIZATION DATA COLLECTION AND ANALYSIS

In this section, the experimental approach and the results of characterizing six different types of antennas to assess their suitability to support a BT-based travel time data collection system are discussed. Antennas were tested using only one of the two DCUs that were installed in Salem, OR.

Available traffic count data was used to facilitate the evaluation of the performance of the antennas. In each lane adjacent to the BT-based DCU, inductive loop detectors were present and collecting traffic count data (counts for every 15 minute period). The DCU equipped with a specific antenna collected MAC addresses for a period of between three and seven days. The test order of the six antennas tested was randomized. The antenna performance measure computed was the total number of unique MAC addresses read divided by the total traffic volume over the same time period. This performance measure will be referred to as the *fraction read*. It was assumed that the fraction of vehicles with active BT-enabled devices present was the same over the periods that each antenna was tested. This assumption was validated by comparing the performance of the same antenna for non-overlapping time periods.

In addition to the *fraction read*, the *average number of reads per MAC address* was computed as the total number of MAC addresses read over the test period (including multiple reads of the same address) divided by the total number of unique MAC addresses read. This measure is indicative of the total volume of data collected, and a higher *fraction read* measure combined with a relatively low reads per MAC address measure seems ideal.

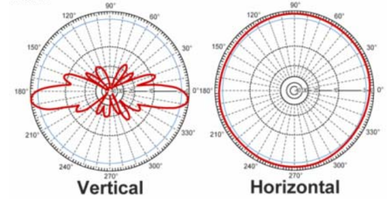
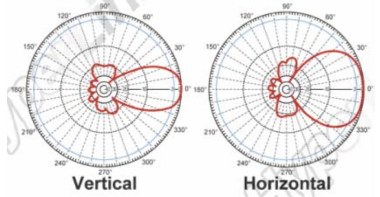
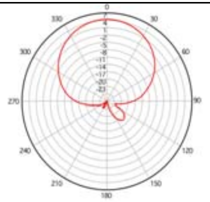
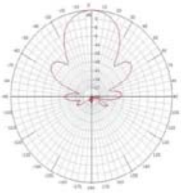
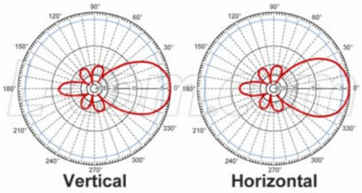
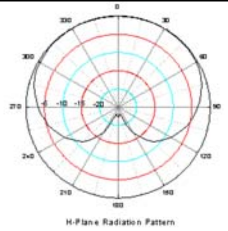
Six different antennas with different polarization and gain characteristics were selected for testing. Table 7.1 presents the most important features of each antenna.

Table 7.1: Main features of the antennas tested

	Antenna					
	1	**2**	**3**	**4**	**5**	**6**
Type	Omnidirectional	Dual Polarization	Directional Circular Polarization	Directional Circular Polarization	Directional Linear Polarization	Directional Wide Pattern
Frequency Range (MHz)	2400-2500	2400-2500	2400-2500	2400-2485	2400-2483	2400-2483
Polarization	Vertical	Both vertical and horizontal	Right Hand Circular	Right Hand Circular	Vertical	Vertical
Maximum Gain (dBi)	9	11	7	12	12	8.6

Table 7.2 shows a picture of each antenna as well as the vertical and horizontal cross-sectional coverage patterns they produce according to the manufacturers’ specifications.

Table 7.2: Antenna pictures and coverage patterns

Ant.	Picture	Antenna Coverage Patterns	Manufacturer
1			L-com, Inc., Antenna model #: HG2409UDT-PRO, http://www.l-com.com
2			L-com, Inc., Antenna model #: RE11DP, http://www.l-com.com
3			Luxul Wireless, Antenna model #: RE11DP, http://www.luxulwireless.com
4			Laird Technologies, Antenna model #: CP24-12, http://www.lairdtech.com
5			L-com, Inc., Antenna model #: HG2412SY, http://www.l-com.com
6			Superpass™, Antenna model #: SPDG13H22, http://www.superpass.com

7.2.1 Test Location

The characterization of the six different antennas was conducted on Oregon Route 221 (Wallace Road NW) on the west side of the city of Salem, Oregon. The Oregon Department of Transportation (ODOT) Intelligent Transportation Systems (ITS) Unit operates a test site on this highway located at latitude-longitude of N 44.972858 and W -123.066321 (see Figure 7.3). The test site is equipped with inductive loop detectors that provide traffic flow volume data to facilitate the assessment of the performance of other technologies that the ODOT ITS Unit evaluates on the site.

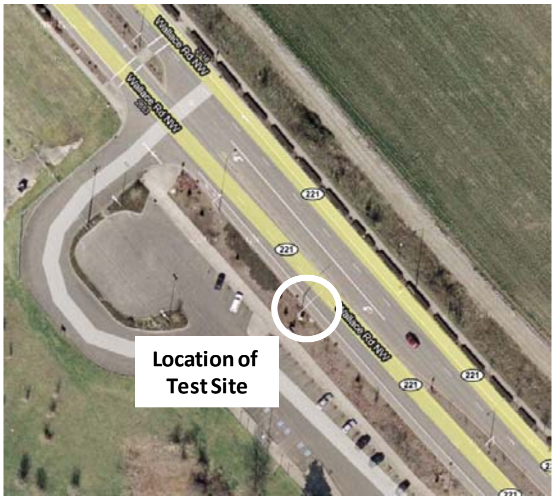


Figure 7.3 Test site on Oregon Route 221

The DCU and the antenna under evaluation were assembled into an enclosure. The enclosure was then mounted onto the signal mast located at the test site. Figure 7.4a depicts a view of the signal mast and shows the approximate height at which the enclosure was mounted (i.e., 8 feet). Figure 7.4b depicts the DCU inside the enclosure and Figure 7.4c depicts antenna #3 mounted on the enclosure.

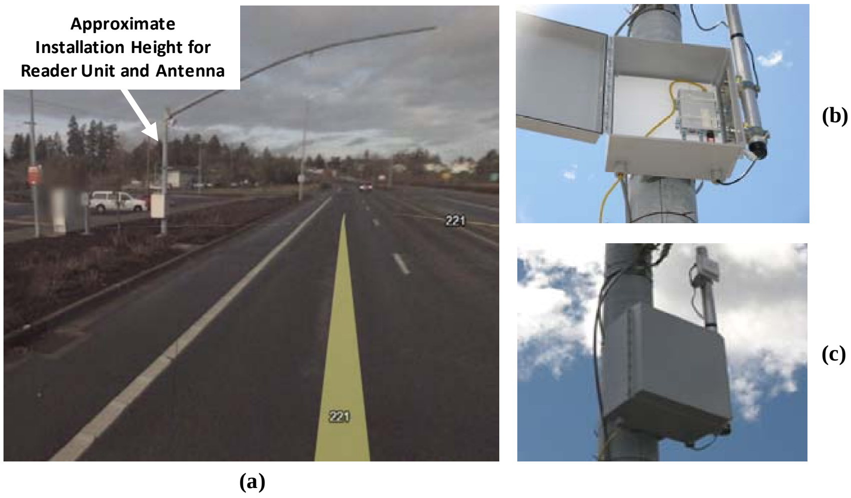


Figure 7.4: Enclosure and antenna installation on Oregon Route 221

7.2.2 Results

The test results indicated that the quantity and the quality (e.g., percentage of duplicates) of the MAC addresses read with each antenna type were different. Table 7.3 presents six metrics calculated for each antenna, i.e., the fraction read, the average number of reads per MAC address, the total traffic volume during the testing period, the average traffic volume during the testing period (i.e., traffic flow volume data divided by the number of days the antenna was tested), the total number of hours the antenna was tested, and whether or not the testing period included the weekend (i.e., Saturday and Sunday). The data shows that traffic volume along the test corridor is very consistent throughout the week.

Table 7.3: Fraction read performance from the antenna tests

	Antenna					
	1	**2**	**3**	**4**	**5**	**6**
Type	Omnidirectional	Dual Polarization	Directional Circular Polarization	Directional Circular Polarization	Directional Linear Polarization	Directional Wide Pattern
Fraction Read	0.109	0.082	0.085	0.090	0.097	0.090
Average Reads Per MAC Address	6.8	3.4	4.0	7.1	8.3	8.2
Total Traffic Volume	85,936	39,374	58,438	54,645	89,280	48,189
Average Traffic Volume	10,742	11,672	11,688	10,929	11,032	10,219
Hours Tested	168.25	70.25	110.5	98.0	167.25	95.25
Included Weekend?	Yes	No	Yes	No	Yes	Yes

As mentioned before, the number of unique MAC addresses collected during a testing period was divided by the traffic volume collected over the same time period via inductive loop detectors to obtain the fraction read performance measure. Computed in this manner, the fraction read measure will underestimate the actual proportion of vehicles from which a MAC address is read since the volume may include vehicles (with the same BT device) traveling in opposite directions past the reader.

It is very likely that some of the MAC addresses read over a test period are not from BT devices in vehicles traveling past the readers. In some cases, the same MAC address was read thousands of times within a test period. To account for this, a simple rule was used to “filter” out these MAC addresses. The data presented in Table 7.3 excludes those MAC addresses that were read more than 100 times over the test period. MAC addresses whose total count was more than 100 most likely were collected from devices located inside nearby apartments/houses or from pedestrians, cyclists, or parked vehicles in the vicinity of the BT reader unit. The value of 100 as a filtering threshold was based on “expert judgment,” and was applied consistently.

A Chi-squared test was conducted to test the null hypothesis that the performance measure *fraction read* was the same for all antennas tested. The null hypothesis was rejected with a p-value of close to zero, which indicates that differences in antenna performance did exist. The Marascuillo procedure was applied to examine significant differences with respect to the performance measure *fraction read* at a 95% confidence level between all pairs of antennas. The results are shown in Table 7.4 where a “1” in a cell indicates a significant difference in the *fraction read* between the two antennas being compared. The antennas in Table 7.4 are ordered from highest to lowest with respect to *fraction read*.

Table 7.4: Pair-wise comparison of antenna with respect to fraction read performance

Antenna #	1	5	4	6	3	2
1	0	1	1	1	1	1
5		0	1	1	1	1
4			0	0	0	1
6				0	0	1
3					0	0
2						0

Antenna 1 (Omnidirectional) had a significantly higher *fraction read* than any other antenna. Antenna 5, with the second highest fraction read, was significantly better than all antennas except antenna 1. There was about equal performance among antennas 4, 6, and 3, with antenna 2 having the worst overall performance. The two best performing antennas were both vertically polarized antennas, but differed in gain and the shape of their coverage pattern.

The average number of reads per MAC address varied widely among the antennas. There is no clear correlation between this measure and the polarization and gain characteristics of the antennas tested. However, the omnidirectional antenna, which had the best overall *fraction read* performance, had the third lowest average reads per MAC address. Thus, it does not appear that a better *fraction read* measure will necessarily come with an increase in data volume.

To better understand the behavior of a DCU with respect to the *fraction read* performance measure, additional data analysis was conducted. For the purposes of modeling and data analysis, it is important to understand the characteristics of the *fraction read* probability distribution. The probability that a MAC address is read from a passing vehicle depends on the probability that the vehicle contains an active BT device, and the probability that the reader reads the devices MAC address during device inquiry in the limited time the vehicle is in the antenna coverage area. If these probabilities are consistent over time and reading of MAC addresses from different DCUs is independent, then the number of MAC addresses read from a given number of vehicles will be binomially distributed. If it is assumed that the number of MAC addresses read is binomially distributed, analyses and plots of the data should produce results consistent with this assumption. Figure 7.5 shows a plot of the fraction read for a 15 minute time period versus the traffic count for that same 15 minute time period for antenna 5. Assuming the overall fraction read is equal to the probability that a MAC address is read from a passing vehicle, the dotted black lines in Figure 7.5 represent +/- two standard deviations from the expected fraction read.

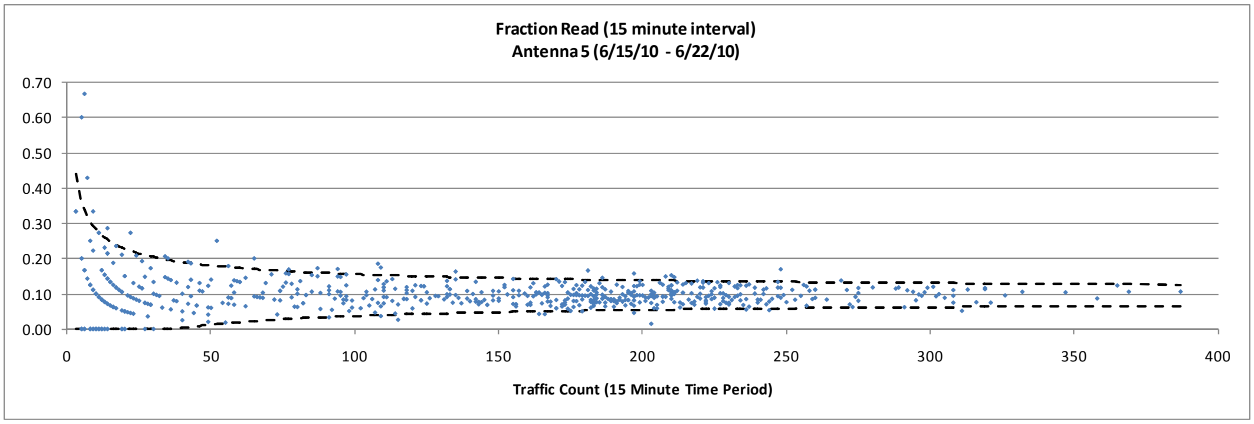


Figure 7.5: Fraction read versus traffic volume for antenna 1 (omnidirectional)

Approximately 94% of the fraction read values plotted in Figure 7.5 fall within +/- two standard deviations from the expected fraction read. Under the binomial assumption, approximately 95% of the fraction read should fall within +/- two standard deviations from the expected fraction read. Other antennas produced similar results with slightly lower percentages within +/- two standard deviations from the expected fraction read except antenna 6, which displayed much higher variability over time.

One critical assumption made in this analysis is that the fraction of vehicles with active BT-enabled devices is constant. To provide evidence that the fraction of vehicles containing active BT-enabled devices does not fluctuate significantly over time, Chi-squared tests were applied to test the null hypothesis that the fraction read is the same for non-overlapping 24-hour time periods using the same antenna. Since the antenna type stays the same, differences in the performance measure *fraction read* may be attributed to differences in the fraction of BT-enabled devices in vehicles passing the reader. This null hypothesis was clearly rejected for antenna 6 (which displayed much higher variability over time). The p-value for this statistical test for antenna 1 was 0.033, and was greater than 0.05 for all other antennas except antenna 6 (i.e., the null hypothesis for antenna 1 would be rejected at a 95% confidence level but not at a 97% confidence level, and the null hypothesis would not be rejected at a 95% confidence level for antennas 2, 3, 4, and 5). The null hypothesis was clearly rejected for antenna 6, which displayed highly variable performance.

7.2.3 Conclusions

Based on the results obtained from characterizing six different types of antennas to assess their suitability to support a BT-based travel time data collection system, the following can be stated:

- Antennas 1 and 5 were the best performers with respect to the performance measure *fraction read*. Both antennas have vertical polarization.

- Antenna 1, the best performer with respect to the performance measure *fraction read*, was ranked third with respect to the *average reads per MAC address*. It was anticipated that an omnidirectional antenna would collect a higher percentage of "redundant" MAC addresses than those with more directionality, but this was not the case.
- Antenna 6, also a vertically polarized, performed very inconsistently when compared to the other vertically polarized antennas in the test (i.e., antennas 1 and 5). This result may be attributed to the way the antenna was mounted on the enclosure during the test. Although the antenna was mounted according to the manufacturer's recommendations, a closer examination of the horizontal cross-sectional coverage pattern along with the results obtained, suggest that mounting it with a slight downward tilt may translate in better results. Restrictions in the project schedule did not allow for a re-test of this antenna.
- The final ranking of the antennas with respect to the performance measure *fraction read* is somewhat counterintuitive. The research team expected the antennas with circular polarization (i.e., antennas 3 and 4) to be the top performers due to the large number of potential orientations in which BT-enabled devices inside moving vehicles can be presented to the reader unit. However, antennas 1 and 5 were the best performers and both have vertical polarization.

In summary, vertically polarized antennas with gains between 9 and 12 dBi are good candidates to support a BT-based travel time data collection system. Antennas with circular polarization do not seem to improve the performance, despite the lack of control regarding the orientation of BT-enabled devices in most applications.

7.3 COLLECTION AND ANALYSIS OF TRAVEL TIME SAMPLES

In this section, the experimental approach and the results of collecting travel time samples with different types of antennas are discussed. Five antennas were tested using two DCUs installed along Oregon Route 221 (Wallace Road NW) on the west side of the city of Salem, Oregon. The reason only five antennas were included in this test (as opposed to the six antenna types shown in Table 7.1) was because the manufacturer of antenna 3 (see Table 7.2Table 7.2) was out of stock. It is important to note, however, that antenna 3 was the second worst performer with respect to the fraction read, as shown in Table 7.3. The research team did not expect a change in performance of this antenna with respect to the collection of travel time samples.

7.3.1 Test Location and Experimental Setup

As with the antenna characterization test described in section 7.2, available traffic count data was used to facilitate the evaluation of the performance of the antennas (counts for every 15 minute period). However, traffic count data was only available at the location where DCU1 was installed. Since DCU2 was located only ¾ of a mile southeast from DCU1, it was assumed that the traffic count data observed at DCU1 would be similar to that observed at the DCU2 installation site. The locations of DCU1 and DCU2 are depicted in Figure 7.6.

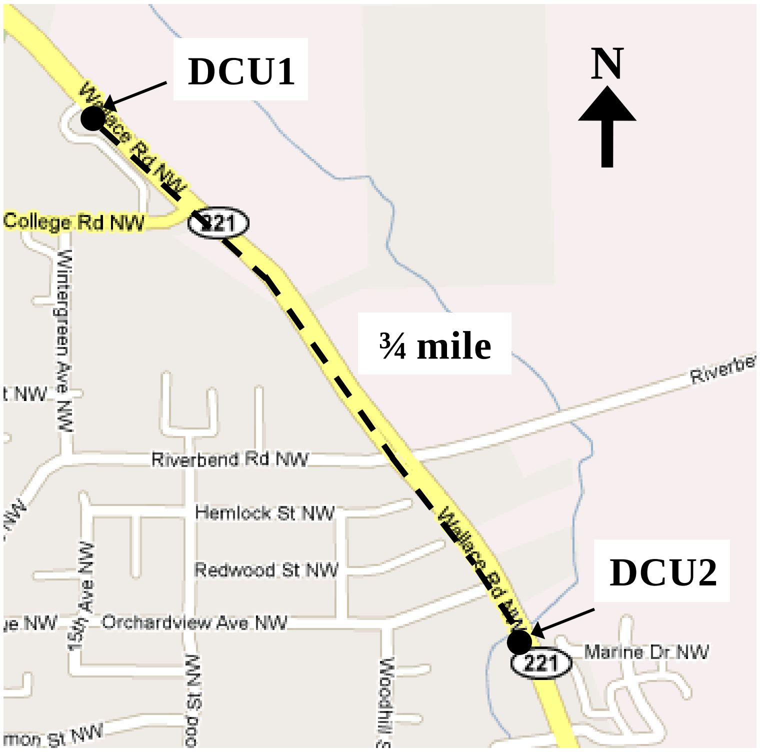
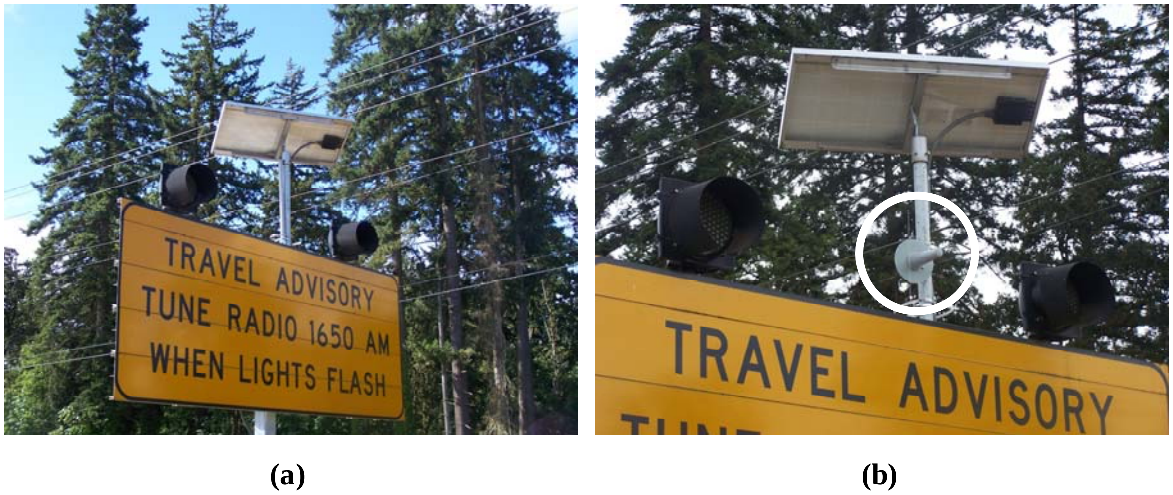


Figure 7.6: Location of DCU1 and DCU2 on Wallace Rd NW

The DCU2 was installed inside a control box located behind the beacon sign. The antennas were mounted at the top of the beacon sign. The beacon sign provided power to the DCU2 and remote connectivity was available via a cellular router. An example of a setup is depicted in Figure 7.7a. The antenna is shown inside the white circle in Figure 7.7b. The antenna was mounted approximately eight feet from the ground. However, since the beacon sign is installed on a slope, the height of the antenna at this site was approximately six feet relative to Wallace Rd NW.



(a) **(b)**

Figure 7.7: Setup for DCU2: (a) Beacon sign, (b) Installation location for antenna during testing

7.3.2 Collection and Processing of the Experimental Data

Both DCUs were equipped with the same antenna type and collected MAC addresses for two separate periods of between three and seven days. Some antennas were tested more days than others either because they were installed right before a weekend or because ODOT personnel were not available to assist the research team with antenna installation. The order of the tests was randomized. During the same periods that the antennas were tested, probe vehicle runs were also conducted. The probe vehicles runs were used to assess the accuracy of the travel time estimates computed from the data collected by the two DCUs. The probe vehicle contained an active BT device with a known MAC address and a clock synchronized with the DCUs. The time that the probe vehicle passed each DCU was recorded. A line drawn from the DCU and perpendicular to the road was used as the location where the vehicle passed the DCU.

For each time period that a specific antenna was tested, the following performance measures were computed:

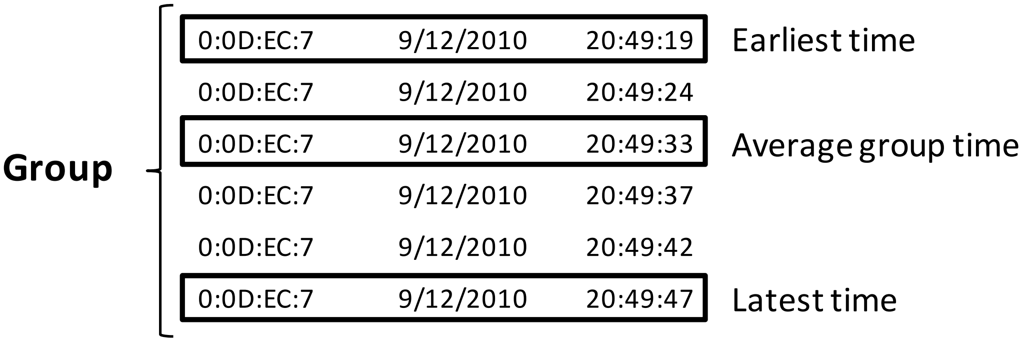
- Traffic volume.
- The number of travel time samples computed from the collected data.
- The percentage of travel time samples obtained with respect to traffic volume.
- The average *relative* percent difference in travel times (between DCU1 and DCU2) between the probe vehicle and the travel times computed for the probe vehicle from collected MAC address data.
- The average *absolute* percent difference in travel times (between DCU1 and DCU2) between the probe vehicle and the travel times computed for the probe vehicle from collected MAC address data.

7.3.3 Computing Travel Time Samples from MAC Address Data

To compute travel time samples from collected MAC address data, a computer program written in Visual Basic for Applications (VBA) in Microsoft Excel was developed. The pseudocode for this application can be found in Appendix D. The general procedure implemented in the code consisted of the following steps:

1. Identify all MAC addresses detected by each DCU.
2. Eliminate those MAC addresses not detected by both DCUs.
3. For data from a single DCU, organize the MAC address data into “groups.”
4. Compute travel time samples from the groups of MAC addresses for each DCU.

A *group* is defined as a collection of data records with the same MAC address sorted sequentially by time, where the time between any adjacent records is no greater than a fixed threshold. Groups of MAC addresses are defined for individual readers and they represent the number of times a single MAC address was captured as a vehicle traveled through the DCU’s antenna coverage area. For this analysis, 30 seconds was the time threshold defining a group. This value is less than the smallest reasonable travel time between the DCUs and greater than the time a vehicle traveling at the speed limit would be in the coverage area of an antenna. Three times were obtained for each MAC address group. These were the earliest, latest, and average of the earliest and latest times in the group (referred to as the average group time). Examples of these times are depicted in Figure 7.8. It is possible (and was seen frequently) for a group to consist of a single record. In this case, the earliest, latest, and average of the earliest and latest times are all the same time.



0:0D:EC:7	9/12/2010	20:49:19
0:0D:EC:7	9/12/2010	20:49:24
0:0D:EC:7	9/12/2010	20:49:33
0:0D:EC:7	9/12/2010	20:49:37
0:0D:EC:7	9/12/2010	20:49:42
0:0D:EC:7	9/12/2010	20:49:47

Figure 7.8: Examples of earliest, latest and average group times for a MAC address in a group

To compute travel time samples from groups of MAC addresses, a second threshold value of 2 minutes was utilized. This threshold value was the maximum time difference between the average group times. The 2-minute threshold worked well since there was little or no congestion at the test site and 2 minutes is the approximate time to complete a round trip passing both DCUs twice. Groups that exceeded this threshold were not used to compute travel time samples. This prevented the generation of extremely large travel time samples from groups of the same MAC address occurring on different days, or on different trips for the same vehicle in the same day.

7.3.4 Results

Results for the percentage of the traffic volume for which a travel time sample was obtained (i.e., the *sampling rate*) are shown in Table **7.5**. With the exception of the Yagi antenna, this percentage was very consistent for the same antenna type over different time periods, which indicates that uncontrollable factors such as climatic conditions and interference from unknown sources had little impact on antenna performance. The performance of the Yagi antenna is sensitive to orientation and inconsistencies in this installation parameter may have contributed to the differences seen over different time periods.

The results indicate that the 180 degree antenna is clearly the best performing antenna with respect to the generating travel time samples. The Yagi also performed well, but showed inconsistencies over time periods which are not well understood.

Table 7.5: Antenna performance – sampling rate

	180 Degree		Omni		Dual		Circular		Yagi	
	8/17 - 8/20	**7/27 - 7/30**	**7/30 - 8/3**	**8/27 - 8/31**	**8/8 - 8/10**	**8/20 - 8/24**	**8/31 - 9/3**	**9/7 - 9/10**	**8/24 - 8/27**	**9/10 - 9/15**
Traffic Volume (TV)	40,164	40,955	49,995	48,177	24,416	48,610	53,148	40,047	39,278	65,371
Number of Travel Time Samples (TTS)	3,837	3,916	3,839	3,693	1,398	2,781	4,125	2,847	3,898	4,987
TTS as Percentage of TV	9.55%	9.56%	7.68%	7.67%	5.73%	5.72%	7.76%	7.11%	9.92%	7.63%
Weighted Avg. % Match	9.56%		7.67%		5.72%		7.48%		8.49%	
Number of unique MAC addresses detected by DCU1	4,046	4,506	4,420	4,496	1,963	3,619	4,872	3,462	3,423	4,899

While a high sampling rate value is desirable, the accuracy of the collected travel time samples is perhaps a more critical performance measure. When computing travel time samples from groups of MAC addresses, different methods for computing samples exist that differ in what time stamp in a group of MAC address is utilized. The following five methods were examined.

1. DCU1 Group Average Time – DCU2 Group Average Time
2. DCU1 Group First Time – DCU2 Group First Time
3. DCU1 Group First Time – DCU2 Group Last Time
4. DCU1 Group Last Time – DCU2 Group First Time
5. DCU1 Group Last Time – DCU2 Group Last Time

Table **7.6** shows the accuracy results of travel time samples obtained using the most accurate method for an antenna type during its corresponding testing time period. Accuracy was determined by comparing the estimated travel time derived from MAC addresses to the average travel times recorded by the probe vehicle for each antenna during the testing period. Both the absolute and relative errors (shaded rows in Table 7.6) were computed as the percent difference from the manually recorded probe vehicle travel times.

The most accurate method for computing travel time samples was to utilize the difference in *average* group times. The other methods generated much larger average errors. The lowest error percentage was generated by the dual polarized antenna. This antenna also had the lowest sampling rate.

Figure 7.9 shows that a trade-off exists between the sampling rate and the accuracy of the travel time samples collected. This is most likely directly related to the road coverage area realized by the different antenna types.

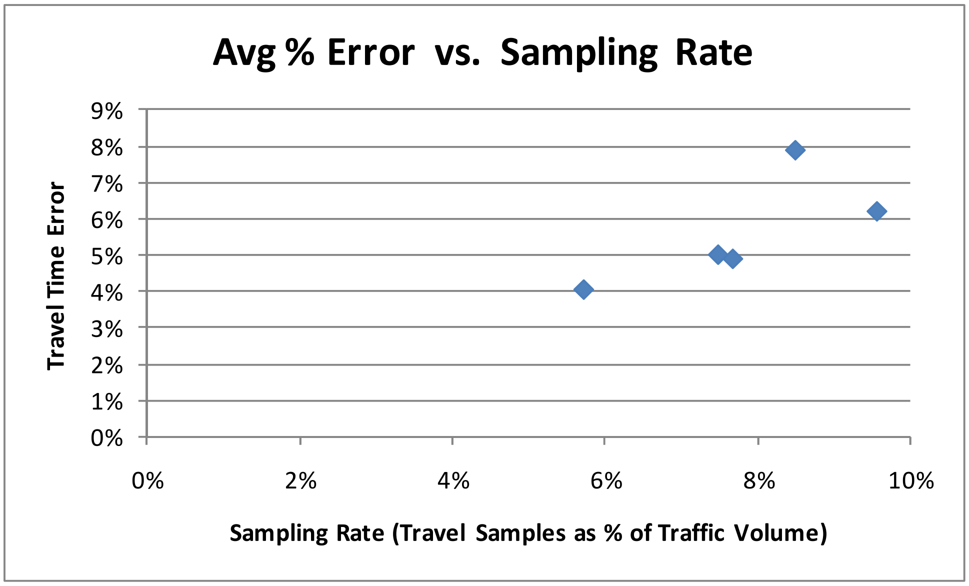


Figure 7.9: Trade-off between the accuracy of travel time samples and sampling rate

Table 7.6: Antenna performance – accuracy of travel time samples

	180 Degree		Omni		Dual		Circular		Yagi	
	8/17 - 8/20	7/27 - 7/30	7/30 - 8/3	8/27 - 8/31	8/8 - 8/10	8/20 - 8/24	8/31 - 9/3	9/7 - 9/10	8/24 - 8/27	9/10 - 9/15
Calculation Method	**Avg-Avg**	**Avg-Avg**	**Avg-Avg**	**Avg-Avg**	**First-First**	**Avg-Avg**	**Avg-Avg**	**Last-Last**	**Avg-Avg**	**Avg-Avg**
Absolute Average Error	6.38%	6.06%	3.26%	6.61%	4.02%	4.08%	4.38%	5.88%	8.05%	7.82%
Maximum	15.79%	16.42%	8.16%	13.28%	11.32%	7.94%	9.84%	15.29%	21.05%	16.45%
Minimum	0.00%	0.00%	0.00%	1.85%	0.00%	0.00%	0.00%	0.00%	1.27%	0.94%
Standard Deviation	5.25%	5.55%	2.51%	3.69%	3.25%	2.96%	3.53%	5.15%	6.74%	5.24%
Combined Absolute Avg Error	6.22%		4.91%		4.06%		5.02%		7.91%	
Relative Average Error	0.08%	-3.63%	-1.21%	-7.08%	0.17%	-2.86%	-3.22%	-6.02%	-1.10%	-7.68%
Standard Deviation	8.20%	8.03%	4.13%	3.83%	5.38%	4.27%	4.95%	4.51%	11.67%	6.64%

7.3.5 Conclusions

The results of utilizing different antennas to read MAC addresses at two locations and then compute travel time samples from this data indicate that a trade-off exists between the number of samples obtained and the accuracy of these travel time samples. This trade-off is most likely the result of differences in road coverage areas provided by the different antenna types. A large area generates large groups of MAC addresses which contain multiple reads of the same MAC address while a vehicle is moving in the antenna coverage area. It is not apparent from the data which of the records best represents the point at which the vehicle is closest to the DCU. This highlights the importance of developing methods to filter MAC address data effectively. If this can be performed effectively, then ideally only the MAC address records that represent the time a vehicle is the closest to the DCU will be used to compute travel times.

The results also indicate that different antenna types will be suited to different uses of the DCUs. If the main focus is the collection of travel time data, as in this research, a lower sampling rate (smaller coverage area) combined with more accurate travel time samples may be desired. If the main focus of utilizing the DCU is to measure intersection performance, then an antenna with a higher sampling rate and larger coverage may be desired.

8.0 CONCLUSIONS

This project pursued several objectives conducive to the implementation and testing of a Bluetooth (BT) based system to collect travel time data. The first objective was to make improvements to an existing BT-based data collection unit (DCU) developed on an earlier project funded by ODOT's ITS Unit. The second objective was to deploy a BT-based travel time data collection system to perform comprehensive testing on all the components. The third objective was to develop and test software applications to collect, process, and store time-stamped MAC addresses. Finally, the fourth objective involved the development of functional requirements and technical specifications for the DCU and the preparation of a user's manual with instructions on how to assemble, configure, and troubleshoot the DCU.

The major accomplishments and main findings of this project were as follows:

- The hardware platform of the original DCU was very reliable, so the focus in this area was to improve the efficiency of the Linux script that collects time-stamped media access control (MAC) addresses from BT-enabled devices in vehicles. The improvements made included the use of a more efficient command to collect MAC addresses, the development of an algorithm to check for duplicate MAC addresses, and the removal of unnecessary lines and variables in the code. Additionally, procedures for the collection, processing and storage of time-stamped MAC addresses were specifically defined to protect citizens' privacy.
- Two different BT-based travel time data collection systems were installed. The first system, composed of two DCUs, was installed on a corridor located in Salem, OR. Extensive testing was done on this system, including the collection of travel time samples. A second system composed of five DCUs was installed along 99W in the city of Tigard, OR. Very limited data collection was done on 99W due to the lack of network connectivity.
- A good match between the DCU and antenna is critical in a BT-based travel time data collection system. Since the orientation of the BT-enabled devices in vehicles cannot be controlled, intuition may dictate that antennas with circular polarization would perform better. However, the results of the antenna characterization tests revealed that vertically polarized antennas with gains between 9 and 12 dBi have very good performance. Antennas with circular polarization did not seem to improve performance, despite the lack of control regarding the orientation of BT-enabled devices in most applications.
- The two-DCU system was used to collect travel time samples. The results indicated that the 180 degree antenna is clearly the best performing antenna with respect to the generating travel time samples. The Yagi also performed well, but showed inconsistencies over time periods which are not well understood. While a high sampling rate value is desirable, the accuracy of the collected travel time samples is perhaps a more critical performance measure. The most accurate method for computing travel time samples was to utilize the difference in *average* group times.

- Three server-based applications were developed in collaboration with ODOT’s ITS Unit. The first application collects a text file from DCU containing individual records of time-stamped MAC addresses collected from BT-enabled devices in vehicles. The second application stores each record in the text file into a Microsoft© Access database. The Microsoft© Access database was the third software component developed. These software components were tested on a limited basis during the antenna characterization tests conducted in Salem, OR.
- The functional requirements and technical specifications for the DCU were updated. A user’s manual was prepared with detailed instructions on how to assemble, configure, and troubleshoot the DCU.

8.1 FUTURE RESEARCH

A new phase of this project is underway and will focus on issues related to the processing and synthesis of data collected from Bluetooth-enabled devices to generate travel time performance measures. A number of issues related to data processing and synthesis need to be addressed, including:

- How data collected by the DCUs should be filtered. Some MAC addresses are read a large number of times on specific days indicating that most of these records may not be used for travel time calculations. Filtering this data before computing travel times will increase computational efficiency.
- Specific procedures for identifying travel time sample data outliers and non-vehicle related data. Under very congested conditions, very slow travel times must be distinguished from non-vehicle travel times.
- The amount, format, and length of time MAC-based address data are stored.
- Development of methods to estimate the precision of travel time performance measures.
- Utilization of MAC-based data to estimate other traffic performance measures at intersections and on highways (e.g., traffic volume).
- Understanding the impact of average vehicle speeds and different traffic states (stationary vs. moving) near an intersection or on a highway, on the MAC addresses recognized by a reader.
- Understanding the effect that antenna and reader design variables have on the precision and accuracy of travel time estimates.
- Evaluation of the impact of pushing data from readers to a central server vs. pulling data from the readers. Examine if pushing data provides significant benefits with respect to providing “real time” travel time information.
- Assessment of the information flow volume assuming a wide-scale implementation of Bluetooth readers.

Additionally, several aspects of the BT-based travel time data collection system are potential areas for further research. For example:

- Additional antenna characterization testing is needed to fully understand the effect that factors such as antenna installation height and antenna orientation have on the accuracy of travel time estimates. Antenna height and orientation change the shape and the size of the antenna's coverage pattern, which in turn affects the number of times a single time-stamped MAC addresses is read by the DCU while the a vehicle is in the antenna's coverage area. The challenge here is to identify the single time-stamped MAC address record that corresponds to the point in time where the vehicle was directly in front of each DCU and use these records to generate a travel time sample.

- Data filtering techniques need to be developed and applied to minimize the amount of data to be processed and to accurately identify outliers. Full scale implementations of a BT-based travel time data collection system could include hundreds (if not thousands) of DCUs. Making the data collection and analysis process as efficient as possible is a must to ease the burden on networking and data storage resources.

- Finally, the use of BT-based technology for measuring the performance of intersections is a promising field. If accurate intersection-related measures could be derived with the collection and processing of time-stamped MAC addresses, this could potentially translate into cost reductions to equip and operate intersections.

9.0 REFERENCES

Andre, T., Shin-Ting, J., & Stephen, R. (2008). ”Design and Initial Implementation of an Inductive Signature-Based Real-Time Traffic Performance Measurement System.” *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems.* Beijing.

Angelo, P.D., Al-Deek, H., and Wang, M.C. (1999). “Travel time prediction for freeway corridors.” *Journal of Transportation Research Record*, 1676, pp. 184–191.

Bertini, R.L., Lasky, M., Monsere, C.M. (2005). “Validating Predicted Rural Corridor Travel Times from an Automated License Plate Recognition System: Oregon’s Frontier Project.” *8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria.

Bremmer, D., Cotton, K.C., Cotey, D., Prestrud, C.E., and Westby, G. (2004). “Measuring Congestion: Learning from Operational Data.” *Journal of Transportation Research Record*, 1895, pp. 188–196.

California DOT (2010). *Bluetooth Travel Time*. Online document. Accessed online July 2010 at: http://www2.dot.ca.gov/newtech/operations/bluetooth_web_page/intro.html.

Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z. (2001). “Freeway Performance Measurement System: Mining Loop Detector Data.” *The 80th Annual Meeting of the Transportation Research Board.* Preprint CDROM, Washington, D.C.

Cheng, H. H. (2001). *Development and Testing of Field-Deployable Real-Time Laser-Based Non-Intrusive Detection.* UC Berkeley, Research Reports, California Partners for Advanced Transit and Highways (PATH). Accessed online July 2010 at: http://escholarship.org/uc/item/56r4492r.

Coifman, B. (2007). *Vehicle Classification from Single Loop Detectors*. Madison, WI: Midwest Regional University Transportation Center. Publication MRUTC 05-02. Accessed online September 2009 at: http://www.ceegs.ohio-state.edu/%7Ecoifman/outgoing/MRUTC070628b.pdf.

Cross, D., Hoeckle, J., Lavine, M., Rubin, J., Snow, K. (2007). “Detecting Non-Discoverable Bluetooth Devices.” *Critical Infrastructure Protection.*

Douma, F. (2009). *The Implications of Current and Emerging Privacy Law for ITS*. Online document. September 10, 2009. Accessed online November 2010 at: http://www.its.umn.edu/Events/SeminarSeries/2009/fall/articles/sept10.html.

Duri, S., Gruteser, M., Liu, X., Moskowitz, P., Perez, R., Singh, M., Tang, J. (2002). "Framework for Security and Privacy in Automotive Telematics." *Proceedings of the 2nd International Workshop on Mobile Commerce*, WMC '02. ACM, New York, NY, pp. 25-32.

Friedrich, M., Jehlicka, P., Schlaich, J. (2008). "Automatic Number Plate Recognition for the Observance of Travel Behavior." *Proceedings of the 8th International Conference on Survey Methods in Transport: Harmonisation and Data Comparability*, Annecy, Frankreich.

FHWA (2006). *Regional ITS Architecture Guidance Document*. Publication FHWA-HOP-06-112. US Department of Transportation, Office of Operations. Washington, DC.

Gibson, D. (2009). *How Loop Detectors Work*. Online document. Accessed online September 2009 at: http://www.splatco.com/tips/loops.htm.

Haghani, A., Hamedi, M., Sadabadi, K.F., Young, S. and Tarnoff, P (2010). "Freeway Travel Time Ground Truth Data Collection Using Bluetooth Sensors." *Transportation Research Record: Journal of the Transportation Research Board*, No. TBD, Transportation Research Board of the National Academies, Washington, D.C.

Haseman, R.J., Wasson, J.S., Bullock, D.M. (2010). "Real Time Measurement of Work Zone Travel Time Delay and Evaluation Metrics Using Bluetooth Probe Tracking." *Transportation Research Record: Journal of the Transportation Research Board*, No. TBD, Transportation Research Board of the National Academies, Washington, D.C.

Herrera, J. C., Work, D. B., Herring, R. B. (2009). *Evaluation of Traffic Data Obtained via GPS-Enabled Mobile Phones: the Mobile Century Field.* Online document. UC Berkeley, UC Berkeley Center for Future Urban Transport. Accessed online September 2009 at: http://www.escholarship.org/uc/item/0sd42014.

Hoffman, D. L., Novak, T. P., Peralta, M. (1999). Building Consumer Trust Online. *Comm. ACM*, Vol. 42, No. 4, pp. 80-85.

Hoh, B., Gruteser, M., Xiong, H., Alrabady, A. (2006). "Enhancing Security and Privacy in Traffc-Monitoring Systems." *IEEE Pervasive Computing Magazine* (Special Issue on Intelligent Transportation Systems), Vol. 5, No. 4.

Hoh, B., Gruteser, M., Xiong, H., Alrabady, A. (2007). "Preserving Privacy in GPS Traces via Density-Aware Path Cloaking." *ACM Conference on Computer and Communications Security (CCS).*

Ituner Networks Corporation. *ALIX WiFi*. Online document. Accessed online July 2010 at: http://www.mini-box.com/ALIX.

Kwong, K., & Kavaler, R. (2009). Arterial Travel Time Estimation Based on Vehicle Re-Identification Using Wireless Magnetic Sensors. *Transportation Research Part C: Emerging Technologies*. Vol 17, No. 6, December 2009, pp. 586-606

Minch, P.R. (2004). “Privacy Issues in Location-Aware Mobile Devices.” *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 10.

Munro, K. (2008). “Breaking into Bluetooth.” *Network Security*, Vol. 2008, No. 6, pp. 4-6.

Nikolas, G., & Alexander, S. (2006). “Real Time Vehicle Identification and Performance Measures on Signalized Arterials.” *IEEE Intelligent Transportation Systems Conference*. Toronto.

Ogden, K. W. (2001). “Privacy Issues in Electronic Toll Collection. *Transportation Research Part C: Emerging Technologies*, Vol. 9, No. 2, Implications of New Information Technology, pp. 123-134.

Oh, J.S., Jayakrishnan, R., Recker, W. (2002). *Section Travel Time Estimation from Point Detection Data*. Irvine, CA: Institute of Transportation Studies. Publication UCI-ITS-TS-WP-02-14. Accessed online September 2009 at: http://www.its.uci.edu/its/publications/papers/CTSS/UCI-ITS-TS-WP-02-14.pdf.

Palen, J., Coifman, B., Sun, C., Ritchie, S., and Varaiya, P. (2000). “California Partners for Advanced Transit and Highways (PATH) Enhanced Loop-Based Traffic Surveillance Program.” *ITS Quarterly*, Fall, pp. 17-25.

Riley P. F. (2008). “The Tolls of Privacy: An Underestimated Roadblock for Electronic Toll Collection Usage.” *Computer Law & Security Report*, Vol. 24, No. 6, pp. 521-528.

Porter, J.D., Kim, D.S., Magaña, M.E. (2009). *Performance Monitoring for Transportation System Management and Operations*. Unpublished manuscript. Oregon Department of Transportation, Salem, OR.

Quayle, S.M., Koonce, P., DePencier, D., Bullock, D.M. (2010). “Arterial Performance Measures Using MAC Readers: Portland Pilot Study.” *Transportation Research Record: Journal of the Transportation Research Board*, No. TBD, Transportation Research Board of the National Academies, Washington, D.C.

Sena Technologies, Inc. *Parani UD100*. Online document. Accessed Online July 2010 at: http://www.sena.com/products/industrial_bluetooth/ud100.php.

Stallings, W. (2005). *Wireless Communications & Networks*. Pearson Education, Inc., Upper Saddle, NJ.

Traffax, Inc. *Traffax Inc.* Online document. Accessed online July 2010 at: http://www.traffaxinc.com.

Ulrich, K.T., Eppinger, S.D. (1995). *Product Design and Development,* McGraw-Hill Inc.

Vizmuller, P. (1995). *RF Design Guide: Systems, Circuits, and Equations*. Artech House, Inc., Boston, MA.

Wasson, J. S., Sturdevant, J.R., Bullock, D.M. (2008). "Real-Time Travel Time Estimates Using MAC Address Matching." *Institute of Transportation Engineers Journal*, ITE, Vol. 78, No. 6, pp. 20-23.

Wong, F., Stajano, F. (2005). "Location Privacy in Bluetooth." *In Proceedings of 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS '05),* Lecture Notes in Computer Science, LNCS 3813, pp. 176-188. Springer-Verlag.

Wright, J., & Joy, D. (2001). *Using Vehicles Equipped with Toll Tags as Probes for Providing Travel Times*. UC Berkeley, Working Papers, California Partners for Advanced Transit and Highways (PATH). Accessed online July 2010 at: http://escholarship.org/uc/item/9f17h2j0.

Zhang, X., Wang, Y.,Nihan, N.L. and Dong, H. (2005). "Monitoring a freeway network in real time using single-loop detectors," *Int. J. Vehicle Information and Communication Systems*, Vol. 1, No. 1/2, pp.119–130

APPENDIX A:
UPDATED FUNCTIONAL AND TECHNICAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

The outcomes of *Task #7: Update the functional and technical requirements document and development of a users' manual* are documented in this appendix. The updated version of the functional requirements and technical specifications are presented first. Next, the different aspects related to the assembling, configuration and troubleshooting of the DCU are presented in the form of a user's manual.

Functional requirements are the various functions that must be executed by a system and its components in order to successfully produce a desired output. Functional requirements are typically defined at a high level and do not detail how these said functions will be implemented (Ulrich & Eppinger, 1995). Technical specifications, on the other hand, are measurable parameters that describe how various functions must be performed by a component or system.

In the context of this project, an example of a functional requirement would be that the DCU *shall capture MAC addresses from Bluetooth-enabled devices often enough so as to allow for the estimation of accurate travel times*. A technical specification for this function would be that the DCU *shall perform a scan to capture MAC addresses from Bluetooth-enabled devices every 0.5 seconds*.

The functional requirements that the Bluetooth-based travel time data collection system needs to fulfill are presented using three different perspectives. These perspectives are the DCU, the database management system (DBMS) and privacy.

FUNCTIONAL REQUIREMENTS OF THE DATA COLLECTION UNIT

The data collection unit (DCU) is the most important component of the Bluetooth-based travel time data collection system. The DCU is responsible for collecting time-stamped MAC addresses from Bluetooth-enabled devices, temporarily storing these MAC addresses in a file, and interfacing with a remote host to enable the transfer of these data so that accurate travel times can be estimated. The functional requirements for the DCU are divided into three major categories: *structural, data collection* and *data communications*.

Structural

From a *structural* perspective, the DCU shall:

- Be capable of operating in diverse environments and under a full spectrum of climatic conditions (e.g., high and low temperatures).
- Protect internal components by means of an enclosure to minimize potential damage from the accumulation of dust or moisture.
- Be small enough to fit inside a standard traffic controller cabinet
- Have accessible connectors for power, network communications and an antenna.

Data Collection

From a *data collection* perspective, the DCU shall:

- Be capable of accurately collecting MAC addresses wirelessly from Bluetooth enabled portable devices such as cell phones, personal digital assistants (PDAs), and laptops, as well as from vehicular devices such as GPS navigation systems.
- Be capable of accurately collecting MAC addresses wirelessly from Bluetooth enabled portable devices under a full spectrum of climatic conditions including sun, rain, snow, ice, humidity, blowing dust or sand, salt water, or sudden, severe changes in ambient air pressure.
- Be capable of producing, for each MAC address collected, a data record comprised of the following elements:
    - MAC address of the reader (key identifier)
    - Unique portion of a Bluetooth enabled device's MAC address
    - Time stamp.
- Be able to store a large quantity of data records in internal memory or a file before these data are extracted from the unit.
- Be capable of reading MAC addresses at different sampling rates.

Data Communications

From a *data communications* perspective, the DCU shall:

- Provide a mechanism (i.e., a combination of firmware and software) for transmitting data.
- Be capable of transmitting/receiving data collected/processed via network communications technology (e.g., Ethernet, frame relay, cellular network, etc.).
- Provide a network interface to perform remote updates and diagnostics.
- Be capable of being accessed via a remote computer to enable firmware/software updates and to monitor performance.

FUNCTIONAL REQUIREMENTS OF THE DATABASE MANAGEMENT SYSTEM

The database management system (DBMS) is the component of the Bluetooth-based travel time data collection system responsible for storing records of time-stamped MAC address collected from the a DCU, filtering these records to extract unique MAC addresses, sorting records, and calculating travel times for specific segments of roads. The functional requirements for the DBMS are defined for the category of *data manipulation and storage*.

Data Manipulation and Storage

From a *data manipulation and storage* perspective, the DBMS shall:

- Be capable of filtering duplicate MAC addresses from the data files extracted from the DCUs.
- Be capable of sorting MAC addresses based on their time stamp.
- Be capable of matching time-stamped MAC addresses that appear at distinct points of travel on the road segment of interest.
- Be capable of storing time-stamped MAC address information in a maintainable database structure.
- Be capable of generating travel time estimates from time-stamped MAC address data.

FUNCTIONAL REQUIREMENTS OF THE SERVER DATA COLLECTION SOFTWARE

The server data collection software is the component of the Bluetooth-based travel time data collection system responsible for connecting remotely to individual DCUs to download the file that stores time-stamped MAC addresses. The functional requirements for the server data collection software are defined for the category of *data communications.*

Data Communications

From a *data communications* perspective, the server data collection software shall:

- Be capable of remotely accessing DCUs to extract data files containing time-stamped MAC address data.

FUNCTIONAL REQUIREMENTS TO PROTECT PRIVACY

Collecting MAC address information is very different from other forms of data collection used to estimate travel times. For example, cell phone tracking and license plate recognition use information that can directly (and easily) be traced to a specific user. Although it can be argued that MAC addresses (especially those captured from Bluetooth-enabled cell phones) could be used for the same purpose, identifying a specific user using a MAC address is not as straightforward.

Despite these facts, public perception about this method of data collection varies widely and could prevent its implementation as a simple and inexpensive way to estimate travel times. Thus, the main objective of defining functional requirements in this area is to minimize (if not eliminate) the possibility of infringing on a user’s right to privacy.

From a *privacy* perspective, the Bluetooth-based travel time data collection system shall:

- Only collect the portion of a MAC address that allows for unique identification of a traveling entity between two data collection points.
- Only store the portion of a MAC address that allows for unique identification of a traveling entity between two data collection points.
- Only temporarily store time stamped MAC addresses for the purpose of estimating travel times.

TECHNICAL REQUIREMENTS

This section describes the technical specifications required for the Bluetooth-based travel time data collection system.

TECHNICAL SPECIFICATIONS FOR THE DATA COLLECTION UNIT

The collection of time-stamped MAC addresses by the DCU shall occur via wireless radio frequency (RF) communications. The DCU controls the communications protocol, reads time-stamped MAC addresses from enabled devices, and ensures data delivery and validity.

RADIO FREQUENCY COMMUNICATIONS VIA BLUETOOTH

The Bluetooth RF (physical layer) in the DCUs shall:

- Operate in the 2.4GHz unlicensed Industrial, Scientific and Medical (ISM) band (i.e., 2400 - 2483.5 MHz).
- Employ a frequency hop transceiver to combat interference and fading, and provide many frequency hopping spread spectrum (FHSS) carriers.
- Utilize a symbol rate of 1 Megasymbol per second (Msps) supporting the bit rate of 1 Megabit per second (Mbps) or, with Enhanced Data Rate, a gross air bit rate of 2 or 3 Mbps. These modes are known as Basic Rate and Enhanced Data Rate respectively.
- Utilize a transmitter rated as “Power Class 1.” Power Class 1 transmitters have a maximum output power (Pmax) of 100 mW (20 dBm) and a minimum output power of 1 mW (0 dBm).

ENVIRONMENTAL OPERATING CONDITIONS

From an environmental operation perspective, the DCU shall:

- Operate under a full spectrum of climatic conditions. It is recommended that they are able to withstand temperatures ranging from -29 to 163 °F and humidity levels from 5 to 95%, non-condensing (*Doug Spencer, ODOT, unpublished data*).
- Operate in industrial environments and in areas with high levels of electromagnetic noise and interference (*Doug Spencer, ODOT, unpublished data*).
- Operate within a range of voltage between 90-130 VAC and 30 Amperes (60 Hz, ± 3 Hz) (*Doug Spencer*, *ODOT*, *unpublished data*).
- Comply with the International Electrotechnical Commission (IEC) standards shown in Table A.1.

Table A.1: International Electrotechnical Commission (IEC) Standards

STANDARD	DESCRIPTION
IEC 60068-1	Environmental testing – Part 1: General and Guidance
IEC 60068-2-6	Environmental testing – Part 2: Test Fc: Vibration (sinusoidal)
IEC 60068-2-27	Environmental testing – Part 2: Test Ea and guidance: Shock

HAZARDOUS ENVIRONMENT

The Bluetooth RF (physical layer) in the DCU shall be certified by Factory Mutual Corporation (FMC) or Underwriters Laboratories (UL) as Intrinsically Safe for operation defined by the National Electrical Code (NEC) for at least Class 1 Division 2, Groups A, B, C, and D. Detailed definitions and specifications for intrinsically safe operation are outlined by the National Fire Prevention Association, as detailed in the NEC, 2008.

RADIO REGULATORY COMPLIANCE

The Bluetooth RF (physical layer) in the DCU shall support non-licensed communication in compliance with the U.S. Department of Commerce National Telecommunication and Information Administration, Manual of Regulation and Procedures for Federal Radio Frequency Management, ANNEX K (non-licensed devices), as well as the U.S. Federal Communications Commission Code of Federal Regulations, Title 47, Part 15, Radio Frequency Devices.

The DCU shall be capable of operating at a frequency and radiated output power that does not require special operational licensing. The Bluetooth RF (physical layer) in the DCU shall also be certified by the Federal Communications Commission (FCC) under U.S. Federal Communications Commission Code of Federal Regulations, Title 47, Part 15, Radio Frequency Devices.

COMPATIBILITY ISSUES

The Bluetooth RF (physical layer) in the DCU, software, cables, connectors, and peripherals shall maintain backward compatibility, so that as technology changes, older technology can still be read.

CONFORMAL COATING

For the purpose of this document, the term conformal coating refers to a type of protective coating used on printed wiring assemblies. Conformal coating is intended to provide protection from moisture and contamination and provide electrical insulation. The DCU should conform to the standards shown in Table A.2 regarding conformal coating.

Table A.2: Conformal coating standards

STANDARD	DESCRIPTION
UL 94	Test for Flammability of Plastic Materials for Parts in Devices and Appliances
ANSI/IPC-CC-830B	Qualification and Performance of Electrical Insulating Compound for Printed Wiring Assemblies

APPENDIX B:
USER’S MANUAL

USER’S MANUAL

The user’s manual includes instructions on how to assemble and configure the DCU to collect MAC addresses from Bluetooth-enabled devices. Troubleshooting procedures are also included to address potential malfunctions that may occur when the unit is deployed in the field. The information has been organized in the following subsections:

- Imaging the Compact Flash Card
- Configuring the ALIX Board to Run the Bluetooth Script
- Assembling the DCU
- Configuring a Cellular Router to Access the DCU Remotely
- Accessing the DCU via a Remote Connection
- Accessing the DCU via a Direct Connection
- Troubleshooting the DCU

IMAGING THE COMPACT FLASH CARD

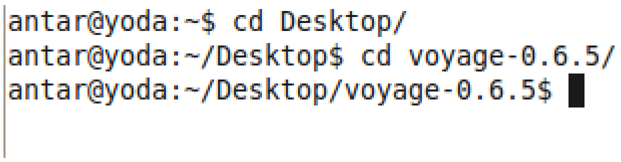
All the software needed for the DCU to function properly, including the operating system and data collection script, is stored in a one (1) gigabyte (GB) compact flash card (CFC). This section outlines the steps required to format and install the software onto the CFC. Two different procedures are described. The procedure described in the section titled “*Installing the Software onto a Blank CFC*” assumes that the software will be installed onto a blank CFC. The procedure described in the section titled “*Cloning the Software Image to a New Compact Flash Card*” assumes that a blank CFC will be “cloned” using a completely formatted CFC.

Installing the Software onto a Blank CFC

This section outlines how to install and configure the software onto a blank CFC. The operating system used on the DCU is Voyage Linux, version 0.6.5. Older versions of Voyage Linux are not compatible with the ALIX board. Voyage Linux can be obtained in the form of “img” files by accessing http://linux.voyage.hk/download. The compressed files with extension “tar.bz” in the live CD must be downloaded.

Once Voyage Linux version 0.6.5 has been downloaded, the detailed steps to install and configure it are as follows:

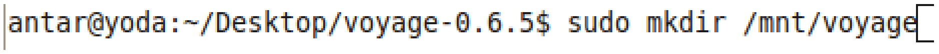
1. Double click on the compressed file voyage-0.6.5.tar.bz2 (or the current version) to see the contents. Extract it to the Desktop.
2. Open a terminal window and locate the file (see Figure B.1).



```
antar@yoda:~$ cd Desktop/
antar@yoda:~/Desktop$ cd voyage-0.6.5/
antar@yoda:~/Desktop/voyage-0.6.5$ █
```

Figure B.1: Changing directory to the Desktop

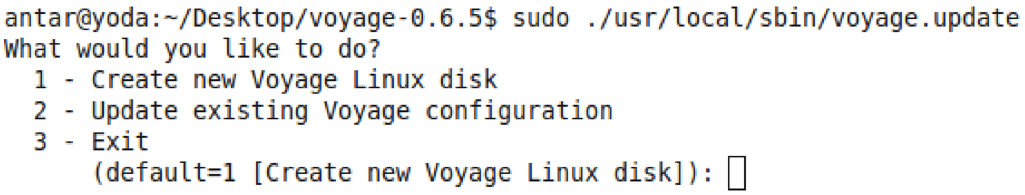
3. Run the command shown in Figure B.2 to create a directory to mount the operating system before transferring it to the CFC.



```
antar@yoda:~/Desktop/voyage-0.6.5$ sudo mkdir /mnt/voyage
```

Figure B.2: Creating a temporary directory for Voyage Linux

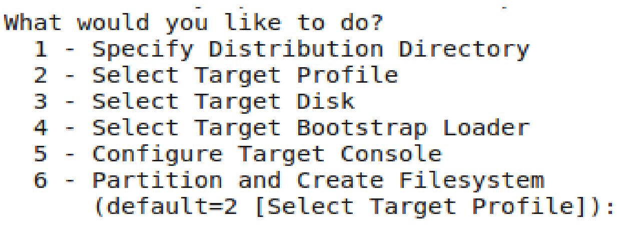
4. Run the command shown in Figure B.3. This command will run the script to install the operating system onto the CFC. Please note that the location where the new CFC is mounted needs to be known. Also, the CFC must have been formatted as a file system of type “`ext3`” (see step 11 in the section titled “*Cloning the Software Image to a New Compact Flash Card*”). The value in brackets (i.e., [Create new Voyage Linux Disk]) is the default value and this will be the selected option if <Enter> is pressed. Press <Enter>, or type 1 and then press <Enter>.



```
antar@yoda:~/Desktop/voyage-0.6.5$ sudo ./usr/local/sbin/voyage.update
What would you like to do?
  1 - Create new Voyage Linux disk
  2 - Update existing Voyage configuration
  3 - Exit
      (default=1 [Create new Voyage Linux disk]): 
```

Figure B.3: Installation script window 1

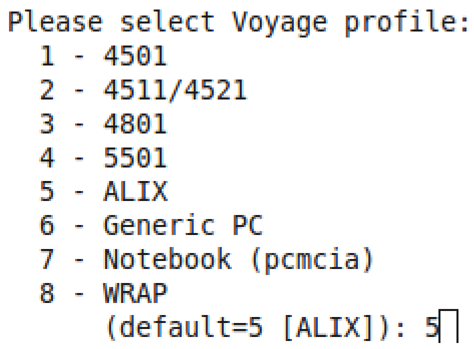
5. The window shown in Figure B.4 will appear. Select option “2” (default option).



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=2 [Select Target Profile]):
```

Figure B.4: Installation script window 2

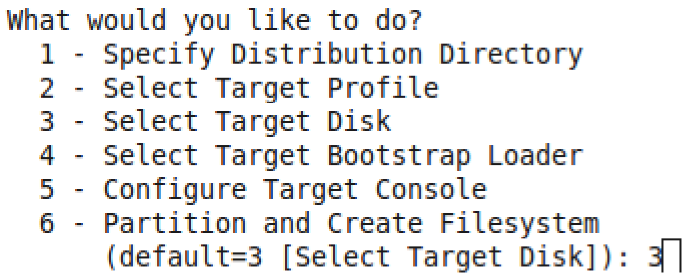
6. The window shown in Figure B.5 will appear. Select option “5” (default option).



```
Please select Voyage profile:
  1 - 4501
  2 - 4511/4521
  3 - 4801
  4 - 5501
  5 - ALIX
  6 - Generic PC
  7 - Notebook (pcmcia)
  8 - WRAP
      (default=5 [ALIX]): 5
```

Figure B.5: Installation script window 3

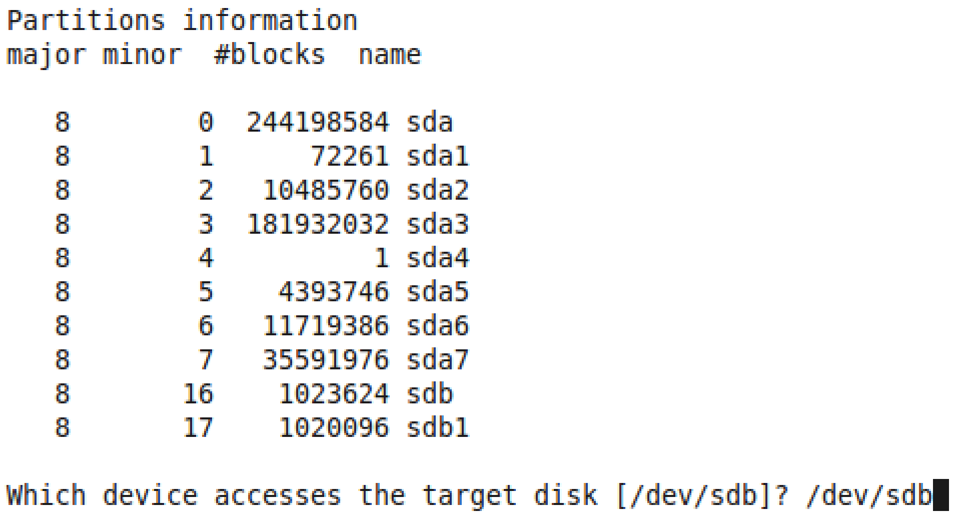
7. The window shown in Figure B.6 will appear. Select option “3” (default option).



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=3 [Select Target Disk]): 3
```

Figure B.6: Installation script window 4

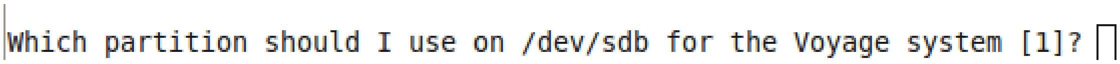
8. The window shown in Figure B.7 will appear. In this step, the mount point of the new CFC needs to be specified. In this case, the mount point is “`/dev/sdb`.”



```
Partitions information
major minor  #blocks  name

   8        0  244198584 sda
   8        1      72261 sda1
   8        2   10485760 sda2
   8        3  181932032 sda3
   8        4          1 sda4
   8        5    4393746 sda5
   8        6   11719386 sda6
   8        7   35591976 sda7
   8       16    1023624 sdb
   8       17    1020096 sdb1

Which device accesses the target disk [/dev/sdb]? /dev/sdb
```

Figure B.7: Installation script window 5

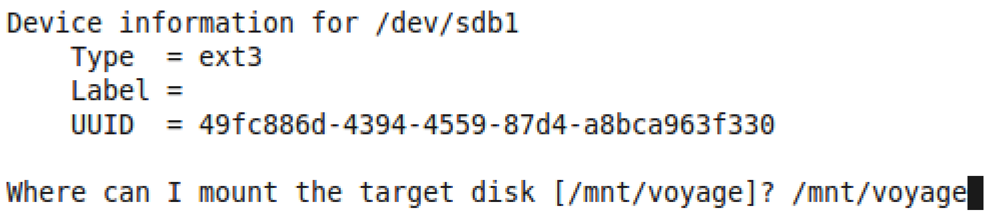
9. The prompt shown in Figure B.8 will appear. In this step, the partition to be used needs to be specified. This is again the default value “1.”



```
Which partition should I use on /dev/sdb for the Voyage system [1]?
```

Figure B.8: Installation script window 6

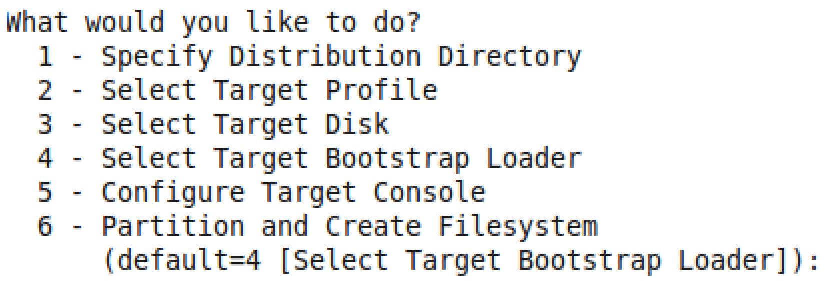
10. The window shown in Figure B.9 will appear. In this step, a mounting place for the CFC needs to be specified while the files are being copied. This is the directory created in step 3 (i.e., "`/mnt/voyage`").



```
Device information for /dev/sdb1
    Type  = ext3
    Label =
    UUID  = 49fc886d-4394-4559-87d4-a8bca963f330

Where can I mount the target disk [/mnt/voyage]? /mnt/voyage
```

Figure B.9: Installation script window 7

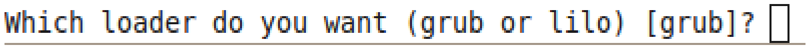
11. The window shown in Figure B.10 will appear. Select option "4" (default value).



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=4 [Select Target Bootstrap Loader]):
```

Figure B.10: Installation script window 8

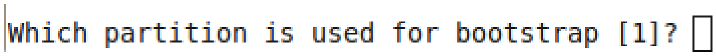
12. The window shown in Figure B.11 will appear. In this step, a default loader for the operating system will be requested. Select the default value (i.e., grub).



```
Which loader do you want (grub or lilo) [grub]?
```

Figure B.11: Installation script window 9

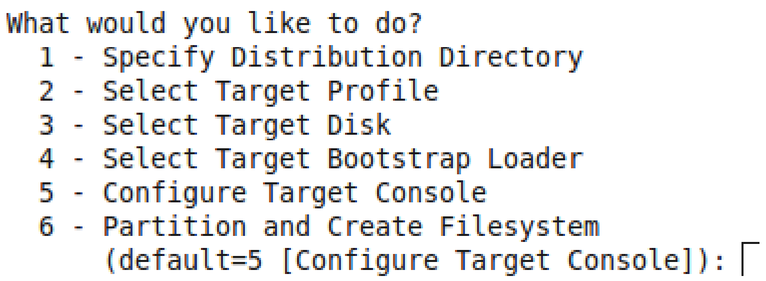
13. The window shown in Figure B.12 will appear. Select the default value (i.e., "1") for the partition to use.



```
Which partition is used for bootstrap [1]?
```

Figure B.12: Installation script window 10

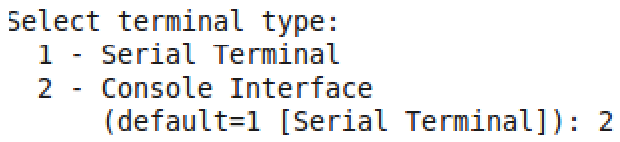
14. The window shown in Figure B.13 will appear. Select the default value (i.e., "5") for the interface.



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=5 [Configure Target Console]):
```

Figure B.13: Installation script window 11

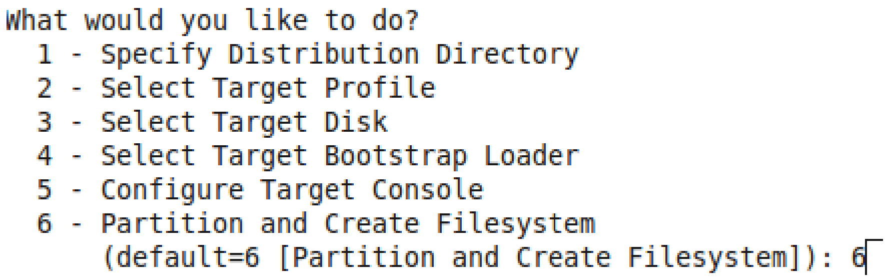
15. The window shown in Figure B.14 will appear. Select "Console Interface" (i.e., option "2") for the terminal type.



```
Select terminal type:
  1 - Serial Terminal
  2 - Console Interface
      (default=1 [Serial Terminal]): 2
```

Figure B.14: Installation script window 12

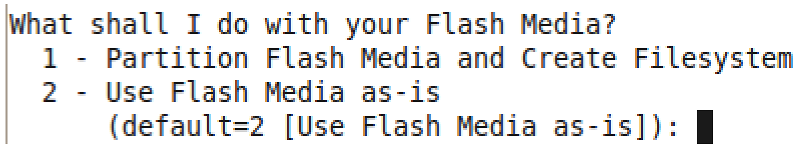
16. The window shown in Figure B.15 will appear. Select the default option (i.e., "6") for the next question.



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=6 [Partition and Create Filesystem]): 6
```

Figure B.15: Installation script window 13

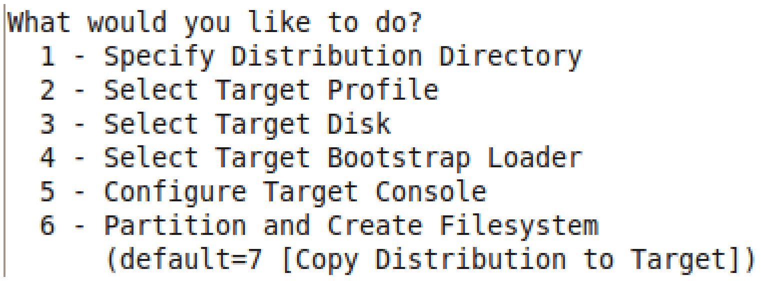
17. The window shown in Figure B.16 will appear. Since the CFC has already been partitioned, there is no need to do this again. Thus, select option "2."



```
What shall I do with your Flash Media?
  1 - Partition Flash Media and Create Filesystem
  2 - Use Flash Media as-is
      (default=2 [Use Flash Media as-is]):
```

Figure B.16: Installation script window 14

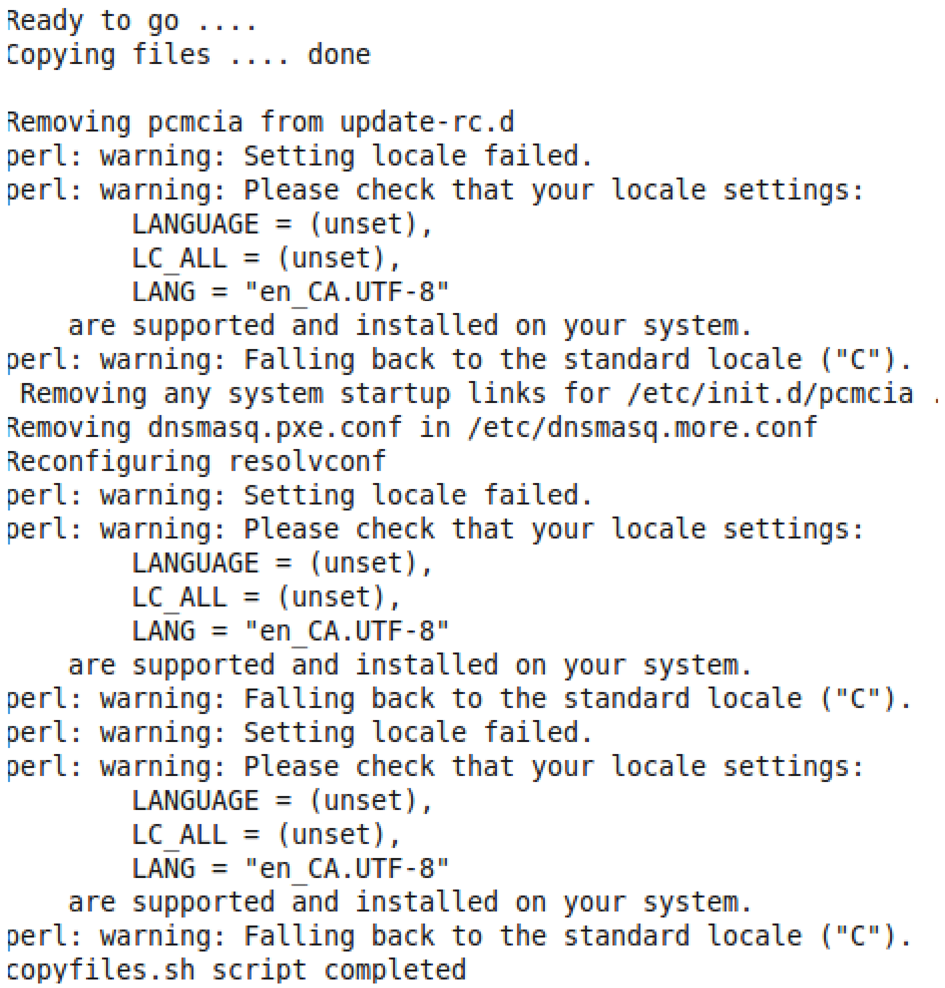
18. The window shown in Figure B.17 will appear. Select the default option (i.e., "7") to copy the files to the CFC.



```
What would you like to do?
  1 - Specify Distribution Directory
  2 - Select Target Profile
  3 - Select Target Disk
  4 - Select Target Bootstrap Loader
  5 - Configure Target Console
  6 - Partition and Create Filesystem
      (default=7 [Copy Distribution to Target])
```

Figure B.17: Installation script window 15

19. A question will come up asking if the process needs to be continued. Type “y” and then press <Enter>. If an error message is generated, you will be returned to the menu in step 4. This error will indicate that the CFC is mounted. Thus, open Gparted and unmount the CFC. Once the CFC has been unmounted, repeat the process starting on step 5 through 19. After the files are copied, the window and messages depicted in Figure B.18 will be displayed.



```
Ready to go ....
Copying files .... done

Removing pcmcia from update-rc.d
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LANG = "en_CA.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
 Removing any system startup links for /etc/init.d/pcmcia .
Removing dnsmasq.pxe.conf in /etc/dnsmasq.more.conf
Reconfiguring resolvconf
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LANG = "en_CA.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LANG = "en_CA.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
copyfiles.sh script completed
```

Figure B.18: Installation script window 16

Verify that the last line “`copyfiles.sh script completed`” is included in the message. The CFC can now be inserted into the ALIX board. However, the operating system still needs to be configured and several packages need to be installed.

Cloning the Software Image to a New Compact Flash Card

This section will explain how to clone the CFC and outline the steps to install the image on a new and blank CFC. It is assumed that the reader possesses a CFC with the operating system, the script running and all the software needed.

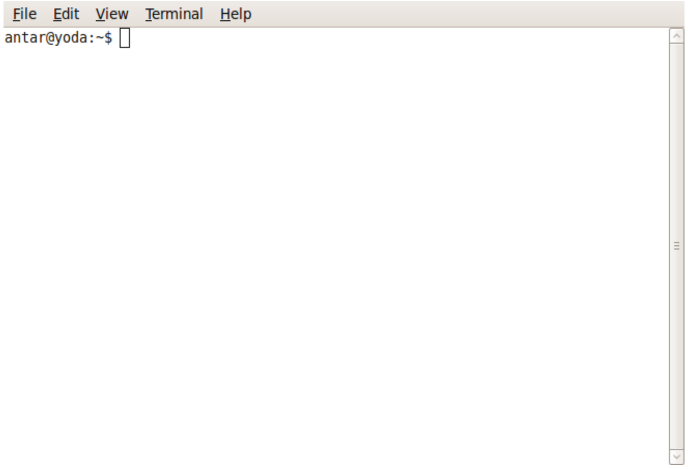
To clone the CFC’s image to a new CFC, the following software and hardware are required:

- A computer running the Ubuntu distribution of Linux (version 9.04 or newer).
- Gparted installed in Ubuntu. Gparted can be installed by running the following command in a command line: “`sudo apt-get install gparted`”.
- A CFC USB reader.

Then, follow these steps:

1. Connect the CFC USB reader to the USB port of the computer running the Ubuntu distribution of Linux.

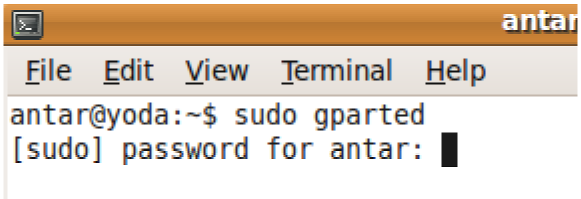
2. Open a Linux command line terminal. In Ubuntu, go to “Applications” ---> “Accessories” ---> “Terminal.” A terminal window similar to that depicted in Figure B.19 should open. In this particular example, the terminal window shows the current user (i.e., antar) and the name of the computer (i.e., yoda).



```
antar@yoda:~$
```

Figure B.19: Linux terminal window

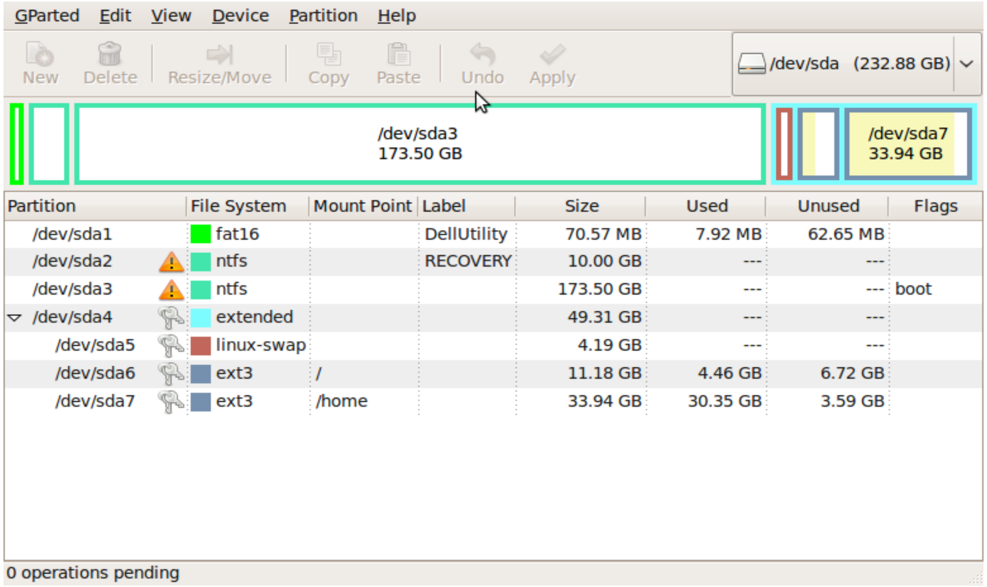
3. In this step, the mount point of the CFC is needed. To get this information, open Gparted in the console login as “`root`” by running the command “`sudo gparted`,” as depicted in Figure B.20.



```
antar@yoda:~$ sudo gparted
[sudo] password for antar: █
```

Figure B.20: Opening Gparted

4. Enter the computer’s user password when prompted and then press <Enter>. The window depicted in Figure B.21 should open.

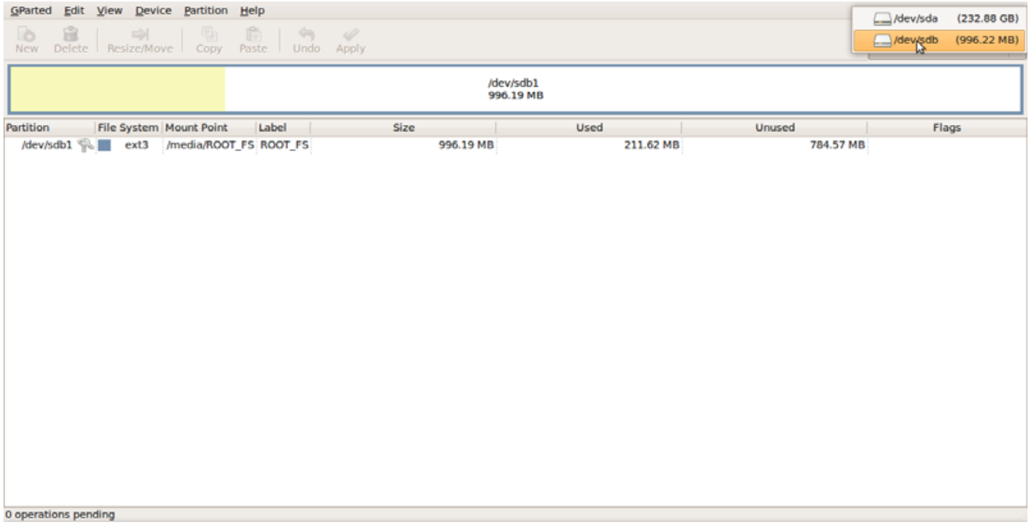

GParted Edit View Device Partition Help
New Delete Resize/Move Copy Paste Undo Apply
/dev/sda (232.88 GB)
/dev/sda3
173.50 GB
/dev/sda7
33.94 GB

Partition		File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sda1		🟩 fat16		DellUtility	70.57 MB	7.92 MB	62.65 MB	
/dev/sda2	⚠	🟩 ntfs		RECOVERY	10.00 GB	---	---	
/dev/sda3	⚠	🟩 ntfs			173.50 GB	---	---	boot
▽ /dev/sda4		🟦 extended			49.31 GB	---	---	
/dev/sda5		🟥 linux-swap			4.19 GB	---	---	
/dev/sda6		🟦 ext3	/		11.18 GB	4.46 GB	6.72 GB	
/dev/sda7		🟦 ext3	/home		33.94 GB	30.35 GB	3.59 GB	

0 operations pending

Figure B.21: Gparted interface

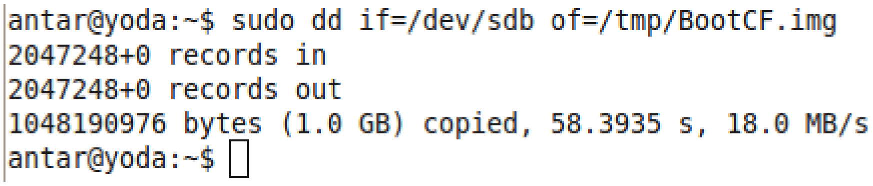
5. On the upper right corner of the Gparted interface, click on the drop down window that displays a hard drive icon (see Figure B.22). The drop down list should indicate where the CFC is mounted. In this case, the CFC is mounted on “`/dev/sdb`.” Note also that the CFC can be easily identified by its storage capacity (i.e., close to 1 GB) which is shown in the middle of the Gparted screen once an entry in the drop down box is selected.



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sdb1	ext3	/media/ROOT_FS	ROOT_FS	996.19 MB	211.62 MB	784.57 MB	

Figure B.22: Selecting the CFC in Gparted

6. Close Gparted and go back to the terminal window. Run the command "`sudo dd if=/dev/sdb of=/tmp/BootCF.img`," as depicted in Figure B.23. The command will run for approximately one (1) minute. This command does not provide any feedback while it is running). If successful, running this command makes an image of the CFC and stores it in "`/tmp/BootCF.img`."

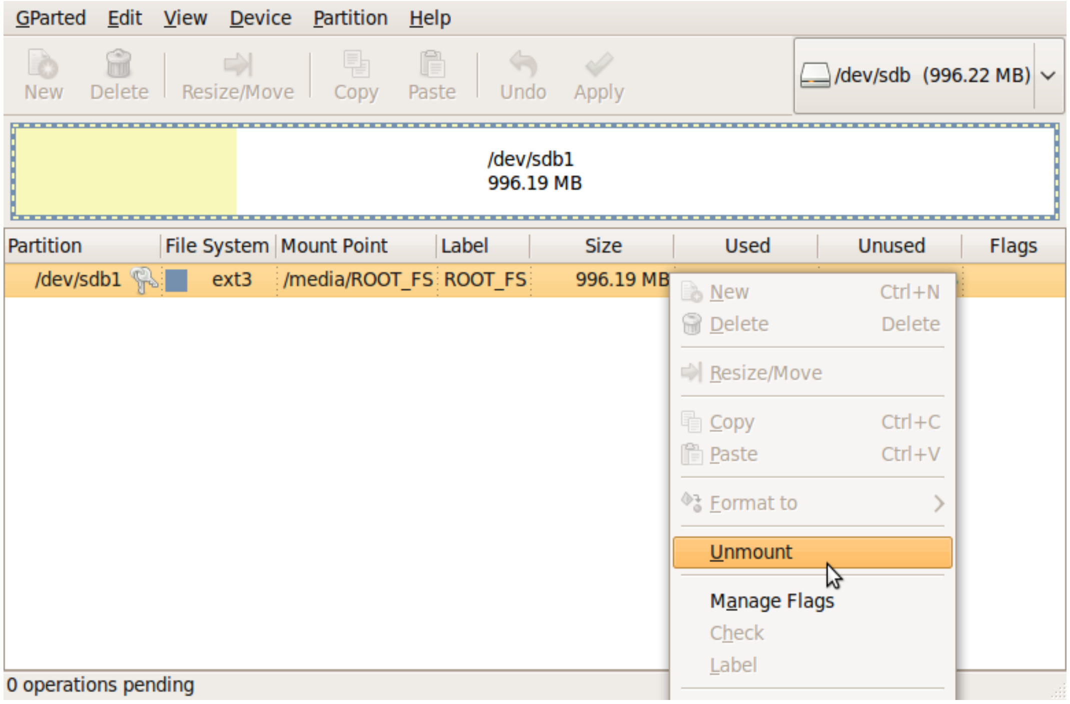


```
antar@yoda:~$ sudo dd if=/dev/sdb of=/tmp/BootCF.img
2047248+0 records in
2047248+0 records out
1048190976 bytes (1.0 GB) copied, 58.3935 s, 18.0 MB/s
antar@yoda:~$ 
```

Figure B.23: Copying the CFC to the hard drive

7. Remove the CFC from the USB card reader (make sure you eject the card reader before unplugging the CFC from the reader). Go to the desktop and right click on the CFC icon and select "Unmount volume." Insert the new CFC and use Gparted to ensure that it is mounted in the same location (i.e., "`/dev/sdb`").

8. Before installing the image in the new and blank CFC, the CFC needs to be formatted. Usign Gparted, select the CFC (i.e., "`/dev/sdb`"). If there is any previous partition in the CFC, it must be erased (see step 9). If there is no pre-existing partition, go to step 10 to create a new partition. Right click on the partition and select "Unmount", as depicted in Figure B.24.

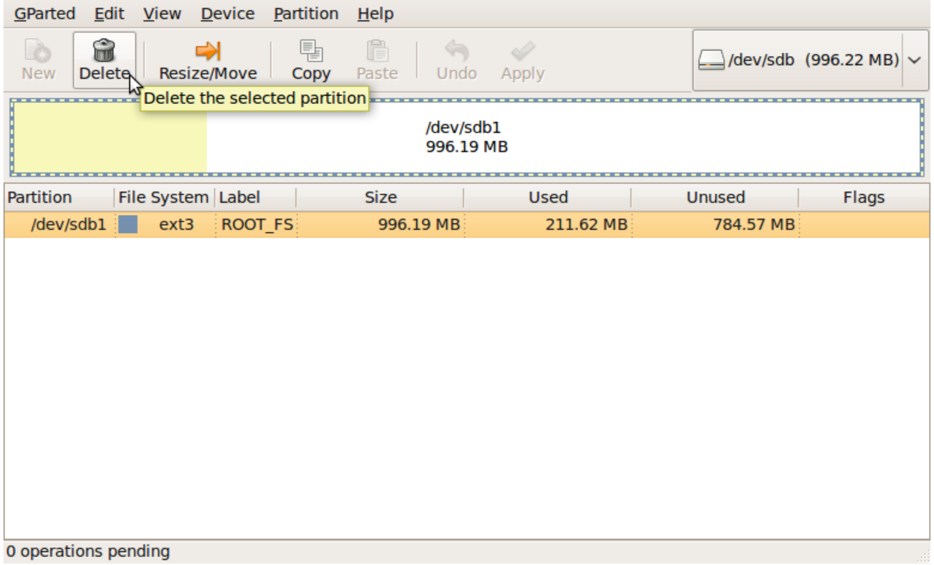


Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sdb1	ext3	/media/ROOT_FS	ROOT_FS	996.19 MB			

0 operations pending

Figure B.24: Unmounting the CFC in Gparted

9. Left mouse click on the partition again to highlight it. The menu icon “Delete” will become active. Press the “Delete” button to delete the partition, as depicted in Figure B.25.

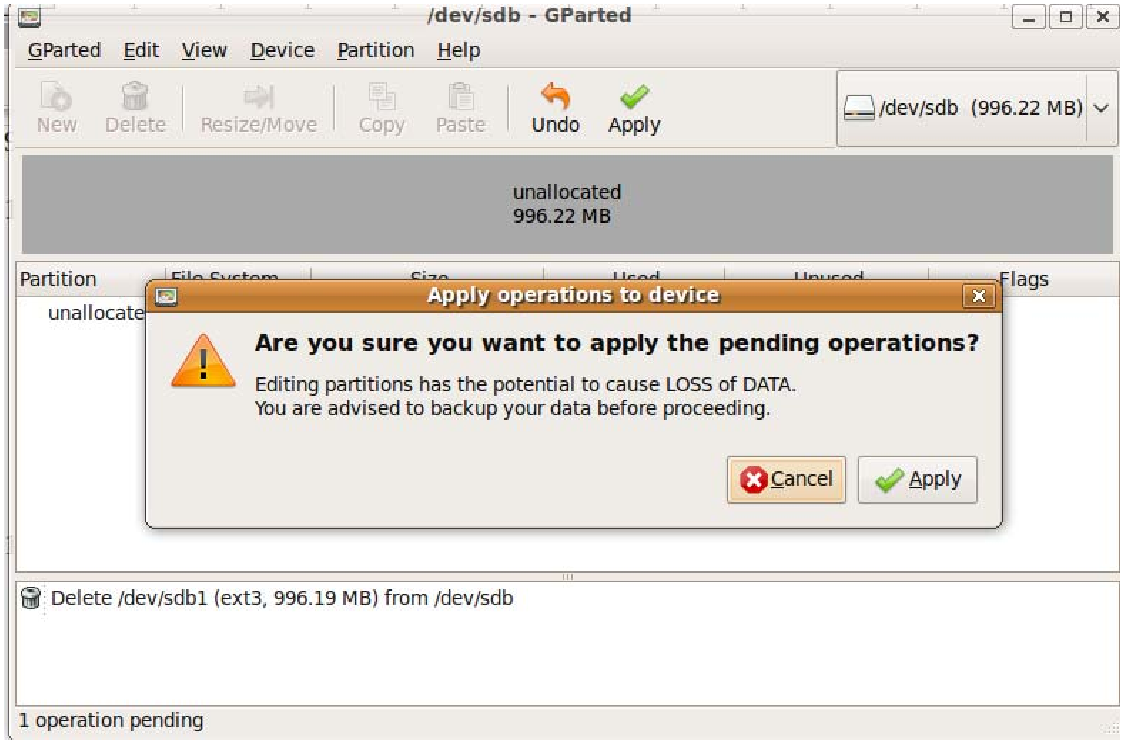


Partition	File System	Label	Size	Used	Unused	Flags
/dev/sdb1	ext3	ROOT_FS	996.19 MB	211.62 MB	784.57 MB	

0 operations pending

Figure B.25: Deleting the CFC’s partition in Gparted

10. After clicking on the “Delete” icon, click on “Apply” to make the changes. Confirm the selection when prompted (see Figure B.26).



Are you sure you want to apply the pending operations?

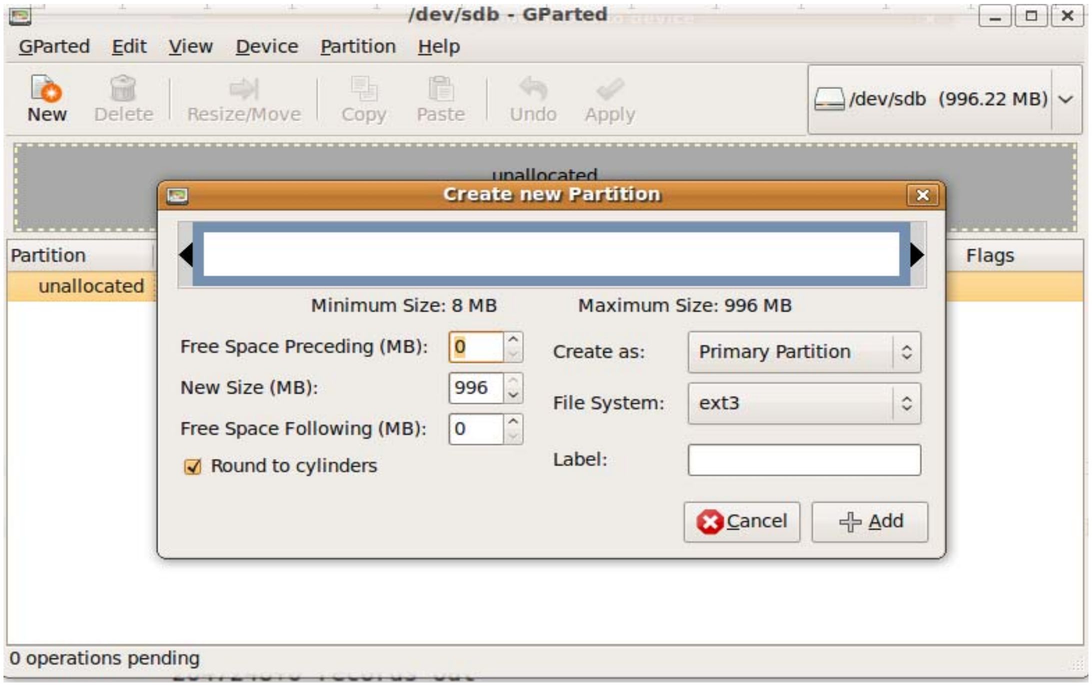
Editing partitions has the potential to cause LOSS of DATA.
You are advised to backup your data before proceeding.

Delete /dev/sdb1 (ext3, 996.19 MB) from /dev/sdb

1 operation pending

Figure B.26: Deleting the CFC’s partition in Gparted (cont.)

11. After the configuration steps are completed in Gparted, a new partition must be created in the CFC. Highlight the partition and press the “New” button. A new window will appear. Select “`ext3`” from the drop down menu labeled “File System.” Press the “Add” button to set the changes (see Figure B.27) and this window will close. Click on “Apply” to create the partition.



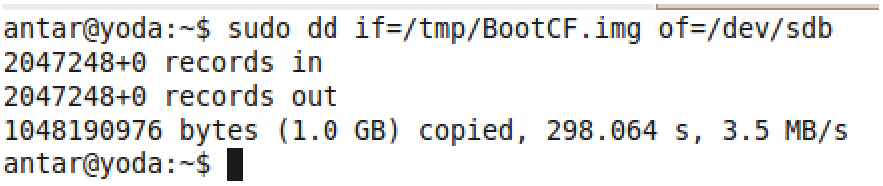
Minimum Size: 8 MB
Maximum Size: 996 MB

Free Space Preceding (MB): 0
New Size (MB): 996
Free Space Following (MB): 0
☑ Round to cylinders

Create as: Primary Partition
File System: ext3
Label:

Figure B.27: Creating a partition in the CFC

12. Close Gparted and go to the Linux terminal window. Run the command shown at the top of Figure B.28. Running this command may take several minutes and no feedback will be provided while running. This command copies the image into the new CFC.



```
antar@yoda:~$ sudo dd if=/tmp/BootCF.img of=/dev/sdb
2047248+0 records in
2047248+0 records out
1048190976 bytes (1.0 GB) copied, 298.064 s, 3.5 MB/s
antar@yoda:~$ █
```

Figure B.28: Copying image to the CFC

13. Once the output shown in Figure B.28 is displayed, the CFC is ready to be inserted into the ALIX board.

CONFIGURING THE ALIX BOARD TO RUN THE BLUETOOTH SCRIPT

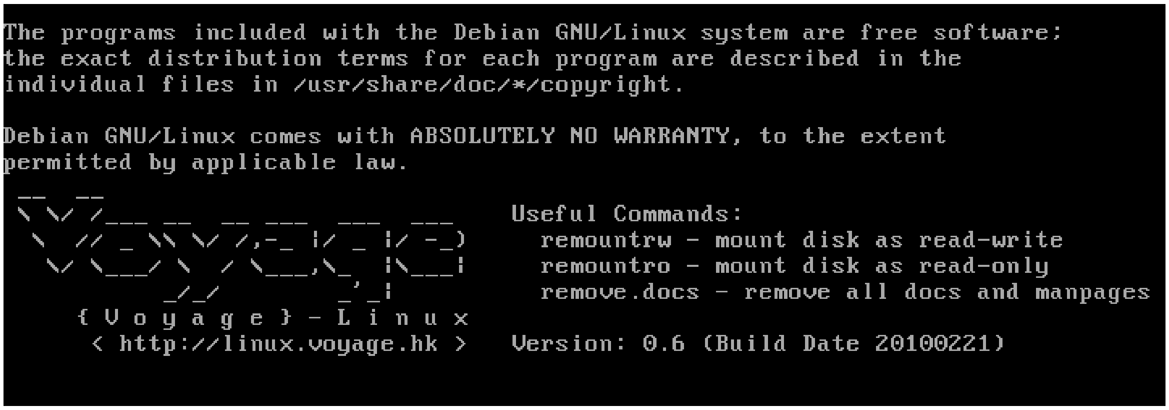
This section details the steps needed to configure the ALIX board so the Bluetooth script can run. It is assumed that the DCU is connected to a LAN and that the following hardware is available:

- ALIX board with newly installed operating system in the CFC.
- A VGA PC monitor.
- A USB keyboard.
- Ethernet cable.

It is important to note that the procedures explained in the sections titled "*Accessing the DCU via a Remote Connection*" and "*Accessing the DCU via a Direct Connection*" can be used to gain access to the DCU instead of the hardware listed above. If this is the case, skip to step 2 in the detailed procedure that follows:

1. Connect the ALIX board to a keyboard and a PC monitor. Do not turn on the power until all hardware components are properly connected. Also, the ALIX board needs to be connected to a power source and needs access to the Internet through the Ethernet port. Once all the connections are completed, turn on the power of the ALIX board.

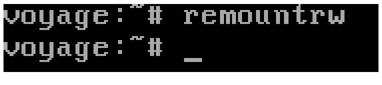
2. As soon as the ALIX board is powered up, the operating system will boot. The booting procedure should be displayed on the PC monitor. After the booting process is complete, the screen displayed in Figure B.29 will appear. Type the login and password (i.e., *root* and *voyage*, respectively).



```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
 __  __
 \ \/ /___ __  __ ___  ___  ___     Useful Commands:
  \  // _ \\ \/ /,-_ |/ _ |/ -_)      remountrw - mount disk as read-write
   \/ \___/ \  / \___,\_  |\___|      remountro - mount disk as read-only
           _/_/        _'_|           remove.docs - remove all docs and manpages
     { V o y a g e } - L i n u x
      < http://linux.voyage.hk >    Version: 0.6 (Build Date 20100221)
```

Figure B.29: Voyage Linux interface

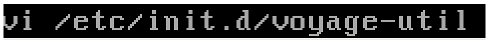
3. After login, the system needs to be "remounted" because it is initially "read only." To remount the file system as "read/write," run the command shown in Figure B.30.



```
voyage:~# remountrw
voyage:~# _
```

Figure B.30: Remount command

4. The system can now be modified and updated. First, the file “`voyage-util`” needs to be modified to make the system writable every time. Type the command shown in Figure B.31 and press <Enter>.



```
vi /etc/init.d/voyage-util
```

Figure B.31: Editing voyage-util file

5. Executing the previous command will open a text file on the screen of the “vi” text editor. A manual on how to use this editor can be found at: http://www.cs.fsu.edu/general/vimanual.html.

6. In this file, the code line “`/usr/local/sbin/remountro`” (see Figure B.32) needs to be modified by adding a “#”.

```
echo -n "Remounting / as read-only ... "
#/bin/mount / -o remount,ro
/usr/local/sbin/remountro
```

Figure B.32: Original voyage-util code lines

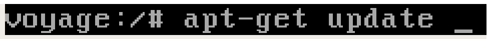
Use the arrow keys to move down to this line and when the cursor gets to the place where the “#” character needs to be added, press the “i” key. The editor will now be in writing mode. Type the “#” character. To save the changes, press the “Esc” key and afterwards the capital “Z” key twice. Verify that the changes look like the code shown in Figure B.33.

```
echo -n "Remounting / as read-only ... "
#/bin/mount / -o remount,ro
#/usr/local/sbin/remountro
```

Figure B.33: Edited voyage-util code lines

7. For the devices already installed on 99 W, the login and password are *root* and *depaT#ts910*, respectively. If you wish to set a different password, type on the command line the command “`passwd`.” You will be asked for the new password for the account “`root`.” When you write the new password the cursor will not move.

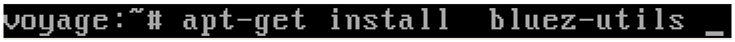
8. Run a command to download updates from the Internet, as shown in Figure B.34. By running the command “`apt-get update`,” the system will grab and install many packages from the Internet.



```
voyage:/# apt-get update _
```

Figure B.34: Update command for ALIX

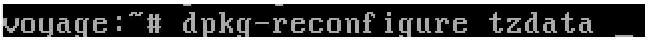
9. Run the command to install the Bluetooth packages, as shown in Figure B.35. This will allow the Bluetooth libraries to run.



```
voyage:~# apt-get install  bluez-utils _
```

Figure B.35: Installing bluez-utils

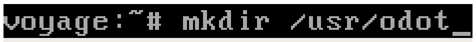
10. Configure the time zone by running the command shown in Figure B.36. A menu with geographical areas will be displayed. Choose option “2” for America. Another section will be displayed with several cities. Press <Enter> three more times to go to the next window with more cities. Once the cursor is next to the words “Time zone,” enter the number “81” and press <Enter>. To verify the time zone has been set correctly, run the command “`date`” in the terminal windows.



```
voyage:~# dpkg-reconfigure tzdata _
```

Figure B.36: Time zone configuration

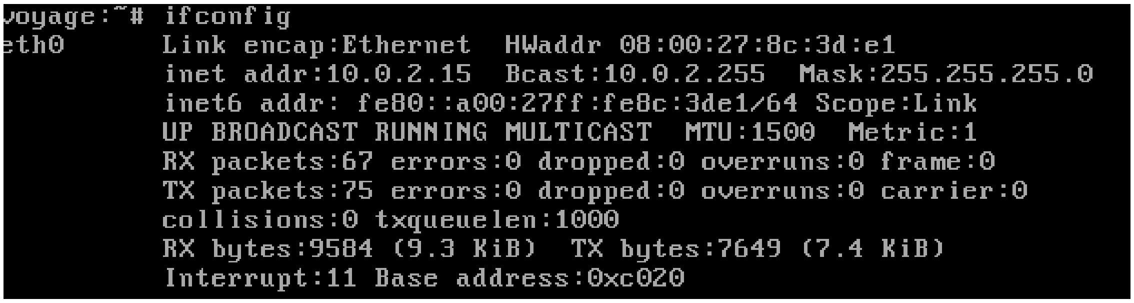
11. Run the command shown in Figure B.37 to create a directory for the Bluetooth script.



```
voyage:~# mkdir /usr/odot_
```

Figure B.37: Creating odot folder in ALIX

12. Run the command “`ifconfig`” in the DCU to obtain its Internet Protocol (IP) address, as shown in Figure B.38. In this example, the IP address is 10.0.2.15.

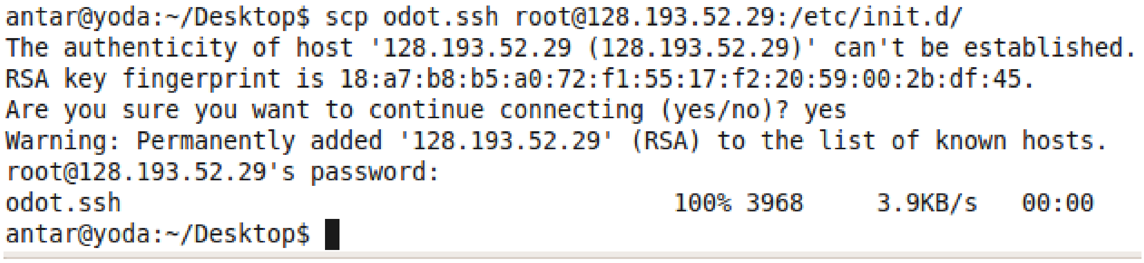


```
voyage:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8c:3d:e1
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8c:3de1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:67 errors:0 dropped:0 overruns:0 frame:0
          TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9584 (9.3 KiB)  TX bytes:7649 (7.4 KiB)
          Interrupt:11 Base address:0xc020
```

Figure B.38: Running the ifconfig command in the ALIX board

13. Now the Bluetooth script must be transferred to the DCU. In this example, the Bluetooth script is located in the Desktop of the computer being used to configure the DCU. Open a terminal window and navigate to the Desktop by typing “`cd ~Desktop`.”

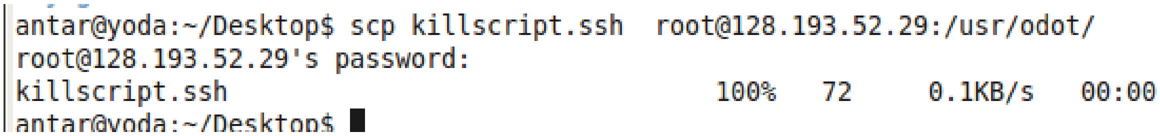
14. Run the command shown at the top of Figure B.39 to transfer the script to the DCU.



```
antar@yoda:~/Desktop$ scp odot.ssh root@128.193.52.29:/etc/init.d/
The authenticity of host '128.193.52.29 (128.193.52.29)' can't be established.
RSA key fingerprint is 18:a7:b8:b5:a0:72:f1:55:17:f2:20:59:00:2b:df:45.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '128.193.52.29' (RSA) to the list of known hosts.
root@128.193.52.29's password:
odot.ssh                                      100% 3968     3.9KB/s   00:00
antar@yoda:~/Desktop$ █
```

Figure B.39: Transferring the script to the DCU

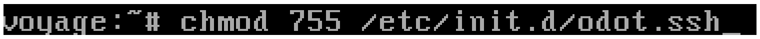
15. An additional file must be transferred to the DCU called "`killscript.ssh`." Run the command shown at the top of Figure B.40 to transfer it.



```
antar@yoda:~/Desktop$ scp killscript.ssh  root@128.193.52.29:/usr/odot/
root@128.193.52.29's password:
killscript.ssh                                100%   72     0.1KB/s   00:00
antar@yoda:~/Desktop$ █
```

Figure B.40: Transferring "killscript.ssh" to the DCU

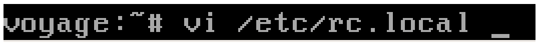
16. Back on the DCU, run the command shown in Figure B.41.



```
voyage:~# chmod 755 /etc/init.d/odot.ssh_
```

Figure B.41: Chmod command

17. The "vi" text editor will be used again in this step. In the DCU, run the command shown in Figure B.42.



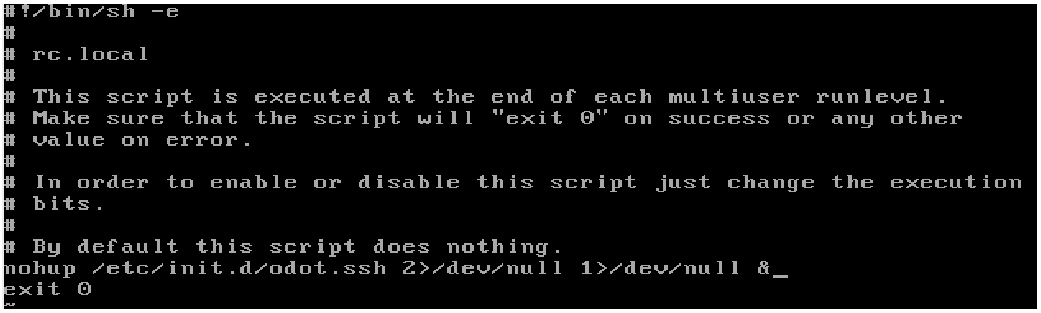
```
voyage:~# vi /etc/rc.local _
```

Figure B.42: Editing rc.local

This action will open a text file. You will notice a line that says "exit 0." Write the following code line just above the "exit 0" line.

```
nohup /etc/init.d/odot.ssh 2>/dev/null 1>/dev/null &
```

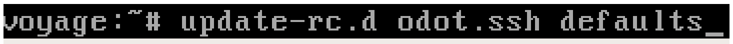
To do this, move the cursor with the arrow keys to above "exit 0." Then press the "i" key and then press <Enter> to enter the writing mode. Write the instruction line shown above. If an error is made while typing, press the <Esc> key, move the cursor to the part to be deleted and press the "x" key. To enter writing mode again, press "i." Figure B.43 shows how the file should look like at the end. Press the "Esc" key and then press "Z" two times when finished with the "vi" editor.



```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
nohup /etc/init.d/odot.ssh 2>/dev/null 1>/dev/null &_
exit 0
~
```

Figure B.43: Rc.local file

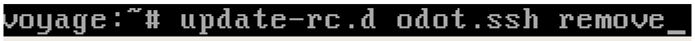
18. Run the command shown in Figure B.44 and press <Enter>.



```
voyage:~# update-rc.d odot.ssh defaults_
```

Figure B.44: Update RC defaults

19. Run the command shown in Figure B.45 and press <Enter>.



```
voyage:~# update-rc.d odot.ssh remove_
```

Figure B.45: Update remove

The DCU is now ready to begin capturing time-stamped MAC addresses from Bluetooth-enabled devices. Reboot the system to test that it works properly.

ASSEMBLING THE DCU

Table B.1 shows the components required to assemble a DCU. Parts names, quantities needed of each part, and URLs to vendors' web sites are included. The last column on Figure B.46 indicates the figure within this section that depicts the corresponding part.

Table B.1: Components needed to assemble DCUs

PART NAME	QTY	WHERE TO BUY IT	FIGURE #
ICF 4000 Industrial Compact Disk 1GByte	1	http://www.mini-box.com/1GB-Inndustrial-Compact-Disk-4000	Figure B.46
ALIX3D3 board	1	http://www.mini-box.com/Alix-3D-Board-3-LAN-1-MINI-PCI-1_3?sc=8&category=19	Figure B.47
Indoor enclosure for ALIX.3 Boards	1	http://www.mini-box.com/ALIX-3-Enclosure?sc=8&category=19	Figure B.48
I/O Bracket for Alix.3D3 (includes 8 screws)	2	http://www.mini-box.com/Bracket-for-Alix-3c2_2	Figure B.49
18w (15v/1.2A) AC-DC Power Adapter	1	http://www.mini-box.com/60w-12v-5A-AC-DC-Power-Adapter_3?sc=8&category=19	Figure B.50a
POE adapter	1	http://www.mini-box.com/s.nl/it.A/id.309/.f	Figure B.50b
SENA UD100 Bluetooth dongle	1	http://www.sena.com/where_to_buy/channel_partners/	Figure B.50c

Figure B.46: ICF 4000 industrial compact disk 1GB

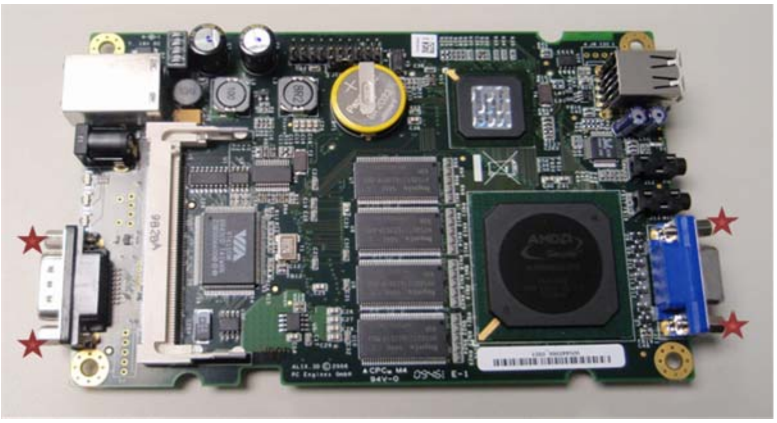


Figure B.47: ALIX3D3 board

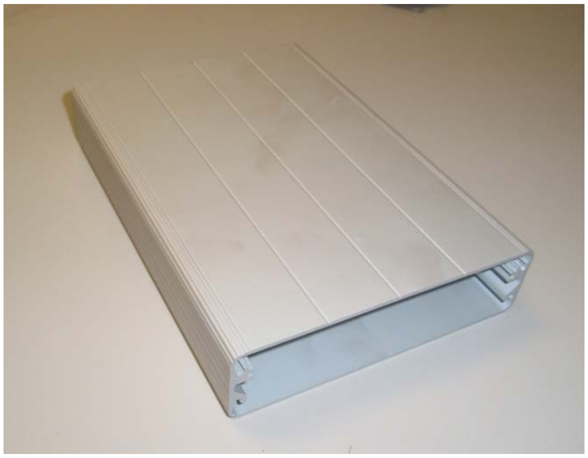


Figure B.48: Indoor enclosure for ALIX.3 Boards

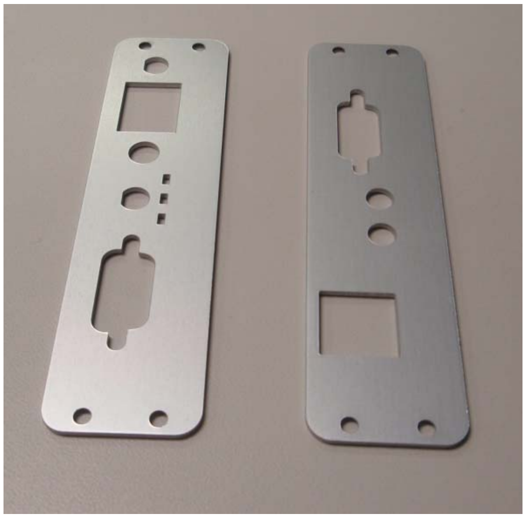


Figure B.49: I/O bracket for ALIX.3D3

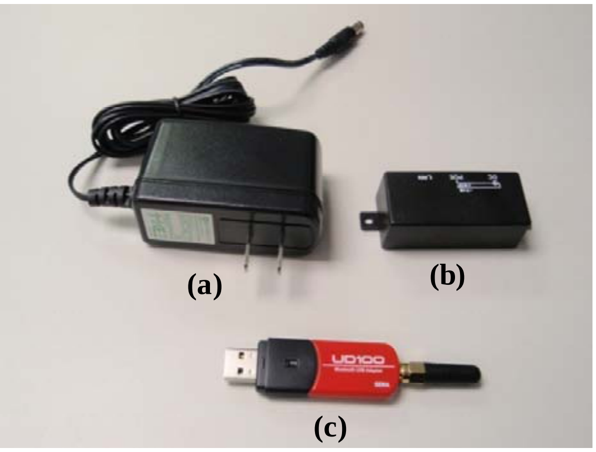


Figure B.50: DCU components: (a) power supply, (b) POE adapter, and (c) Bluetooth USB adapter

The following steps detailed the instructions to assemble a DCU from the parts listed on Table B.1.

1. Install the CFC on the ALIX3D3 board.
2. Figure B.51 shows the top side of the board. The slot where the Compact Flash Card needs to be installed is on the bottom side. Slide the card in the direction of the red arrow, as depicted in Figure B.51.

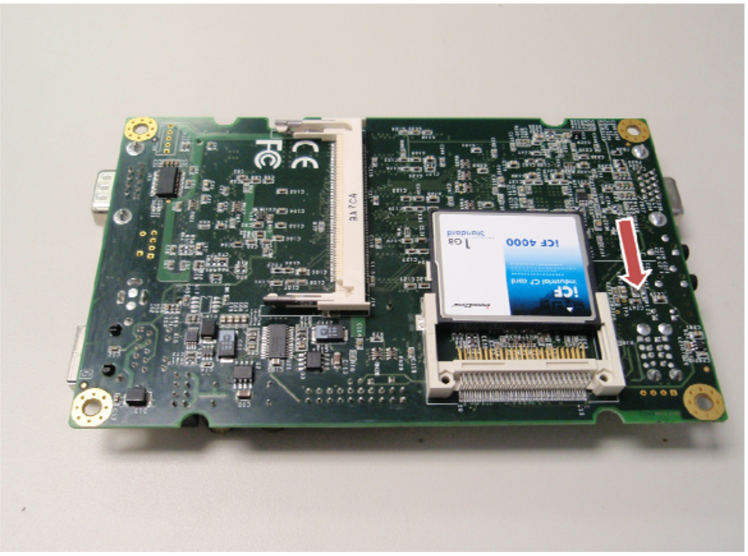


Figure B.51: Insert compact flash card into ALIX board

3. The ALIX board needs to be inserted into the indoor enclosure, as depicted in Figure B.52. The screws marked with red stars in Figure B.47 will need to be removed to install the I/O brackets.

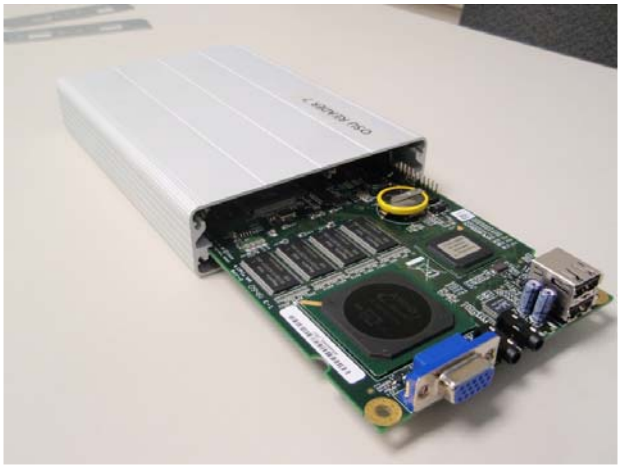


Figure B.52: Slide ALIX board into enclosure

4. The I/O brackets can now be installed on both ends of the DCU's enclosure, as shown in Figure B.53. Note that the screws marked with red stars in Figure B.47 are now back in place.

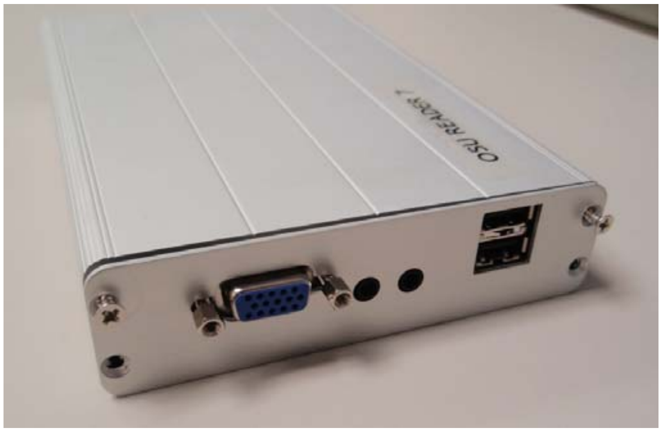


Figure B.53: Installation of I/O brackets

5. The last step in completing the assembly of the DCU is to connect the Bluetooth USB adapter, the power supply, and the Ethernet cables. Two Ethernet cables are needed. Figure B.54 shows how the final DCU assembly looks like. The yellow Ethernet cable goes from the POE to a device that provides network connectivity, such as a cellular router or a LAN router. The grey Ethernet cable provides the DCU with power and network communications. Finally, the power supply needs to be connected to a power source.

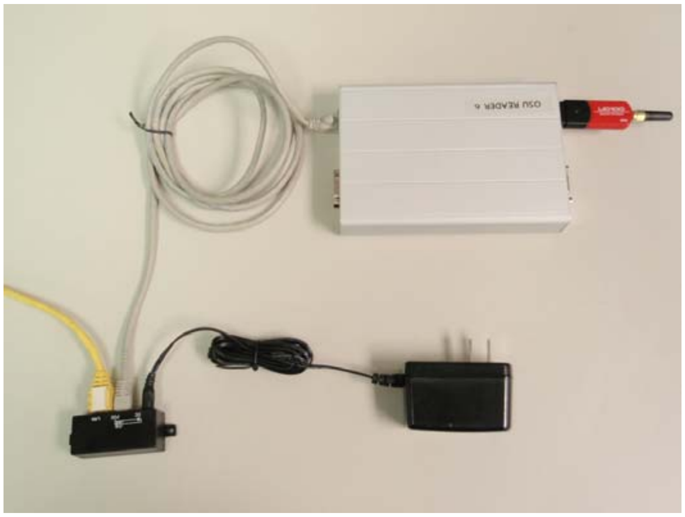


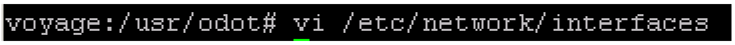
Figure B.54: DCU with POE and Bluetooth USB adapter

Depending on the final installation in the field, this configuration needs to be established in order for the device to work properly.

CONFIGURING A CELLULAR ROUTER TO ACCESS THE DCU REMOTELY

The travel time data collection system installed in Salem, OR, utilized cellular routers to enable remote connectivity to the DCUs for downloading data files and for running diagnostics. This section describes the steps needed to configure a cellular router to enable communications between a DCU and a remote computer.

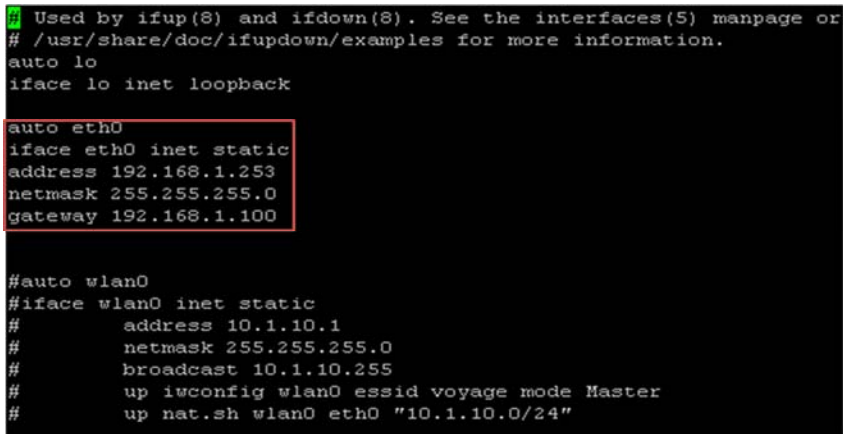
The DCU needs to be configured first. This requires the user to log in into the DCU directly, as explained in the section titled "*Accessing the DCU via a Direct Connection*". Once logged in, the user needs to modify the configuration file "`interfaces`," which is typically located in "`/etc/network`." This can be accomplished by running the command shown in Figure B.55 directly on the DCU interface. Executing this command will open the "vi" editor.



```
voyage:/usr/odot# vi /etc/network/interfaces
```

Figure B.55: Opening the configuration file "interfaces"

The information depicted in Figure B.56 will be displayed. The text in Figure B.56 shown inside the red rectangle is the new information that needs to be added. The gateway and subnet mask are the internal IP addresses from the cellular router.

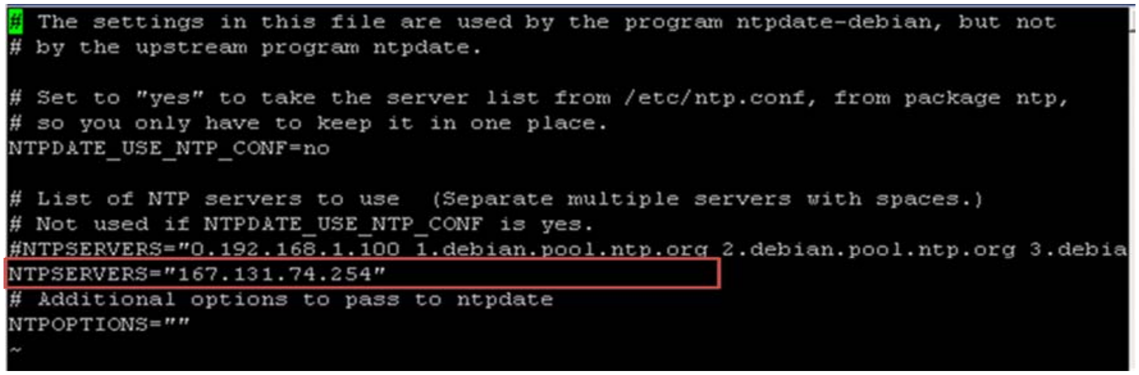


```
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.253
netmask 255.255.255.0
gateway 192.168.1.100


#auto wlan0
#iface wlan0 inet static
#        address 10.1.10.1
#        netmask 255.255.255.0
#        broadcast 10.1.10.255
#        up iwconfig wlan0 essid voyage mode Master
#        up nat.sh wlan0 eth0 "10.1.10.0/24"
```

Figure B.56: Contents of the configuration file "interfaces"

After the configuration file “`interfaces`” has been edited and closed, the command “`/etc/init.d/networking restart`” needs to be executed to restart the network.

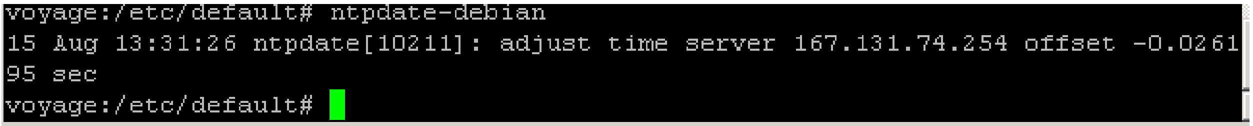
To configure the time synchronization of the DCU, the file “`ntpdate`” needs to be modified. This file is typically located in “`/etc/default`.” Execute the command “`vi /etc/default/ntpdate`” to open it. The contents of file are depicted in Figure B.57. Once again, the text enclosed by the red rectangle needs to be added to the file. In this case, the IP address of the NTP SERVER that provides the cellular router with the correct time is 167.131.74.254.



```
# The settings in this file are used by the program ntpdate-debian, but not
# by the upstream program ntpdate.

# Set to "yes" to take the server list from /etc/ntp.conf, from package ntp,
# so you only have to keep it in one place.
NTPDATE_USE_NTP_CONF=no

# List of NTP servers to use  (Separate multiple servers with spaces.)
# Not used if NTPDATE_USE_NTP_CONF is yes.
#NTPSERVERS="0.192.168.1.100 1.debian.pool.ntp.org 2.debian.pool.ntp.org 3.debia
NTPSERVERS="167.131.74.254"
# Additional options to pass to ntpdate
NTPOPTIONS=""
~
```

Figure B.57: Contents of the file “ntpdate”

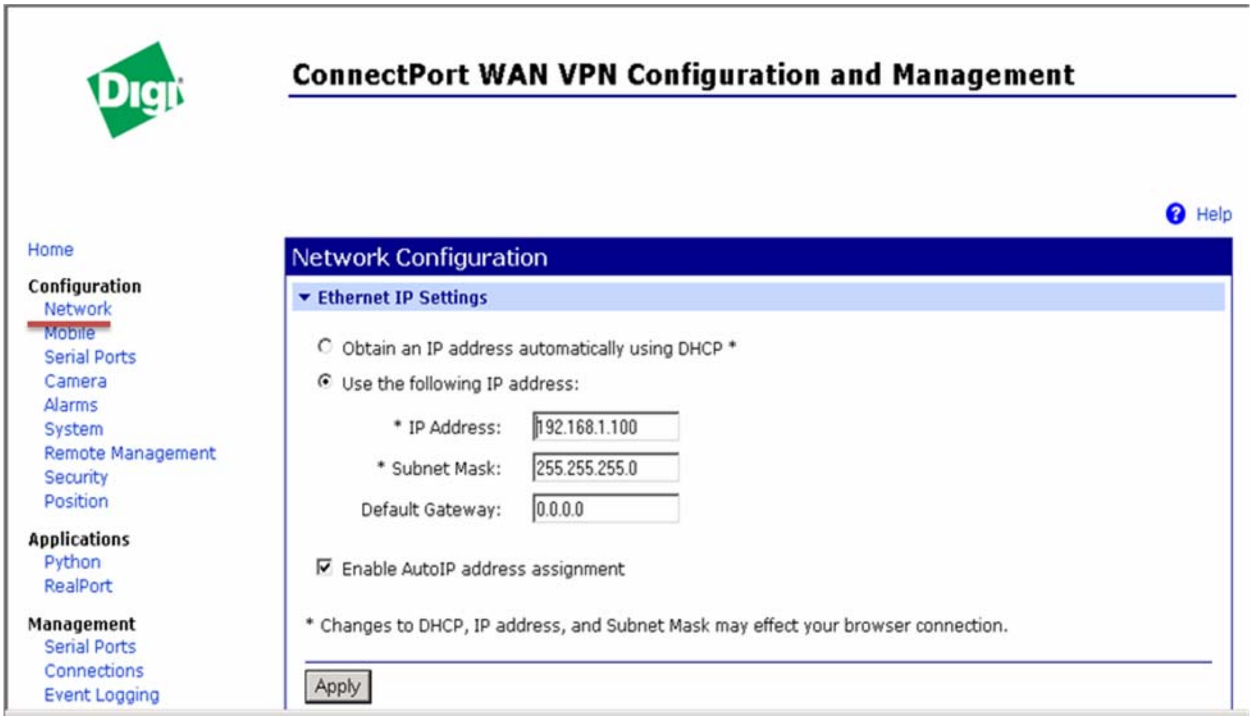
Once the changes to the file “`ntpdate`” are saved, it can be verified that the DCU is time synchronized by running the command “`ntpdate-debian`,” as depicted in Figure B.58.



```
voyage:/etc/default# ntpdate-debian
15 Aug 13:31:26 ntpdate[10211]: adjust time server 167.131.74.254 offset -0.0261
95 sec
voyage:/etc/default#
```

Figure B.58: Executing the command ntpdate-debian

The cellular router should have the configuration depicted in Figure B.59. In the cellular router’s web interface, select the “Network” option from the configuration menu (left side of the screen).



Network Configuration

▼ Ethernet IP Settings

○ Obtain an IP address automatically using DHCP *

● Use the following IP address:

* IP Address: 192.168.1.100

* Subnet Mask: 255.255.255.0

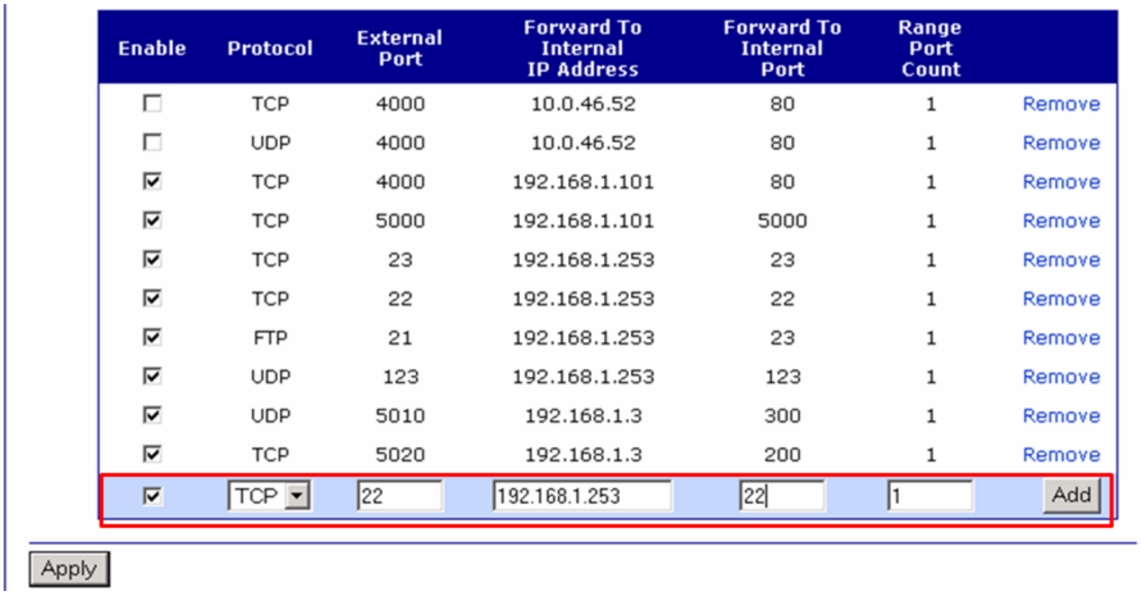
Default Gateway: 0.0.0.0

☑ Enable AutoIP address assignment

* Changes to DHCP, IP address, and Subnet Mask may effect your browser connection.

Figure B.59: Cellular router’s web interface configuration menu

Since data communication between the DCU and the remote computer will take place via port 22, we need to configure the cellular router to forward all communications through this port. Scroll down on this window until the option “IP forwarding settings” is reached. Select this option and scroll down to find the table depicted in Figure B.60. An entry will be added to this table to forward a port to the DCU.



Enable	Protocol	External Port	Forward To Internal IP Address	Forward To Internal Port	Range Port Count	
□	TCP	4000	10.0.46.52	80	1	Remove
□	UDP	4000	10.0.46.52	80	1	Remove
☑	TCP	4000	192.168.1.101	80	1	Remove
☑	TCP	5000	192.168.1.101	5000	1	Remove
☑	TCP	23	192.168.1.253	23	1	Remove
☑	TCP	22	192.168.1.253	22	1	Remove
☑	FTP	21	192.168.1.253	23	1	Remove
☑	UDP	123	192.168.1.253	123	1	Remove
☑	UDP	5010	192.168.1.3	300	1	Remove
☑	TCP	5020	192.168.1.3	200	1	Remove
☑	TCP	22	192.168.1.253	22	1	Add

Figure B.60: IP forwarding table

In this example, the IP address of the DCU is 192.168.1.253, it uses port 22, and the transport protocol is TCP. To add the new entry, enter the following information:

1. Protocol: TCP
2. External port: 22
3. Forward to internal IP address: 192.168.1.253
4. Forward to internal port: 22
5. Press “Add”
6. Press “Apply”

The cellular router is now configured and should allow a remote computer to access the DCU for maintenance and troubleshooting operations.

ACCESSING THE DCU VIA A REMOTE CONNECTION

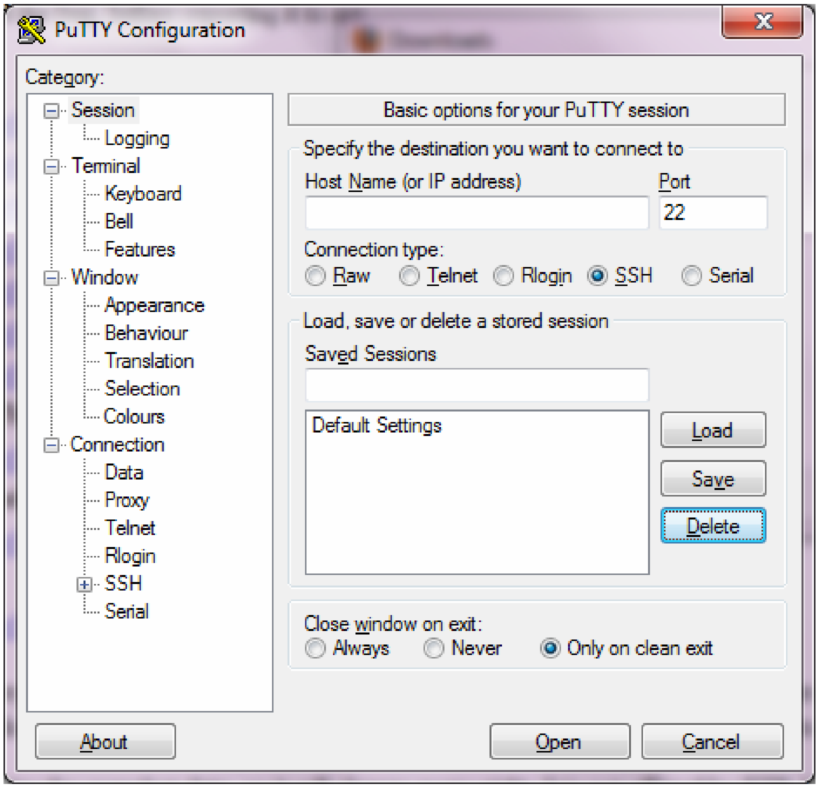
This section describes the process to remotely connect to the DCU. The connection process detailed here may be helpful if it is necessary to access the DCU to verify that the Bluetooth USB adapter is working properly or whether or not the data collection script is running. It is important to note that DCU and the computer (i.e., laptop or desktop) from which the remote communication is being established need to be in the same local area network (LAN).

Connecting Remotely via PuTTY

The DCU can be accessed remotely by using the program PuTTY. PuTTY is a Secure Shell (SSH) and teletype network (Telnet) client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software and is available for download at www.putty.org/.

Once the file PuTTY.exe has been downloaded and saved to the Desktop of the computer, follow these steps to remotely connect to the DCU:

1. Double click on the file PuTTY.exe to open it. The interface depicted in Figure B.61 will appear.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

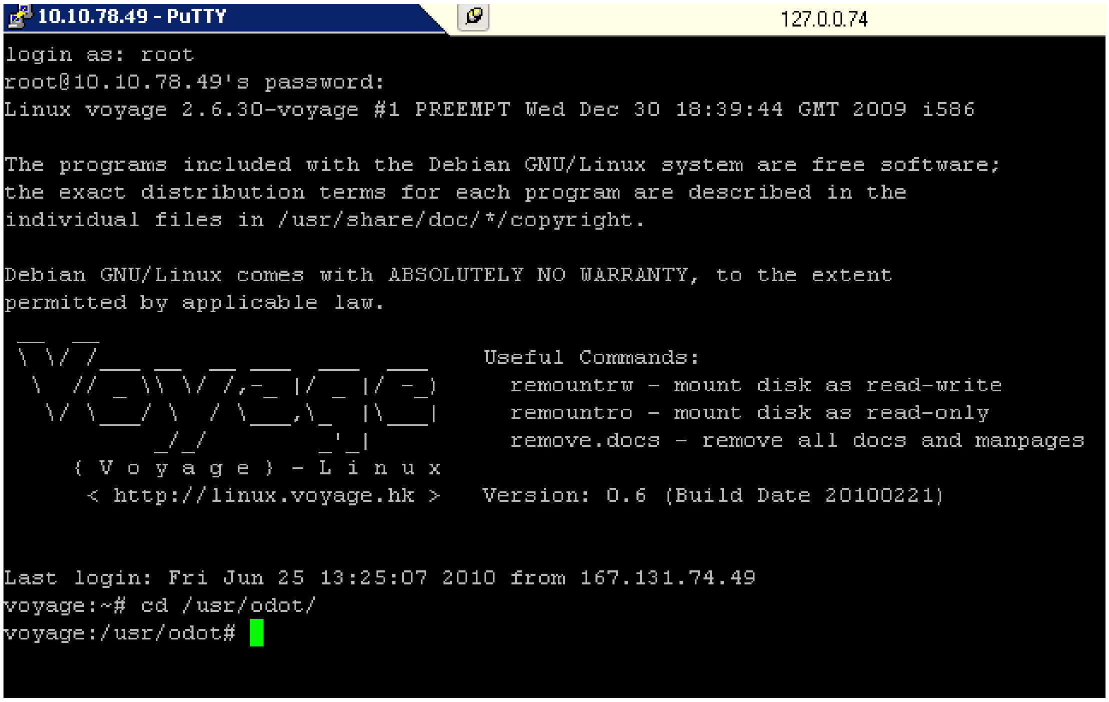
Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure B.61: PuTTY program interface

2. Enter the IP address of the DCU (e.g., 10.10.78.49) in the field “Host Name.” If a cellular router is being used, then the IP address of this device needs to be entered. Verify that the connection type selected is SSH and the press the “Open” button.
3. Another window will open. When prompted, enter “`root`” as the login and press <Enter>. Type “`depaT#ts910`” as the password and press <Enter>. (Note: the cursor will not move while the password is input)
4. The DCU interface will open (see Figure B.62). Now, the processes described in the section titled “*Troubleshooting the DCU*” can be executed.



```
login as: root
root@10.10.78.49's password:
Linux voyage 2.6.30-voyage #1 PREEMPT Wed Dec 30 18:39:44 GMT 2009 i586

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
 __  __
 \ \/ /___ __  __ ___  ___  ___        Useful Commands:
  \  // _ \\ \/ /,-_ |/ _ |/ -_)         remountrw - mount disk as read-write
   \/ \___/ \  / \___,\_  |\___|         remountro - mount disk as read-only
          _/_/        _'_|               remove.docs - remove all docs and manpages
     { V o y a g e } - L i n u x
      < http://linux.voyage.hk >   Version: 0.6 (Build Date 20100221)


Last login: Fri Jun 25 13:25:07 2010 from 167.131.74.49
voyage:~# cd /usr/odot/
voyage:/usr/odot#
```

Figure B.62: DCU interface window

Connecting Remotely via a Linux Terminal

A remote connection to the DCU can be established using a computer running the Ubuntu distribution of Linux.

1. Open a terminal window on the computer running Ubuntu.
2. Type the command “`ssh root@[IP Address]`.” The `[IP address]` entered should be either the DCU’s IP address or the cellular router’s IP address.
3. The DCU interface will open (see Figure B.62). Now, the processes described in the section titled “*Troubleshooting the DCU*” can be executed.

ACCESSING THE DCU VIA A DIRECT CONNECTION

This section describes the process to connect directly to the DCU in case a connection via a cellular router or a network is not available. The connection process outlined here may also be helpful if it is necessary to check the status of the DCUs after completing the software installation process described in the section titled "*Configuring the ALIX Board to Run the Bluetooth Script*." The hardware and software required includes:

- Laptop computer running either Windows or the Ubuntu distribution of Linux (version 9.04 or higher).
- Router. In this example, a DLINK WBR-2310 wireless router (http://www.dlink.com/products/?pid=470) was used.
- Ethernet cable.
- PuTTY program (www.putty.org/)

The steps to directly connect to the DCU are as follows:

1. Identify the slot labeled "LAN" on the power over Ethernet (POE) adapter depicted in Figure B.63. In a typical installation, the LAN slot is used to insert an Ethernet cable whose other end is connected to a cellular router or network router. If this is the case, unplug the end of the Ethernet cable connected to the cellular router or network router and connect it to the router being used to connect directly to the DCU.

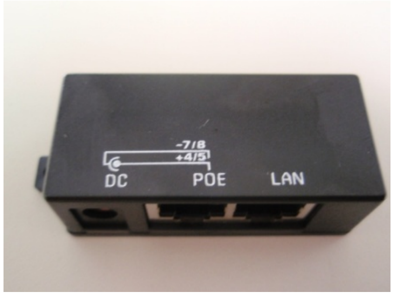


Figure B.63: Power over Ethernet (POE) adapter

2. Connect the laptop computer to the router using an Ethernet cable.
3. Turn on the router.
4. Power cycle the DCU (i.e., unplug it from power and plug it back to power).
5. Open a web browser on the laptop computer and open the router's configuration page. In some laptop computers you will need to turn off the wireless network (if one is enabled). In this example, a D-Link wireless router is used and its configuration page is depicted in Figure B.64. The user name for this router is *admin* and there is no password.

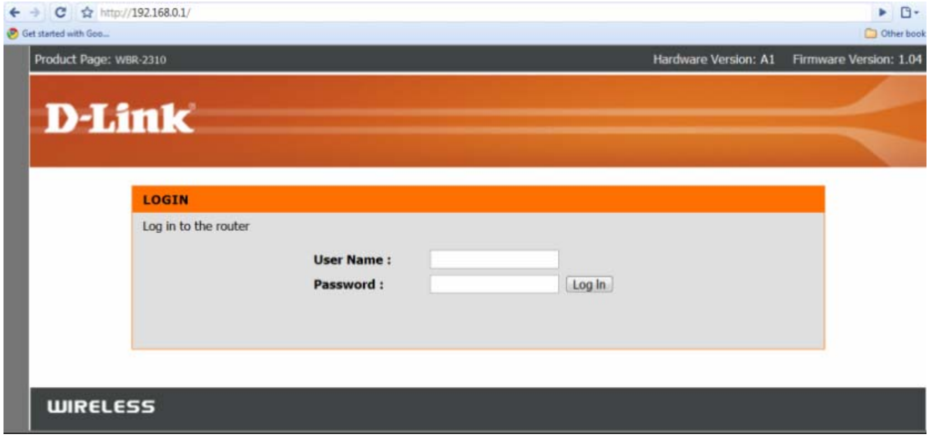
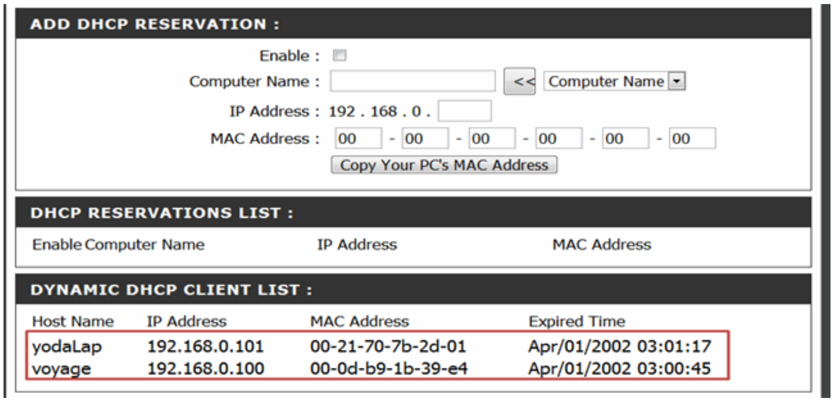


Figure B.64: D-Link configuration interface log in screen

6. Once inside the configuration menu of the D-Link wireless router, access the network settings page to see which devices are listed as being connected to the D-Link wireless router. An entry identifying the DCU should be listed as “voyage” (see Figure B.65). Copy the IP address shown for the DCU (in this example, it is 192.168.0.100).


ADD DHCP RESERVATION :

Enable : □

Computer Name : [ ] << Computer Name ▾

IP Address : 192 . 168 . 0 . [ ]

MAC Address : 00 - 00 - 00 - 00 - 00 - 00

Copy Your PC's MAC Address

DHCP RESERVATIONS LIST :

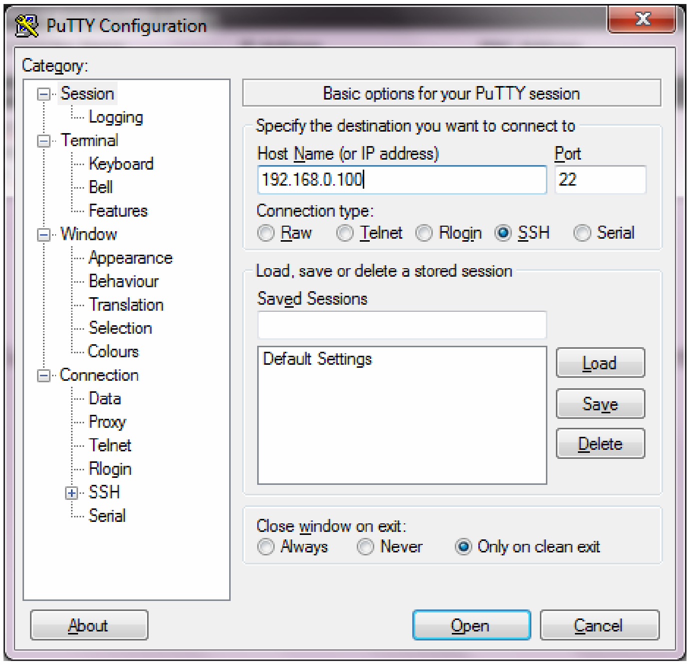
Enable	Computer Name	IP Address	MAC Address

DYNAMIC DHCP CLIENT LIST :

Host Name	IP Address	MAC Address	Expired Time
yodaLap	192.168.0.101	00-21-70-7b-2d-01	Apr/01/2002 03:01:17
voyage	192.168.0.100	00-0d-b9-1b-39-e4	Apr/01/2002 03:00:45

Figure B.65: Network settings page of the D-Link router

7. Open PuTTY (or if in Ubuntu open a console window) to connect to the DCU using the IP address copied from the D-Link wireless router (see Figure B.66). Once connected to the DCU, any of the troubleshooting tests can be performed on the unit.



Host Name (or IP address): 192.168.0.100

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure B.66: PuTTY screen to connect to the router

TROUBLESHOOTING THE DCU

This section describes the steps needed to verify that:

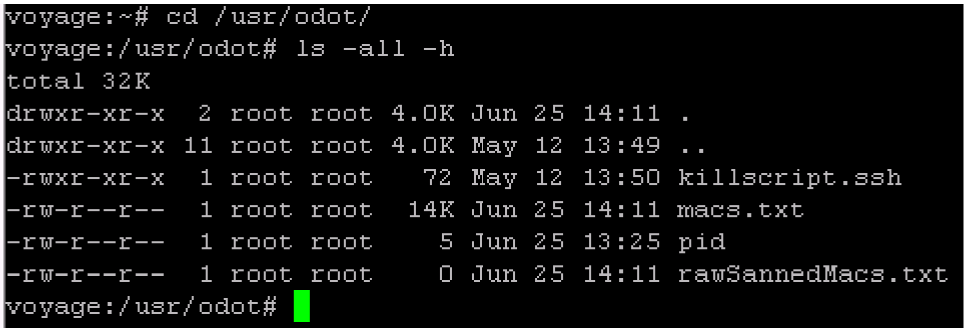
- The Bluetooth adapter attached to the DCU is functioning properly, and
- The script in the DCU that collects MAC addresses from Bluetooth-enabled devices is running.

In order to perform these diagnostics, the user must be connected directly to the DCU or through a remote network connection. Please refer to sections titled “*Accessing the DCU via a Remote Connection*” and “*Accessing the DCU via a Direct Connection*” in this appendix to review how to achieve connectivity to the DCU with either method.

Running Diagnostics on the Bluetooth USB Adapter

The Bluetooth USB adapter connected to the DCU permits the collection of MAC addresses from Bluetooth-enabled devices. If there are reasons to suspect that the adapter is not working properly, execute the following steps:

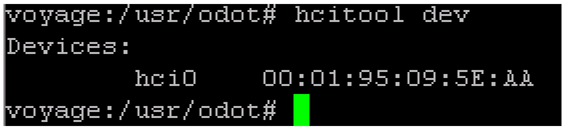
1. Access the DCU via the PuTTY program, as described in the section titled “*Accessing the DCU via a Remote Connection.*”
2. Navigate to the folder where the file that stores the collected MAC addresses is located by typing the command “`cd /usr/odot`” followed by the <Enter> key.
3. List all the files contained in the folder “`/usr/odot`” by typing the command “`ls –all –h`” followed by the <Enter> key (see Figure B.67).



```
voyage:~# cd /usr/odot/
voyage:/usr/odot# ls -all -h
total 32K
drwxr-xr-x  2 root root 4.0K Jun 25 14:11 .
drwxr-xr-x 11 root root 4.0K May 12 13:49 ..
-rwxr-xr-x  1 root root   72 May 12 13:50 killscript.ssh
-rw-r--r--  1 root root  14K Jun 25 14:11 macs.txt
-rw-r--r--  1 root root    5 Jun 25 13:25 pid
-rw-r--r--  1 root root    0 Jun 25 14:11 rawSannedMacs.txt
voyage:/usr/odot# 
```

Figure B.67: MAC address folder

4. Verify that the file ***macs.txt*** file resides in the folder “`/usr/odot`.” If it is not listed, then the Bluetooth USB adapter could be damaged.
5. First, verify that the Bluetooth USB adapter is working by typing the command “`hcitool dev`” followed by the <Enter> key. If the Bluetooth USB adapter is working properly, then its MAC address should be listed as part of the output of the “`hcitool dev`” command, as shown in Figure B.68.



```
voyage:/usr/odot# hcitool dev
Devices:
        hci0    00:01:95:09:5E:AA
voyage:/usr/odot#
```

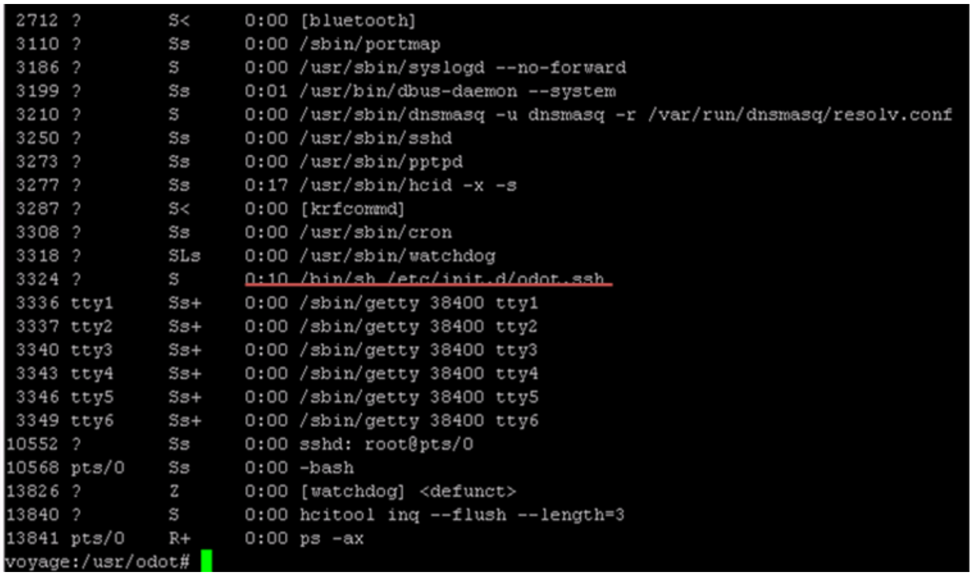
Figure B.68: Output of the `hcitool dev` command

6. If the MAC address of the Bluetooth USB adapter is not shown in the output of the "`hcitool dev`" command, then the adapter is either off or not working properly. To verify that the adapter is powered, issue a reboot on the DCU by typing the command "`reboot -p`" followed by the <Enter> key.
7. Repeat steps 1 through 6 to verify that the Bluetooth USB adapter is working properly. If the problems persist, replace the Bluetooth USB adapter.

Verifying that the Linux Script is Running

A Linux script file running in the DCU enables the collection of MAC addresses from Bluetooth-enabled devices. To verify that the script is running, follow these steps:

1. Access the DCU via the PuTTY, as described in the section titled "*Accessing the DCU via a Remote Connection.*"
2. Type the command "`ps -ax`" followed by the <Enter> key. This command lists all the processes that are running in the DCU. The output depicted in Figure B.69 should be displayed on the DCU interface.



```
 2712 ?        S<     0:00 [bluetooth]
 3110 ?        Ss     0:00 /sbin/portmap
 3186 ?        S      0:00 /usr/sbin/syslogd --no-forward
 3199 ?        Ss     0:01 /usr/bin/dbus-daemon --system
 3210 ?        S      0:00 /usr/sbin/dnsmasq -u dnsmasq -r /var/run/dnsmasq/resolv.conf
 3250 ?        Ss     0:00 /usr/sbin/sshd
 3273 ?        Ss     0:00 /usr/sbin/pptpd
 3277 ?        Ss     0:17 /usr/sbin/hcid -x -s
 3287 ?        S<     0:00 [krfcommd]
 3308 ?        Ss     0:00 /usr/sbin/cron
 3318 ?        SLs    0:00 /usr/sbin/watchdog
 3324 ?        S      0:10 /bin/sh /etc/init.d/odot.ssh
 3336 tty1     Ss+    0:00 /sbin/getty 38400 tty1
 3337 tty2     Ss+    0:00 /sbin/getty 38400 tty2
 3340 tty3     Ss+    0:00 /sbin/getty 38400 tty3
 3343 tty4     Ss+    0:00 /sbin/getty 38400 tty4
 3346 tty5     Ss+    0:00 /sbin/getty 38400 tty5
 3349 tty6     Ss+    0:00 /sbin/getty 38400 tty6
10552 ?        Ss     0:00 sshd: root@pts/0
10568 pts/0    Ss     0:00 -bash
13826 ?        Z      0:00 [watchdog] <defunct>
13840 ?        S      0:00 hcitool inq --flush --length=3
13841 pts/0    R+     0:00 ps -ax
voyage:/usr/odot#
```

Figure B.69: Output of the command `ps –ax`

3. Verify that the process "`/bin/sh /etc/init.d/odot.ssh`" is listed in the output. If is not listed, issue a reboot on the DCU by typing the command "`reboot -p`" followed by the <Enter> key.

4. Repeat steps 1 and 2 and verify that the process "`/bin/sh /etc/init.d/odot.ssh`" is listed. If it is still not listed, then refer to the procedure explained in the section titled "*Configuring the ALIX Board to Run the Bluetooth Script*."

APPENDIX C:
LINUX SHELL SCRIPT CODE

```
sleep 40
get_mac ()
{
echo $$ > /usr/odot/pid
cd /usr/odot

     START_SCRIPT=$(date +%s) #Start of the script
     START=$START_SCRIPT     #Start of the Loop
     TIME_RUN=$1              #Time of running

        currDate=`date +"%b-%d-%y"`
#       hcitool dev > bluetoothDeviceINFO1.txt
#       tail -1 bluetoothDeviceINFO1.txt> bluetoothDeviceJustMAC1.txt
#       eval `awk '{print "export ownMAC1="$2}' bluetoothDeviceJustMAC1.txt`

#       LOGFILE="Start-$currDate.txt" #   -$ownMAC1.txt"
LOGFILE="macs.txt"
        #touch $LOGFILE
rm bluetoothDeviceINFO1.txt bluetoothDeviceJustMAC1.txt

if [ -f macs.txt ]
then
cat macs.txt >> backup.txt
rm macs.txt
fi

while :
     do


trap "{ /usr/odot/killscript.ssh ; exit 255; }" EXIT TERM
        datevariable=`date '+%m/%d/%y,%H:%M:%S'`
        date2=`date '+%m/%d/%y'`
        #echo $date2
        #muffin=cats.txt
        time=`date '+%S.%N'`
        time2=`date '+%S.%N'`


 hcitool inq  --flush --length=3 > rawScannedMacs.txt

num=`wc -l < rawScannedMacs.txt`
 #count number of lines in this file echo $num usually the first line says
"inquiring" and then the macs


if [ $num -ge 2 ] # if any device is detected ge=greater than or equal
     then
        hcitool dev > bluetoothDeviceINFO.txt
        tail -1 bluetoothDeviceINFO.txt> bluetoothDeviceJustMAC.txt
        eval `awk '{print "export ownMAC="$2}' bluetoothDeviceJustMAC.txt`
  #gets the MAC address to the variable ownMAC,
#the $2 is the second column of the bluetoothDeviceJustMAC.txt file
```

```
        num2=`expr $num - 1`

        tail -$num2  rawScannedMacs.txt > JustScannedMacs.txt #cuts the first
line, the one says "inquiry"
        awk '{print substr($1,8,9)}' JustScannedMacs.txt > EditedMacs.txt
 #1st argument= the first text file, 2nd = cuts the first 8 characters.
#now we use the data in EditedMacs.txt to compare that data with the one we
#have in FinalData.txt


touch $LOGFILE  #creates final file with mac addresses.text file
awk '{print substr($1,0,10)}' $LOGFILE > FinalDataToCompare.txt
#check if the FinalData file has something, if it does then we do the
comparisons
isEmpty=`wc -l < $LOGFILE`
 #to check if the mac address is found in the next loop
        if [ $isEmpty -eq 0 ]; then
        writeFinalData #if the file is empty we write the MACs found
        else
        #compare each line in FinalDataToCompare to EditedMacs
        for line1 in $(cat EditedMacs.txt)
                do
                macFound="notFound"
                for line2 in $(cat FinalDataToCompare.txt)
                        do
                                if [[ $line1 = $line2 ]];
                                 then
                                        macFound="notFound"
                                        break

                                fi
                done
        #here we write the mac if it was not found
        if [[ $macFound = "notFound" ]];
        then
         echo "$line1",$datevariable,$ownMAC >> $LOGFILE

        echo writing $line1

        else

        echo not writing $line1

        fi

        done

        fi
        rm bluetoothDeviceJustMAC.txt EditedMacs.txt FinalDataToCompare.txt
JustScannedMacs.txt
        rm rawScannedMacs.txt bluetoothDeviceINFO.txt
fi

   # END=$(date +%s) # end time
    # DIFF=$(( $END - $START )) #time of running
     #DIFF_SCRIPT=$(( $END - $START_SCRIPT )) # time of running
```

```
       # if [ $DIFF_SCRIPT -gt $TIME_RUN ] # gt = greater than
     #then
      #   echo DONE SCANNING


       #  return 0  #when we return 0 this is the equivalent of a break
     #fi
     done # end of while loop

}

writeFinalData()
{

 for line in $(cat EditedMacs.txt)
         do
          echo "$line",$datevariable,$ownMAC >> $LOGFILE
 #Fields: First is the scanned MAC, second is the date, third is the time,
fourth is the MAC # of the reader, all fields delimited by comas

         done
}


if [ $# -eq 0  ]  # $# means total number of arguments supplied, -eq means
equal

then
       echo "Running for 1 second";
       get_mac  #& #pipeline

else

        case "$1" in
            start)
              nohup /etc/init.d/odot.ssh &
               exit 0;;

        *) echo "odot.ssh start";;

        esac

fi

exit 0
```

APPENDIX D:
PSEUDO-CODE FOR TRAVEL TIME SAMPLES THROUGH MAC ADDRESS MATCHING

```
NumberofDays = to be input based on data file (1-5).

MaxTime = 0.1677;  /* units = hours */
MaxReadTimeBetween = 20;

/* Create arrays */

Array1[5000,7];
Array2[5000,7];
Array3[50000,3];
Array4[50000,3];
Groups1[50000,5];
Groups2[50000,5];
TravelTimes[10000,8];

/* Other variables */

Integer TotalAddresses1, TotalAddresses2; TotalRecords1; TotalRecords2; TotalGroupss1; TotalGroups2; TotalEstimates;

/* Files below are created from Excel pivot tables. Each record is for a unique address read by the reader, and the number of times it was read for each day. Sorted by MACaddress. */

File1 = macaddresspivot1.txt;    /* Reader 1 */
File2 = macaddresspivot2.txt;    /* Reader 2 */


/* Read data into arrays */
RecordNumber  = 1;
while (File1(RecordNumber, 1) not end of file) do
{
        for i = 1 to NumberofDays+1 do
        {
                Array1[RecordNumber, i] =  File1(RecordNumber, i);
        }
        Array1[RecordNumber, 7] = 0;
        RecordNumber  = RecordNumber +1;
        TotalAddresses1 = RecordNumber;
}

RecordNumber  = 1;
while (File2(RecordNumber, 1) not end of file) do
{
        for i = 1 to NumberofDays+1 do
        {
                Array2[RecordNumber, i] =  File2(RecordNumber, i);
        }
        Array2[RecordNumber, 7] = 0;
        RecordNumber  = RecordNumber +1;
        TotalAddresses2 = RecordNumber;

}


/* First identify matches – record in 7th column of Array1 and Array2 */
```

```
for i = 1 to TotalAddresses1 do
{
        match = 0;
        j = 1;
        while ( j <= TotalAddresses2 and match = 0) do
        {
                if (Array1[i,1] = Array2[j,1]) then
                {
                        match = 1;
                        Array1[i,7] = 1;
                        Array2[i,7] = 1;
                }
                Else
                {
                        j = j+1;
                }
        }
}

/* Next, extract only those addresses that have matches from the data files */

/* Assume files have been modified so that time has been converted to hours with 12:00AM on the starting date = 0
hours. For example if start date is 7/23/10 then 1PM (13:00:00) on 7/23/10  = 13.0 hrs. 1:15AM (01:15:00) on
7/24/10 = 25.25 hrs. These files are sorted by Address, Date, Hours. Very easy to do in Excel. */

File1 = macaddress1.txt;    /* Reader 1 */
File2 = macaddress2.txt;    /* Reader 2 */

j = 1;
RecordNumber =1;
for i = 1 to TotalAddresses1 do
{
        if (Array1[i,7] = 1) then
        {
                found = 0;
                while (found = 0) do
                {
                        if (Array1[i,1] = File1(RecordNumber, 1)) then
                        {
                                while(Array1[i,1] = File1(RecordNumber, 1)) do
                                {
                                        Array3[j,1] = File1(RecordNumber, 1);
                                        Array3[j,2] = File1(RecordNumber, 2);
                                        Array3[j,3] = File1(RecordNumber, 3);
                                        RecordNumber = RecordNumber +1;
                                        j = j+1;
                                }
                                found = 1;
                        }
                        Else
                        {
                                RecordNumber = RecordNumber +1;
                        }
                }
        }
```

```
}
TotalRecords1 = RecordNumber -1;

/* Repeat for second file */

j = 1;
RecordNumber =1;
for i = 1 to TotalAddresses2 do
{
    if (Array2[i,7] = 1) then
    {
        found = 0;
        while (found = 0) do
        {
            if (Array2[i,1] = File2(RecordNumber, 1)) then
            {
                while(Array2[i,1] = File2(RecordNumber, 1)) do
                {
                    Array4[j,1] = File2(RecordNumber, 1);
                    Array4[j,2] = File2(RecordNumber, 2);
                    Array4[j,3] = File2(RecordNumber, 3);
                    RecordNumber = RecordNumber +1;
                    j = j+1;
                }
                found = 1;
            }
            Else
            {
                RecordNumber = RecordNumber +1;
            }
        }
    }
}
TotalRecords2 = RecordNumber -1;

/* Create "groups" for MAC address with matches */

/* Group1[i,1] = MACaddress, Group1[i,2] = Date, Group1[i,3] = first time, Group1[i,4] = last time, Group1[i,5] =
count */

j = 1;    /*Group number */

Group1[j,1] = Array3[1,1];
Group1[j,2] = Array3[1,2];
Group1[j,3] = Array3[1,3];

groupcount = 1;
prioraddress = Array3[1,1];
priortime  = Array3[1,3];

/* Need to add code to write the Group arrays to files */

for i = 2 to TotalRecords1 do
{
    if (Array3[i,1] = prioraddress  and (Array3[i,3] -  priortime)  < MaxReadTimeBetween ) then
    {
        priortime  = Array3[i,3];
        groupcount = groupcount +1;
        if (i = TotalRecords1)
        {
            Group1[j,4] = priortime;
            Group1[j,5] = groupcount;
        }
    }
    Else
    {
        Group1[j,4] = priortime;
        Group1[j,5] = groupcount;
        j =j+1;
        Group1[j,1] = Array3[i,1];
        Group1[j,2] = Array3[i,2];
        Group1[j,3] = Array3[i,3];
        priortime  = Array3[i,3];
        groupcount = 1;
        if (i = TotalRecords1)
        {
            Group1[j,4] = priortime;
            Group1[j,5] = groupcount;
        }

    }
}
TotalGroups1 = j;

/* Repeat for second reader */

j = 1;    /*Group number */

Group2[j,1] = Array4[1,1];
Group2[j,2] = Array4[1,2];
Group2[j,3] = Array4[1,3];

groupcount = 1;
prioraddress = Array4[1,1];
priortime  = Array4[1,3];

for i = 2 to TotalRecords2 do
{
    if (Array4[i,1] = prioraddress  and (Array4[i,3] -  priortime)  < MaxReadTimeBetween ) then
    {
        priortime  = Array4[i,3];
        groupcount = groupcount +1;
        if (i = TotalRecords2)
        {
            Group2[j,4] = priortime;
            Group2[j,5] = groupcount;
        }
    }
    Else
    {
        Group2[j,4] = priortime;
        Group2[j,5] = groupcount;
```

```
                j =j+1;
                Group2[j,1] = Array4[i,1];
                Group2[j,2] = Array4[i,2];
                Group2[j,3] = Array4[i,3];
                priortime  = Array4[i,3];
                groupcount = 1;
                if (i = TotalRecords2)
                {
                    Group2[j,4] = priortime;
                    Group2[j,5] = groupcount;
                }

        }
}
TotalGroups2 = j;

/* Generate  travel time samples */
/* Group1[i,1] = MACaddress, Group1[i,2] = Date, Group1[i,3] = first time, Group1[i,4] = last time, Group1[i,5] = count */
k  = 1;
for i = 1 to TotalGroups1 do
{
        match = 0;
        j = 1;
        while ( j <= TotalGroups2 and match = 0) do
        {
                if (Group1[i,1] = Group2[j,1]) then
                {
                        if(abs[(Group1[i,3] + Group1[i,4])/2 – (Group2[j,3] + Group2[j,4])/2] < MaxTime)
                        {
/* Store address, date, time, and five travel time samples: avg- avg, first-first, first- last, last- first, last – last */
                                match = 1;
                                TravelTimes[k,1] = Group1[i,1];
                                TravelTimes[k,2] = Group1[i,2];
                                TravelTimes[k,3] = Max(Group1[i,4], Group2[j,4]);


                                TravelTimes[k,4] = (Group1[i,3] + Group1[i,4])/2 – (Group2[j,3] + Group2[j,4])/2;
                                TravelTimes[k,5] = (Group1[i,3]– (Group2[j,3]);
                                TravelTimes[k,6] = (Group1[i,3] - Group2[j,4]);
                                TravelTimes[k,7] = (Group1[i,4] – (Group2[j,3]);
                                TravelTimes[k,8] = (Group1[i,4]) –Group2[j,4]);
                                k =k+1;
                        }
                }
                Else
                {
                        j = j+1;
                }
        }
}
 TotalEstimates = k -1;

/* Need to add code to write the Travel Time array to a file */
```