Vehicular Ad Hoc Network Microsimulation System for Transportation, Wireless, and Traveler Behavior

BY

JAMES GABRIEL HARAN
B.S., University of Illinois at Urbana-Champaign, 1997

PH.D. DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate College of University of Illinois at Chicago, 2011

Chicago, Illinois

Defense Committee:

Peter Nelson, Chair and Advisor
Abolfazl Mohammadian, Co-Advisor, Civil and Materials Engineering
Jakob Eriksson
Prasad Sistla
Ouri Wolfson
For Shamus and Brendan

that you may explore the boundless possibilities available as a result of

imagination, creativity, perseverance, and determination.
ACKNOWLEDGMENTS

Special thanks are due to the teachers, advisors, family, friends, colleagues, and staff who have helped me during my years of graduate study and research by sharing bits of advice, challenging me to innovate, inspiring new paths of learning, or simply providing a smiling face and helpful hand. Their contributions are far more than academic and helped me truly enjoy my graduate experience.

Most importantly, I thank my wife, Denise, and boys, Shamus and Brendan, who withstood many hours waiting for me to emerge from my office and were a constant reminder of the value of balancing work with life. Their efforts are as much a part of this research as my own.

I thank my mom, Margaret Haran, for her encouragement and guidance throughout my life and the never-ending support and love she has provided. I also thank my dad, Patrick Haran, for inspiring my education with his unquenchable thirst for knowledge, passion for learning, and wonder in discovery. Their dedication to their family will always be remembered and will be a lesson to the many generations which follow.

I thank my siblings Margaret, Ann, Patrick, Michael, John Paul, Tommy, Rory, Kathy, Finbarr, Sheila, Mary, and Brian who all endured the loss of my company during my studies and each provided bits of advice and laughter to help make these years pass quickly. Special thanks are due to my sister Margaret for accompanying me to present my work in Dublin and helping me waste away many opportunities for study drinking tea with my dad in her kitchen and talking about everything but computer science.

I thank my brother-in-law, Harry Barrett, for driving vehicle B during my wireless simulation field analysis calibration experiments thereby helping collect the data for this research.

I thank my father-in-law, Shamus McLoughlin, for his help keeping my house functioning, my mother-in-law, Nancy McLoughlin, for the many meals and hours of babysitting she provided and my sisters-in-law, Rita and Brenda, for the many ways they have been available to help throughout this research.
ACKNOWLEDGMENTS (continued)

I thank my advisors Dr. Peter Nelson for his patience and direction during my studies and Dr. Abolfazl Kouros Mohammadian for providing insight into the practical aspects of transportation engineering and keeping my research grounded in engineering practice. Their advice has been a source of clarity in my education and our work together a welcomed friendship. I have greatly enjoyed our many discussions on topics both academic and not.

I also thank Dr. Jakob Eriksson for challenging me to understand the tradeoffs between simulation and reality in every aspect of modeling and data collection, Dr. Ouri Wolfson and Dr. Bo Xu whose early discussions on implementation helped my future developments achieve the best academic value, and Dr. Prasad Sistla for providing a clear basis for understanding the mathematical theory behind experimental results.

Thank you to Dr. John Dillenburg who helped guide my initial research and provided insight into the importance of coupling theory with practice and Richard Peng Fan who contributed to the preliminary works on clustering algorithms discussed in this thesis. Additional thanks are due to Dr. Jingtao Ma for aiding this research with the development of a Chicago transportation simulation model. I also thank the members of the IGERT Computational Transportation Science program as a set of colleagues with which I, as an IGERT Associate, could share my research, understand and discuss the works of others, and gain exposure to many other aspects of this field.

JGH
# TABLE OF CONTENTS

1 **Introduction** ................................................................................................................. 1
   1.1 Overview ................................................................................................................... 1
   1.2 Ad hoc Networks ....................................................................................................... 2
   1.3 Vehicular Ad hoc Networks ...................................................................................... 8
   1.4 Traffic Simulation ...................................................................................................... 16
   1.5 Wireless Simulation ................................................................................................. 18
   1.6 Agent-Based Systems ............................................................................................... 19
   1.7 Hybrid Simulation ................................................................................................... 20
   1.8 Summary of Dissertation Contributions ................................................................... 21
   1.9 Thesis Outline .......................................................................................................... 22

2 **Related Work** ............................................................................................................... 24
   2.1 Survey of ITS Research Projects ............................................................................. 24
   2.2 Communication Considerations for ITS ................................................................. 28
   2.3 Motivation from Mobile Ad Hoc Networks ............................................................. 30

3 **Clustering Algorithms** ................................................................................................. 31
   3.1 Mobile Ad Hoc Networks ....................................................................................... 31
   3.2 Graph Representation ............................................................................................... 33
   3.3 Maintaining Cluster Stability .................................................................................. 36
   3.4 Lowest-ID Clustering ............................................................................................. 40
   3.5 Highest-Degree Clustering ..................................................................................... 40
   3.6 Overhead and Efficiency .......................................................................................... 41
   3.7 Desirable Features for VANET Algorithms ............................................................ 42
   3.8 Summary .................................................................................................................. 43

4 **Utility-Based Clustering in VANETs** ........................................................................... 44
   4.1 Introduction of Traffic Simulation Model ................................................................. 44
   4.2 Utility-based Approach ............................................................................................. 46
   4.3 Simulation Metrics ................................................................................................... 48
   4.4 Preliminary Experiments and Discussion ................................................................. 50
   4.5 Compound Utility-based Clustering ......................................................................... 53
   4.6 Simulation Study of Utility-based Algorithms ......................................................... 54
   4.7 Summary .................................................................................................................. 58

5 **Simulation and Platform Evaluation** ............................................................................ 59
   5.1 Traffic Simulation Model .......................................................................................... 59
   5.2 Wireless Simulation Model ....................................................................................... 61
   5.3 Integrated Simulator for VANETs ............................................................................ 63
   5.4 Testing Scenarios ..................................................................................................... 64
   5.5 Realistic Traffic Data ............................................................................................... 66
   5.6 Summary .................................................................................................................. 67

6 **Traveler Modeling** ......................................................................................................... 68
   6.1 Foundation ................................................................................................................ 68
   6.2 Location Management ............................................................................................... 72
   6.3 Preference and Activity Modeling ............................................................................. 74
   6.4 Summary .................................................................................................................. 74

7 **System Design and Architecture** .................................................................................. 76
   7.1 JIST / SWANS / SiDnet Architecture ....................................................................... 76
   7.2 VISSIM Traffic Simulation Model ............................................................................ 81
   7.3 General Design ........................................................................................................ 85
   7.4 Initialization and Synchronization ............................................................................ 90
   7.5 Transportation Network Proxy Architecture ........................................................ 92
# TABLE OF CONTENTS (continued)

7.6 Simulation Control and Agent Modeling .......................................................... 97
7.7 Traffic Simulation Service Library ...................................................................... 101
7.8 Traffic Simulation Model .................................................................................... 102
7.9 Smartphone Simulation ..................................................................................... 106
7.10 Summary ........................................................................................................... 108

8 Simulation Controller and Sample Applications .................................................. 110
8.1 Controller Application ........................................................................................ 110
8.2 Preference-Action Model for Behavioral Characteristics .................................... 112
8.3 Active Driver Alert System ................................................................................. 114
8.4 Summary ............................................................................................................. 118

9 Wireless Simulation Field Analysis ...................................................................... 120
9.1 Overview .............................................................................................................. 120
9.2 Propagation and Reachability ............................................................................. 126
9.3 Environmental Effects ......................................................................................... 127
9.4 Transmission Radius .......................................................................................... 128
9.5 Cliff Effect .......................................................................................................... 131
9.6 Summary ............................................................................................................. 131

10 Summary and Proposed Future Work ................................................................. 135
10.1 Summary of Research Contributions ................................................................. 135
10.2 Transit Scheduling Elements ............................................................................. 136
10.3 More Robust Profile Information ..................................................................... 137
10.4 Additional Infrastructure Components ............................................................... 138
10.5 Expanded Traffic Network ................................................................................ 138
10.6 Pedestrian Modeling .......................................................................................... 139
10.7 Additional Transportation Simulators ................................................................. 139

11 Appendix ............................................................................................................. 140
11.1 Modification to JiST / SWANS / SIDnet Libraries ............................................. 140
11.2 Java Framework Upgrade ................................................................................ 143
11.3 Bytecode Engineering Library Changes .......................................................... 143

12 Cited Literature ................................................................................................... 144

VITA .......................................................................................................................... 154
LIST OF TABLES

Table 1. Experimental settings for weighted clustering. .......................................................... 54
Table 2. Subset of services in the TrAITS Traffic Services library. ........................................... 102
LIST OF FIGURES

Figure 1. Traditional cellular network topology ......................................................... 5
Figure 2. Cellular network topology ............................................................................. 6
Figure 3. Wireless ad hoc network topology ................................................................. 6
Figure 4. MANET communication relay to cellular network ........................................... 7
Figure 5. Disconnected MANET relay through cellular network ................................. 8
Figure 6. IVC communication in urban environment ...................................................... 15
Figure 7. Transportation simulation model types ........................................................ 16
Figure 8. Clustering of ad hoc network nodes ............................................................... 37
Figure 9. Cluster changes vs. transmission range ........................................................ 50
Figure 10. Average clusterhead changes vs. speed limit ............................................... 51
Figure 11. Clustering ratio vs. transmission range ........................................................ 52
Figure 12. Clustering ratio vs. speed limit ................................................................. 52
Figure 13. Clusterhead selection ................................................................................. 53
Figure 14. Clusterhead changes vs. range for utility-based algorithms ......................... 55
Figure 15. Clusterhead changes vs. speed for utility-based algorithms ......................... 56
Figure 16. Clustering ratio vs. range for utility-based algorithms ................................. 57
Figure 17. Clustering ratio vs. speed for utility-based algorithms ................................. 57
Figure 18. SWANS++/STRAW modeling Cook County, IL ......................................... 65
Figure 19. JiST Simulation Process (Barr et al., 2005) .................................................. 77
Figure 20. SWANS execution model (Barr et al., 2005) ................................................ 78
Figure 21. Sidnet-SWANS execution model (Ghica, 2010) ......................................... 80
Figure 22. The VISSIM traffic simulator in 3D mode ................................................... 82
Figure 23. Widemann car-following model (Wiedemann, 1991; VISSIM, 2011) ........... 84
Figure 24. TraITS as a VISSIM-SWANS hybrid simulation platform ......................... 87
Figure 25. Basic Object Model for VISSIM API ........................................................ 87
Figure 26. Initialization and processing of TraITS simulation ........................................ 90
Figure 27. Example runtime processing of the TraITS environment ............................. 93
Figure 28. Partial class diagram for VanetSimulation and members .............................. 94
Figure 29. TraITS architecture detailing interface and control components .................. 96
Figure 30. The ITSAgent intelligent and autonomous agent class model ...................... 98
Figure 31. Key components of TraITS architecture .................................................... 99
Figure 32. Additional mobility models available in TraITS platform ............................ 100
Figure 33. The SimpleGPS implementation ............................................................... 100
Figure 34. Aerial photo of Boise, ID (© 2011 Google) .................................................. 103
LIST OF FIGURES (continued)

Figure 35. Boise, ID VISSIM Network. ................................................................. 104
Figure 36. Aerial photo of Chicago, IL (© 2011 Google, © 2011 TerraMetrics). ............ 105
Figure 37. Chicago, IL VISSIM network. ..................................................................... 106
Figure 38. The VanetController class which initializes and manages simulation. .......... 111
Figure 39. Pseudocode for createNode method of VanetController. ......................... 111
Figure 40. The behavior modeling class implementation. ............................................. 112
Figure 41. Agent processing flow for preference-action model. .................................... 113
Figure 42. The relationship between perception and stopping distance. ....................... 115
Figure 43. The active driver alert agent classes. .......................................................... 116
Figure 44. Three-agent processing flow for active driver alert system. ......................... 118
Figure 45. Vehicle A path and signal activity (© 2011 Google). .................................... 125
Figure 46. Vehicle B path and signal activity (© 2011 Google). .................................... 126
Figure 47. Real-world VANET communication breakdown (© 2011 Google). ............... 130
Figure 48. Relative distance measurements between vehicles. ..................................... 132
Figure 49. Successful mutual vehicle connectivity. ..................................................... 132
Figure 50. SWANS simulation following real-world traces. ....................................... 133
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate for signal communication errors</td>
</tr>
<tr>
<td>COM</td>
<td>Component Object Model</td>
</tr>
<tr>
<td>CTS</td>
<td>Computational Transportation Science</td>
</tr>
<tr>
<td>DOT</td>
<td>Department of Transportation</td>
</tr>
<tr>
<td>DSRC</td>
<td>Dedicated Short Range Communication</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IGERT</td>
<td>Integrative Graduate Education and Research Traineeship</td>
</tr>
<tr>
<td>ITA</td>
<td>Intelligent Traveler Assistant</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation Systems</td>
</tr>
<tr>
<td>IVC</td>
<td>Inter-Vehicle Communication.</td>
</tr>
<tr>
<td>JiST</td>
<td>Java in Simulation Time Discrete Event Simulator</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad Hoc Network</td>
</tr>
<tr>
<td>SN</td>
<td>SIDnet, SIDnet-SWANS</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SWANS</td>
<td>Scalable Wireless Ad Hoc Network Simulator</td>
</tr>
<tr>
<td>TIGER</td>
<td>Topographically Integrated Geographic Encoding and Referencing system</td>
</tr>
<tr>
<td>TrAITs</td>
<td>Traveler Agent Intelligent Transportation Simulator</td>
</tr>
<tr>
<td>USDOT</td>
<td>United Stated Department of Transportation</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle to Infrastructure Communication</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle to Vehicle Communication</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad Hoc Network</td>
</tr>
<tr>
<td>VII</td>
<td>Vehicle-Infrastructure Integration</td>
</tr>
<tr>
<td>VRC</td>
<td>Vehicle to Roadside Communication</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wireless Access in Vehicular Environments</td>
</tr>
</tbody>
</table>
SUMMARY

The vehicular ad hoc network (VANET) environment is best understood as the integration of a dynamic multi-modal transportation network, wireless communication, handheld and vehicular computing devices, and software services and applications that provide real-time access to transportation resources and facilitate the efficient coordination of activities. Real implementation of such an environment requires significant investment in transportation infrastructure, improvement to current wireless hardware devices, development of collaborative software systems, and integration of thousands of disparate transportsations sensors. Given the prohibitive cost and logistics of implementing such an environment, the focus of this research is to provide a simulation platform for the rapid prototyping and analysis of the VANET devices using proven methods for simulation from the fields of wireless networking, transportation science and traffic modeling, and computer science.

This research builds upon techniques determined during previous investigations into the development of an Intelligent Traveler Assistant (ITA) which constitutes the fusion of many technologies into a device or series of services which assist a traveler in navigating the transportation network in an efficient and collaborative fashion. The ITA provides an individual user an interface into the transportation features and options available by transportation providers and other ITA users. Some key capabilities previously identified include dynamic ridesharing, multi-modal route planning, dynamic route adaptation, public transportation option identification, and mass transit payment services, but this model may be similarly applied to any transportation or location-aware applications and is more readily realized today in the transit applications available on smartphones and in-vehicle driver assistance and navigation systems.

This research identifies several key infrastructure components of the VANET simulation framework through the design and implementation of an Intelligent Transportation System (ITS) simulation platform capable of modeling the characteristics of multi-modal transportation, wireless communication, and user action or behavior within a federated platform.
collaborative transportation / wireless device or set of services, will require many key features to accomplish the goals within this and previous research. The goal of this work is to identify key components for wireless service, management infrastructure, transportation service infrastructure, ITA interaction, profile management, and routing while developing a robust simulation platform that combines the VISSIM transportation microsimulator with the SWANS wireless network simulator into the Traveler Agent Intelligent Transportation Simulator or TrAITS platform. The TrAITS platform designed and built as part of this research provides a foundation for future research into the characteristics of adoption and use of applications, devices, and services within ITS environments. By providing the basic foundation for follow-up analysis into the identified sub-components and services of the ITA environment this work will aid future research and real-world implementation.
1 Introduction

1.1 Overview

This research outlines the design considerations for implementation of the TrAITS, Traveler Agent Intelligent Transportation Simulator, as a combination of wireless communication, traffic simulation, and application interaction in an agent environment. To aid in the understanding of the discussion in this document, this chapter highlights the high-level concepts motivating the research and provides a background for the technologies that interact in an intelligent transportation environment.

This research builds upon previous investigations into simulation, wireless ad hoc networking, and transportation science to define the essential components of a wireless transportation navigation device and build a hybrid wireless networking / transportation simulator capable of testing the interaction of travelers and infrastructure components and services within an active traffic network. This hybrid simulator has been used to simulate the interaction of travelers, modeled as software agents with various usage and preference profiles, in an active multi-modal traffic network.

The Intelligent Traveler Assistant (ITA) defined in (Dillenburg et al., 2002) was originally envisioned as a dedicated handheld device but has grown in concept to a collection of software components and application services capable of augmenting any mobile platform. These applications coordinate with infrastructure elements, services from transportation providers, and other devices for the purpose of guiding an individual traveler through the complexities of modern transportation with ease and efficiency.

Further chapters detail previous work into clustering algorithms as a basis for wireless communication within vehicular ad hoc networks (VANETs). In this context the term vehicle denotes a wireless enabled mobile node available for simulation within a transportation network. Therefore, it’s important to note that the ITA user, from the perspective of simulation, is considered a mobile node interacting in a vehicle simulation. In other words, once equipped with
an ITA device, people represent a specialized simulation vehicle following transportation routing and movement algorithms specific to human movement. Hence, special focus is given to understanding the characteristics of VANETs as a foundation for simulating travelers and their individualized behavior. Components such as traffic signals and sensors are similarly available for simulation but classified as infrastructure components. Individual user behavior is modeled with an agent-based system providing individualized utility modeling such that the preference or decision making of any individual can be configured and monitored in the TrAITS environment.

TrAITS provides features to allow configuration of the number of equipped nodes within the transportation network to facilitate modeling of adoption scenarios for ITA devices and applications. Because transport movement patterns are based upon interaction of various actors, such as vehicles in car following algorithms, any movement patterns must consider all components of the transportation network, whether or not radio-equipped, to produce realistic movement patterns. To accommodate this mandate, the transportation simulation components model all components of the system using a micro-simulation approach inclusive of all vehicles and all types of nodes.

The remaining chapters of this document outline the background of the ITA concept, highlight the available infrastructure components, and finally outline the architecture of the TrAITS platform. Details regarding the design, implementation, configuration, and calibration of the simulation platform are provided throughout the document for persons interested in leveraging or augmenting this simulation platform for future research.

As previously discussed, this chapter highlights the key considerations of wireless ad hoc networks, VANETs, simulation of traffic and wireless networks, agent based systems, and hybrid simulation thereby forming the foundation for the detailed coverage of the implementation available in future chapters.

1.2 Ad hoc Networks

A wireless ad hoc network is a set of nodes which communicate with one another by establishing a wireless network and maintaining connectivity. The general hardware for an ad
hoc node is availability of both a transmitter and receiver capable of interacting over one or more protocols, typically IEEE 802.11. These nodes face the same conditions that affect all wireless communication including noise, interference, and fading and typically collaborate to form self-organizing networks where each node acts as both a host and router such that connectivity results from the availability of multiple paths of network communication between any two nodes through a series of single-hop transmissions among network members. This topology provides a true distributed management of the network among the nodes which results in high redundancy benefiting availability but hindering bandwidth and throughput. Typically, it is assumed that nodes in an ad hoc network are mobile, although the degree of mobility may vary greatly between nodes. A typical implementation may simply require the interaction of essentially static devices within a local environment but more sophisticated implementation such as those in use by military operations may require collaboration of highly distributed fast moving nodes with major topology and line-of-sight challenges. As a further challenge, nodes in an ad hoc network may disappear or join randomly as devices are powered on/off during normal operation or become obscured by obstacles or distance. Thus one goal of ad hoc networks is to provide rapid access to network resources under the highly unpredictable nature of mobile nodes. These networks typically have lower bandwidth than their wired counterparts leading to a desire to reduce unnecessary communication traffic by efficiently routing and recovering from network changes. A further challenge exists in the efficient use of power within these devices since mobile equipment is often battery powered and inefficient use of this equipment can lead to more rapid loss of power thereby disrupting the network infrastructure. Likewise, any wireless node used as a relay or routing point potentially suffers a further power drain which is compounded by inefficient network algorithms.

Advances in wireless infrastructure in personal and industrial devices coupled with widespread adoption of the 802.11 protocols during the 1990s led to a more extensive interest in wireless communication and the principles of creating, maintaining and leveraging wireless ad hoc networks from both industry and academia. Since this time the 802.11 family has grown to include a multitude of standards, each with differing characteristics for the maintenance, security,
and stability of network connections. Most recently the development of the 802.11p (IEEE, 2011) standard as an outgrowth of the Wireless Access in Vehicular Environments (WAVE) (Fisher, 2007) research and implementation of the dedicated short-range communications (DSRC) (DSRC, 2010) class of standards has helped expand the range, reliability and possible uses of wireless technology in mobile or specifically vehicular environments.

Additionally, use of wireless and cellular networks in recent history exposed the intricacies of wireless communication that compose the many open problems of the field. Increasingly, devices are equipped with multiple methods of wireless communication providing platforms upon which multi-protocol wireless ad hoc networks can be built which collaborate among IEEE 802.11, IEEE 802.15, 802.16 (WiMax, Mobile Multi-hop Relay), Bluetooth, or cellular protocols, among others, in a single network. Such a device may regularly adjust from using any of these or other protocols to determine the best path for relay of communication packets.

During the next decade, many independent mobile devices providing a multitude of services for standalone and collaborative interaction among human and intelligent automatic services will be deployed into the next generation of wireless networks. Similarly, self-contained sensors are increasingly wireless enabled and available for integration with other devices for monitoring and alerting travelers to pertinent information. Due to the highly mobile feature of these networks and frequent topology changes experiences as a result of regular movement throughout the network, the traditional centralized control paradigm is no longer feasible because it not only requires a pre-established infrastructure of relay base stations but also generates a large network communication overhead thereby decreasing the overall network throughput. Therefore, the distributed peer-to-peer network communication paradigm of Mobile Ad Hoc Networks (MANETs) (Corson and Macker, 1999; Chiang and Gerla, 1997) has become the most appropriate solution to deal with large dynamic wireless network.

Figures 1 and 3 illustrate the differences between traditional cellular networks and wireless ad hoc networks. In traditional cellular networks, the whole communication field is partitioned into cells, each of which is serviced by a series of fixed base stations which are interconnected with each other. Establishment of cells requires large investment in the construction and maintenance
of the base station. Mobile objects within a cell communicate with the closest fixed base station and regularly change base stations as they travel throughout the network. These base stations utilize predefined infrastructure networks to provide inter-network communication and access to services within the network. Physical locations not reachable by a fixed base station are disconnected from the network. Such an infrastructures costs increase dramatically when the features necessary to provide redundancy in the case of a base outage are considered. Without well-established redundancy infrastructure traditional cellular infrastructures may also be prone to accidental or intentional outages.

The configuration of towers and cell within a cellular network is typically modeled as network of hexagons with hexagon vertices representing the locations of base stations or cellular towers. This model, shown in Figure 2 reflects that the assumed circular coverage area. Note that the coverage areas of cell towers overlap to allow basic redundancy. Cellular devices typically connect to the tower with the best available signal which reasonably implies the closest tower to the device. Characteristics of normal wireless radio signal loss and propagation do apply such that local towers may become unreachable and alternatives may be used.
Within wireless ad hoc networks the mobile objects communicate with each other without going through a base station. This occurs either directly, if the mobile objects (nodes) are in close proximity, or through other nodes acting as routers for longer distances. These self-organizing networks can be established very quickly in a variety of situations and have the ability
to bridge many problems of topology and connectivity that plague traditional networks. Nodes within the network are only completely disconnected if they cannot reach any other node for communication. This model allows for the formation of disconnected subnets when a group of nodes remain coupled with one another but do not have access to nodes outside of their local connectivity. These subnets, although disconnected from a wider array of service, still provide the individual with access to valuable information in a VANET environment through the use of peer-to-peer (P2P) state exchange and alerting. As an additional benefit in this situation, network connectivity may be further aided by allowing the wireless network to communicate through base stations and other fixed relay points to bridge disconnectedness from peer nodes either through dedicated infrastructure backbones such as cellular networks or through general purpose networks such as the Internet.

![Figure 4. MANET communication relay to cellular network.](image)

Figures 4 and 5 highlight two scenarios where wireless connectivity is aided by the availability of both cellular and 802.11 based wireless technology in a cooperative manner. In the first, Figure 4, the loss of a series of cellular base stations would typically fragment the cellular network and leave users without service. The use of an ad hoc relay network instead allows the users to
relay traffic through local ad hoc devices back to the nearest base station. As an alternative, local disconnected ad hoc networks are able to relay through base stations (Figure 5) to maintain connectivity with one another.

![Disconnected MANET relay through cellular network.](image)

As availability of low cost wireless devices increases and people, services, and machinery become reliant upon communication networks, the advantages of an ad hoc network with the ability to bridge accidental or intentional infrastructure outages and provide reliable communication channel cannot be understated.

### 1.3 Vehicular Ad hoc Networks

A Vehicular Ad Hoc Network (VANET) is an integrated communication network leveraging various wireless technologies in a collaborative fashion for the purpose of inter-vehicle (IVC) and vehicle-to-roadside (VRC) or vehicle-to-infrastructure (V2I) communication. As deployment of
intelligent vehicles becomes more widespread. Intelligent Transportation Systems (ITS) related applications in VANETs, built upon in-vehicle communication and computing devices, have become increasingly popular and more specialized. This increased popularity has brought increased focus on the challenges of the support of reliable and efficient communication in large scale networks and is the basis for the field of Computational Transportation Science (CTS). One approach to efficient network utilization is to reduce network overhead is to abstract a hierarchical clustering structure within the network. Therefore, an analysis of these clustering methods is provided within the next two chapters of this document as a background into previous work of the author and motivation for the simulation platform defined in later chapters. The analysis of VANET clustering methodologies and discussion of various improvements has the goal of reducing the overhead of wireless communication and providing an efficient hierarchical network structure. This work defines various clustering algorithms and a cluster based broadcasting algorithm for VANETs application and addresses both theoretical analysis and simulation study. The context for this research concerns the exploration of the use of IVC and VRC to disseminate information in VANETs and the modeling of this research is performed within the simulation environment.

Short-range communications (e.g., DSRC), designed to facilitate IVC and VRC, are important component of ITS (Werner, 2004). Many proposed safety applications, such as cooperative driving and accident alerting, (Morsink et al., 2002; Xu et al., 2004, CVIS, 2011) rely on short-range communications to efficiently operate. Services such as toll collection, traffic light control, parking enforcement, emission control, and vehicle maintenance may also leverage short-range communication for interaction with their local peers. This potential justifies the extensive recent research into the performance (Singh et al., 2002; Moske et al., 2004; Ott and Kutscher, 2004) and reliability (Torrent-Moreno et al., 2004; Xu et al., 2004; Yin et al., 2004) of short-range communications. In ad hoc networks, short-range communications may be utilized to disseminate information over a large area offering an alternative to the much more expensive infrastructure based services. This deployment, however, must first overcome several
instabilities resulting from topology changes, line of sight restrictions, low penetration ratios, and unreliable wireless channels.

Interest in the exploitation of advances in mobile computing and wireless communications in surface transportation systems (Bechler et al., 2006; Kutzner et al., 2003; Morsink et al., 2002) has grown in recent years such that the modern vehicles are often equipped with communication and computing capabilities which enable better traveler-based services. This opens up possibilities to study how to enhance existing transportation systems to improve driving safety, reduce traffic congestion, reduce emissions, etc. The advent of driver-assistance services such as BMW Assist, Mercedes TeleAid, or GM OnStar and in-vehicle communications and maintenance services such as Ford Sync has also helped drive enhancement of vehicular computing platforms into networks coupling vehicle sensors and state information, Global Positioning System (GPS), and wireless communication, accommodated by either a built-in cellular device or by coupling the driver’s device using Bluetooth or wired integration.

There is a wealth of desirable applications for ad hoc communication ranging from emergency services such as warning alerts or distribution of traffic and road conditions to personal applications such as social networking, file sharing (Lee et al., 2007), and distributed games. As a result, many vehicle manufacturers and their suppliers are actively supporting research on how to integrate mobile ad hoc networks into their products. Many of these applications extend to location-aware services which allow consumers to identify local services and allow marketers to target based on current GPS location thus providing a means by which ITS devices have become the foundation for new business services.

Possible applications for VANETs can be generally categorized into safety and non-safety applications. Safety applications include collision avoidance, emergency accident warning and cooperative driving (Morsink et al., 2002; Xu et al., 2003). Non-safety applications include traffic information propagation (Ziliaskopoulos and Zhang, 2003; Wischhof et al., 2003), on-demand video streaming, Internet access (Bechler et al., 2006), tourist information, cooperative ride sharing and entertainment, among others. The Vehicle-Infrastructure Integration (VII) of USDOT’s ITS initiative (DOT, 2007) is attempting to capitalize on intelligent vehicles by
encouraging private-public partnerships through the deployment of wireless communication devices into privately-owned vehicles and infrastructure devices along the arterial and highway networks of the existing transportation system (Werner, 2004). According to research, such a deployment has the potential to improve safety, reduce congestion (Morsink et al., 2002; Xu et al., 2004) and provide new public and commercial services (Kutzner et al., 2003; Wischhof et al., 2003; Ziliaskopoulos and Zhang, 2003) while reducing the overall adoption cost and timeline.

According to previous studies, these opportunities become feasible when adoption rates show approximately 10% of the vehicles equipped with communication and computing resources (Werner, 2004).

Current traffic monitoring infrastructure incurs fairly high costs. For example, surveillance cameras at road intersections are expensive to deploy and maintain. Intelligent vehicles offer the potential to reduce infrastructure dependent costs such as device deployment, system operations and maintenance by integrating the devices within standard vehicle hardware such that the cost is distributed to consumers and reduced by deployment volume. Communications between vehicles allow coverage to extend beyond the extent of roadside infrastructure. Additionally, subject to privacy and security considerations, these devices offer the potential for much more timely information (e.g., on-road vehicle activity) than would otherwise be possible, creating better opportunities to improve the current transportation system and support a variety of interesting applications. Moreover, computing systems inside vehicles and in handheld devices enable the personalization of information based on the needs of individual travelers based on localized profile specifications and common infrastructure services.

Wireless communication is the key technology for the aforementioned applications as illustrated by the U.S. Federal Communications Commission’s (FCC) allocation of the 5.85-5.925 GHz portion of the spectrum to IVC and VRC under the umbrella of dedicated short-range communications (DSRC) (FCC, 2004; DOT, 2006; Fisher, 2007; DSRC, 2010), a communication service for safety and private applications and the active work by ASTM International (ASTM, 2010) and the Society of Automotive Engineers (SAE) to establish guidelines and standards regarding the performance requirements of DSRC in vehicular applications (SAE, 2011). DSRC
is best represented today in the IEEE 802.11p standard (IEEE, 2011; Rohde and Schwarz, 2009) which extends the 802.11 family of protocols to include a specific focus on the challenges of V2V and V2I communication and adds support for wireless local networks within vehicular environments. DSRC when coupled with cellular communications provides network connectivity to moving vehicles. Aided by DSRC and wireless devices, a vehicular network may form between vehicles and roadside communication infrastructure without a centralized architecture while also using roadside infrastructure for relay to wired networks via the Internet. At a minimum, each such vehicle is equipped with computing devices, wireless communication devices, and a GPS device for the purpose of tracking its spatial and temporal trajectory. Additional integration may also provide digital maps, sensors for reporting accidents or monitoring environmental conditions, and parameters or vehicle state and component activity.

Vehicular networks differ from other wireless networks such as sensor networks primarily because the devices are bound to vehicles and therefore follow the movement patterns of vehicular networks. The percentage of the vehicles on the road with these capabilities will gradually increase as manufacturers and consumers adopt the technology.

Current technologies for establishing vehicular networks include Wireless Wide Area Networks (WWAN), Wireless Metro Area Networks (WMAN), Wireless Local Area Networks (WLAN) using roadside access points, and wireless ad hoc networks using IVC communications. These technologies offer different cost and performance tradeoffs and network construction options. These alternatives help establish a multi-protocol architecture for VRC and IVC communication without reliance on fixed centralized infrastructure.

A pure IVC network is an infrastructureless network consisting only of intelligent vehicles. Vehicles are typically equipped with wireless communication devices and can exchange information with other vehicles within their communication range leading to the creation of ad hoc wireless networks that can disseminate information in a P2P fashion. This deployment not only offers the benefits of P2P mode (no centralized operation), low cost and easy deployment (no pre-existing infrastructure is required), but also is necessary and ideal for many applications (e.g., cooperative driving, travel assistance).
VANETs, an outgrowth of traditional MANETs, provide one of the basic network communication frameworks for Intelligent Transportation Systems (ITS) applications. Essentially, it is a subclass of Mobile Ad Hoc Network (MANET) specified in (Corson and Macker, 1999). A MANET consists of ad hoc mobile nodes equipped with wireless communication devices and capable of autonomously recognizing other ad hoc nodes within their communication range. They can create communication links spontaneously with all neighboring nodes and execute a packet forwarding procedure thereby allowing each node in an ad hoc network a mechanism for communication with nodes beyond its own communication range via relay across multiple network node hops.

(Corson and Macker, 1999) lists three primary characteristics of MANETs:

- Topology changes: Random and rapidly changing network topology because devices (nodes) move freely and are regularly added or removed.
- Bandwidth constraints: Wireless connections rely on link conditions and the distance between nodes which result in bandwidth limitations.
- Energy constraints: Nodes have a limited power supply, e.g., a battery.

As a special type of MANET, some unique characteristics (Tian and Rothermel, 2002; Wu et al., 2004; Haran et al., 2005) differentiate VANETs from MANETs including:

- High mobility: Vehicles move at much higher speeds than nodes in MANETs.
- Constrained movement: Due to the transportation rules of the car following, lane changing and gap acceptance on the road network, vehicle mobility is more predictable. Vehicles are not only constrained by the road itself, but also by traffic congestion and regulations, e.g., maximum and minimum speeds.
- Topology constraints: Unlike ordinary mobile nodes, vehicles do not randomly move within a physical space, but instead follow topological constraints set in place by the physical road network in a generalized two-dimensional space.
• Large scale: A VANET can include a very large road network, such as a metropolitan area, and potentially thousands or even millions of equipped vehicles whereas a MANET often has a fixed number of nodes moving inside a limited small space.

• Minimal power constraints: Vehicles can easily provide the power required for wireless communication devices and are not seriously affected by the addition of extra weight for antennas and additional hardware. Because of this many supplementary devices such as in-vehicle computers and on-vehicle sensors can be considered.

• Network connectivity: an inter-connected network is usually assumed in ad hoc networking research. However, (Dousse et al., 2002) showed that the probability of the connectivity decreases with distance for networks such as VANETs. So VANETs are more likely to be partitioned. This is particularly true at lower penetration ratios where the number of intelligent vehicle is small. This observation is also confirmed by analytical models and simulation studies (Wu et al., 2005b).

• Other features: Compared to mobile nodes in MANETs, vehicles generally have accurate knowledge of their own geographical position, a clear view of its local configuration, and a computing capability for large amount of data, etc. Thus, many of the issues making deployment and long-term use of ad hoc networks problematic in other scenarios are less relevant in VANETs.

These characteristics have many implications on algorithm design for VANETs. First, high mobility and topology constraints can be exploited to facilitate information dissemination (Wu et al., 2005b). Second, due to the partitioned, highly dynamic nature of VANETs, large-scale and complicated structures (e.g., node-based trees) are undesirable; rather, clustering algorithms (Meguerdichian et al., 2001) based on vehicles interacting with neighbors are preferred. Third, with the availability of more accurate information such as geographical position, direction, and driving behavior etc, VANETs provide opportunities for improving efficiencies (Benslimane and Bachir, 2003) of communication and information distribution. Finally, large-scale, high mobility,
traffic rules and virtually unlimited power introduce completely different designs for optimal communication structures in VANETs (Fan et al., 2007).

Figure 6. IVC communication in urban environment.

Figure 6 illustrates an IVC scenario one can easily imagine for the near future. The notification of an accident and resulting traffic jam is passed among nearby streets using the fully distributed wireless infrastructure such that each upstream driver is aware of the traffic jam and may leverage that knowledge to choose alternative routes. This information relay covers various vehicle types and transportation modes and may also consider a particular driver’s behavior or planned travel route. The desire of providing such state awareness of the transportation network is that each vehicle within the network can adapt to updated information, either with or without driver interaction, to reduce the occurrences of traffic accidents or bottlenecks, cut transit costs from fuel, emissions, or time lost, and provide a more stable and fluid network flow.
1.4 Traffic Simulation

Transportation traffic simulations have been categorized as either macroscopic, mesoscopic, or microscopic. The determination of type depends on the representation of many parameters including traffic flow within the model, configuring detail, and level of statistics detail available. When evaluating the various types of simulation models special attention was given to determining which model would provide the most realistic movement of VANET nodes while still maintaining individual node control such that behavior and traffic routing aspects of a single traveler could be adjusted. Figure 7 illustrates the differences between model types.

**Microscopic:**
Individual actions determine the network flow. These may be autos, people, trains, or other modes of transit.

**Mesoscopic:**
Platoons of vehicles are configured and studied in aggregate.

**Macroscopic:**
Traffic within zones is modeled as network density or flow without individual behavior.

Figure 7. Transportation simulation model types.
In a macroscopic model, system parameters such as network density or traffic flow, i.e. number of vehicles per time unit entering the system, are used to compute the distribution of traffic in a transportation road network by simply ignoring the expression of individual vehicular behavior. Because macroscopic models do not provide facilities for modeling or adapting the behavior of an individual driver they are typically not well suited for VANET applications. These models typically use broad rules for defining the flow on individual network segments or zones based on calibration data collected for the zone boundaries and assumptions which model travel along routes as the action of compressible fluids in fluid dynamics. Simulation of the movement of people is not available. Multiple models of transit may, however, be provided.

Mesoscopic models such as METROPOLIS, DynaNIT, and Dynasmart analyze platoons of vehicles and their characteristics as a single unit to study traffic flow. These models focus on the how vehicles behave under certain traffic circumstances but without transparency into the actions of an individual vehicle or the capability to effect change on the vehicle’s characteristics or driver’s behavior. Instead, aggregate travel times and speeds are used to determine the characteristic of flow throughout the network.

Microscopic simulation, or micro-simulation, provides a more realistic traffic representation by showing individual vehicle movement for simulations based on the knowledge of velocity, position, and acceleration of each individual. These models focus on simulating the movement of each individual vehicle traveling a road network by applying models for car-following, lane-changing, and other behaviors, typically determines as a result of psycho-physical research into user behavior. Microscopic models allow the researcher to access the characteristics and sensors of any vehicle within the network as well as modify these characteristics in simulation time. Microscopic models also allow configuration of vehicle types to help model different types of drivers within a network. More advanced microscopic models extend their simulation to multiple modes of transportation including the movement of people.

Traffic simulation is further discussed later in this document as the various properties of traffic simulation are detailed with respect to clustering algorithms, performance evaluation, and finally simulator design. From a high level, the atomic-level modeling of microsimulation is an ideal
counterpart to traditional node-based wireless network simulation. This type of simulation, however, is resource intensive and requires much more extensive investment in hardware for more detailed analysis. Microsimulation models such as Paramics and VISSIM accommodate this overhead by allowing distributed collaboration of the simulation executables. Some models, such as VISTA provide scalability to supercomputing platforms. Some models, including VISSIM, have accommodated the overhead by abstracting a microsimulation base with meso- or macro-simulation. Microscopic simulation provides the detailed control necessary to model the interaction of individual vehicles or people with traffic signals and each other in a realistic manner. Therefore microscopic is the preferred method for VANET and ITA simulation and discussed further throughout this document.

Traffic simulators vary in the variety of vehicles available for simulation. Microsimulation platforms typically allow modeling of different vehicle types in the simulation environment. For example, Paramics, VISSIM, and VISTA all provide the capability to model specific vehicle models by implementing a simple vehicle interface and configuring parameters such as acceleration rate, car following rules, number of passenger seats, and fuel consumption. Building upon these interfaces, many types of vehicles may be modeled within a single simulation using the common modeling framework. VISSIM, the principle simulator used in this study, models vehicle types such as buses, ferries, trains, automobiles of various types, and even people in an attempt to gather the most accurate simulation data for transportation planning.

1.5 Wireless Simulation

Like traffic simulation, a number of environments exist for modeling the behavior of wireless ad hoc networks and the many available algorithms and protocols involved. The most common of these models is the open source ns-2 (NS2, 2011), but various other simulators such as SWANS, Qualnet, and Opnet provide different capabilities with different usage considerations. Each environment allows users to model standard network protocols such as IEEE 802.11, IEEE 802.15, 802.16, or Bluetooth with various tradeoffs of configuration, scalability, and speed.
The majority of existing wireless simulation tools model random node movements in a fixed field area. This model is adequate for modeling worst-case movement scenarios in ad hoc wireless networks but has proven inadequate for the analysis of the constrained and somewhat predictive nature of transportation studies. Attempts to build upon these network simulators to model inter-vehicle communication often fail to adequately model the interaction of vehicles with one another or the transition of mobile nodes to other states as they park or leave the network. Car following and lane changing behavior is typically absent because the core design is the evaluation of wireless networks under linear movement constraints. These models also typically do not provide the sensor and actuator control features that is available in most traffic simulations to allow transportation scientists to reconfigure live simulations through adjustments to signaling frequencies or collection of loop sensor data. Finally, multi-modal transportation simulation on these platforms is typically not possible. These restrictions leave the contribution of IVC research based upon these platforms uncertain or difficult to understand.

To accommodate the shortcomings of these wireless platforms, other research has attempted to model the movement of vehicles within the network through a series of vehicle movement trace inputs. This approach facilitates the accurate recreation or modeling of recorded vehicular movements, but is inadequate for modeling the dynamic nature of a true VANET environment. Put simply, a principle goal of ITS research is to understand the effect of enabling intelligent vehicles and offering driver assistance services, a model based solely on historical records or movement states precludes driver behavior adjustment and such effects.

1.6 Agent-Based Systems

The concept of an agent or agent-based system is well understood and studied within many fields of computer science with special focus in the study of game theoretic algorithms and automated systems. The general concept is intuitive and easily understood as the interaction of multiple software components in an effort to maximize some benefit represented with a utility function. Many simulation systems exist for modeling the interaction of intelligent agents such as JADE, NetLogo, Swarm, Ascape, Repast, and MASON. This research will not focus on
implementing any specific framework and will instead implement a conceptual agent system upon
the application framework offered within the Swans simulator. An important distinction should be
made, however, between agents working to maximize a shared utility function versus those
attempting to maximize a personal utility goal.

Within a shared-goal or collaborative agent system the agents seek to satisfy a common goal
my maximizing some shared, known utility. In an individualized agent system the utility for each
agent may be based on a common perceived goal or method for evaluating utility, but is tuned for
the features or preferences specific to that agent.

In this research, interaction between multiple infrastructure components, vehicles, and ITA
agents follow the common interaction patterns of autonomous agents in an agent-based
simulation. Automated vehicle systems may be understood as a collection of multiple
independent agents for a specific purpose of reporting status to requestors. Although this may
not fit the traditional utility-based model, these service-components may interact and collaborate
on a vehicle scale to optimize the driver’s efficient use of vehicle resources. Furthermore,
interaction between an individual ITA participant and any common infrastructure services is
performed with the goal of optimizing the use of the transportation infrastructure and reducing
overall travel times. The key focus of this research is the modeling of ITA users exhibiting
individualized agent utility based upon profile information for the purpose of realistic user behavior
modeling in an ITA scenario. This approach will allow multiple ITA users to interact while
following user-specific goals based on patterns for normal ITA usage.

1.7 Hybrid Simulation

Hybrid simulation involves the integration of multiple standalone simulation components into a
coordinated simulation platform for a purpose not satisfied with either system individually. In its
simplest form, hybrid simulation may involve the interaction of multiple simulation components in
a scripted form utilizing either UNIX shell applications or simulation tools such as Simulink. More
detailed hybridization however has been performed in multiple fields for the purpose of leveraging
the best possible simulation subcomponents.
Recent hybrid simulation research (Lochert et al., 2005; Killat et al., 2007; Schroth et al., 2005) focuses on the coordination of VISSIM and ns-2, this project integrates VISSIM with the SWANS wireless ad hoc network simulator.

Studies on SWANS (Schoch and Kargl, 2007) show it provides results comparable to ns-2 and may scale more efficiently while offering better interoperability with the desired application and infrastructure components of the transportation infrastructure model. This research leverages the considerations of these previous investigations while building a uniquely suited hybrid platform for the purpose of ITA simulation and interaction with infrastructure components and services.

Methods of integration range from full integration of the simulator source code, script based integration using data export and imports, to truly coupled simulators based on a shared messaging platform (Sommers et al., 2008). The simulation design of this research provides offers bi-directional communication between the VISSIM and SWANS simulators such that each environment maintains an up to date record of the other’s activity and vehicle positions. This model also facilitates the modeling of environment-specific components such as base relay towers in the wireless simulator or stoplights in the transportation simulator without the overhead of translating foreign concepts between models.

TrAITS, further described throughout this document, is aided by the availability of an open platform for wireless simulation such that research may augment the environment to model new methods of wireless communication.

1.8 Summary of Dissertation Contributions

This work explores several issues concerning the analysis and design of algorithms for use in VANETs and details key considerations for any researcher seeking to analyze the aspects of ITS through the use of simulation tools. This study is intended not as a comprehensive coverage of the topics of ITS or VANETs, but rather as a foundation for researchers seeking to perform additional work in the interesting aspects of ITS detailed within this document. As a basis for such research the resulting source code, examples, test cases, and data sets are available from the author.
The contributions of this research are as follows:

1. Design of the Traveler Agent Intelligent Transportation Simulator (TrAITS), an integrated multi-modal simulation environment for the study of algorithms related to ITS, VANETs, and CTS.

2. Creation of an agent-based simulation model which provides a foundation upon which research into individualized aspects of traveler behavior may be modeled and analyzed.

3. Comparison of the characteristics of VANET simulation and real environments with detailed analysis of the challenges posed to wireless communication in vehicular situations.

4. Design of alternative routing / route request algorithms based upon the characteristics of traffic movement and verified using simulation studies.

1.9 Thesis Outline

The remainder of this paper is organized as follows. Chapter 2 outlines some related work in the study of VANETs including analysis of IVC and VRC. Chapters 3 and 4 detail considerations for designing clustering algorithms in VANETs and previous work by the author to propose and analyze alternative methods for clustering based upon the aspects of vehicular networks. Chapter 5 provides a foundation for the analysis of simulation environments for vehicular, wireless, and integrated platforms as a basis for the detailed coverage of the subject material and architecture designs in later chapters. Chapter 6 outlines some previous work into the ITA and the evolution of the ITA into a collection of traveler services offered through a number of devices and applications while leveraging the data available in transportation networks. Chapter 7 discusses the design and architecture of TrAITS as the key contribution of this research. Chapter 8 details a simulation driver example and sample ITS applications designed to run on TrAITS. Chapter 9 details calibration and validation experiments conducted in a small scale VANET environment developed by the author to validate the properties of the proposed simulator and guide configuration of experiments. Finally, Chapter 10 offers some suggestions for future
enhancements and studies. The Appendix lists some specific changes made to the SWANS platform to correct issues, enhance performance, and enable the integration with VISSIM.
2 Related Work

2.1 Survey of ITS Research Projects

The availability of low cost cellular and wireless consumer devices as well as the availability of smartphone environments for the deployment of ITS applications has uncovered an enormous untapped market for applications and services based upon VANET technology. This chapter provides an overview of current VANET research as and how they relate to the study of ITA simulation. These studies cover various subfields such as cooperative driver assistance, emergency response, mobile sensing, and automated driving and detail how VANETs have developed from the more traditional MANETs and helped form the motivation for much of this research.

Within the subfield of cooperative driver assistance, the European project CarTalk 2000 (Reichardt et al., 2002; Fan, 2007) investigates problems related to driver safety and comfort utilizing IVC and VRC. In this study, VRC is used to relay vehicle communication with fixed wired networks (Ott and Kutscher, 2004). VRC interaction does not eliminate the challenges of VANET communication, however, since interaction of a vehicle with a fixed roadside base station may still involve a significant difference in velocity or large relative velocity. The advantage may come from having one fixed base point and leveraging the velocity knowledge of the mobile node to reduce communication problems. From this perspective, VRC interaction models that of traditional cellular networks. CarTalk 2000 has proposed various algorithms and protocols for the interaction of wireless devices in networks with highly dynamic topologies. Car Talk 2000 also collaborates with the FleetNet (FleetNet, 2009; Franz et al., 2001) project for the analysis of IVC interaction.

FleetNet, a project researching multi-hop IVC platforms, studies cooperative driver assistance programs, efficient traffic flow, increased driver safety, and mobile Internet services via IVC-VRC interaction across multiple hops. This platform employs location awareness and position data (Bechler et al., 2006) to efficiently route data among vehicles in the network and leverages
tunneling through roadside access points when multi-hop communication would be inefficient (FleetNet, 2009). Network on Wheels (NOW) (NOW, 2008) continues the FleetNet project with the goal of investigating issues related to protocol and data security within IVC.

Mobile sensing is the focus of the CarTel project which aims to gather, interpret, and deliver data from sensors mounted on vehicles (Hull et al., 2006; Fan, 2007). In addition, CarTel utilized the data gathered by this interaction to visualize the transportation network’s state at varying degrees of abstraction. In this model, an embedded on-board vehicle computer interfaces with local sensors, processes the collected data, and delivers it to an Internet server where follow-up analysis is performed such that additional services may be provided to users. CarTel uses a combination of WiFi, Bluetooth, and cellular connectivity to provide high availability of the service while shielding the switching between these connection modes from the user (CarTel, 2011; Bychkovsky et al., 2006). This protocol-agnostic approach allows for redundancy in the communication backbone. CarTel has the added advantage of providing open source software and hardware diagrams utilizing readily available components to allow any group to model their own CarTel network. Within the CarTel project is CafNet, a carry-and-forward protocol for bridging disconnectedness in the network by allowing vehicles to deliver messages to disconnected sensors on behalf of the sender. All information collected by CarTel devices is logged into a central database such that trace information from the vehicle’s sensors may be visualized along with a map of one’s travel route (CarTel, 2011).

CarNet is an application for large VANET systems that scales well fixed network routing infrastructure. It supports IP connectivity for applications monitoring highway congestion, tracking vehicles, and location-aware points-of-interest discovery (Morris et al., 2000; Fan, 2007) through a distributed location service and geographic routing which aspires to scale to hundreds of thousands of nodes efficiently without static hierarchical techniques such as cellular networks. The main benefit of CarNet’s approach is the simplicity of its algorithms for dealing with topology changes in the network that prompt routing adjustments. Simplicity in algorithm design is a key indicator of adoption and implementation success for network protocols.
Self Organizing Traffic Information System (SOTIS), a subproject of FleetNet, deals with the disadvantages of a centralized traffic news service (Eckhoff et al., 2011; SOTIS, 2009; Wischhof et al., 2003; Fan, 2007) by utilizing an aggregate and forward technique to gather transportation information from neighboring vehicles and relay the critical information to other vehicles. This way, the overall information overhead is reduced without requiring central aggregation services to provide up-to-date information. Rather than focus on relaying this information to a news service where it could be provided in an aggregate form to all vehicles in the network, SOTIS allows each vehicle to automatically gather all relevant information by analyzing the communication broadcasts of vehicles within some specific radius of interest. Each SOTIS participant periodically broadcasts its knowledge about the traffic situation (e.g., its velocity, emergency information, etc.) to other vehicles within its transmission range which in turn evaluates and stores the information. This aggregate information is used to create a traffic profile of the surrounding area and is also included in the next periodic broadcast. By this combination of autonomous analysis and broadcasting procedures, each SOTIS equipped vehicle can create a detailed map about the driving conditions in the surrounding area (Wischhof and Rohling, 2005; SOTIS, 2009).

TrafficView (Nadeem et al., 2004; Fan, 2007), a part of the e-Road project, defines a framework to disseminate and gather information about the vehicles on the road. Using this system, an individual driver may gather information about road traffic in areas out of normal sight or obscured by environmental conditions such as foggy weather or storms for distances of several miles. A key component of this implementation is the Vehicular Information Transfer Protocol (VITP) which provides traffic related information to drivers by leveraging vehicle sensors and on-board GPS navigation devices (Dikaiakos et al., 2005).

Many projects focus on the implementation of safety applications for drivers and road conditions. Preventative and Active Safety Applications Integrated Project, PReVENT, (Schulze et al., 2005; PReVENT, 2011) is a European automotive industry activity co-funded by the European Commission to contribute to road safety through the development of preventative safety applications that assist drivers in avoiding accidents or mitigating the damage from these
accidents. Utilizing in-vehicle systems, PReVENT is able to sense the significance of the traffic danger and alert the driver while considering the driver’s location and vehicle state (Zang et al., 2005; PReVENT, 2011). In addition to standard vehicle hardware and navigation systems, PReVENT seeks to leverage sophisticated sensing equipment such as laser-range scanners, radar sensors, 2D and 3D visualization, and driver monitoring devices to increase traveler safety. The PReVENT project has numerous subprojects including RESPONSE 3 which seeks to develop a code of practice for market introduction of driver assistance products, ProFusion and ProFusion 2 which focuses on understanding the needs of reliable sensor fusion applications, and PReVAL which assesses the technical performance of PReVENT, among others geared at promoting driver visibility and alert systems for car following, navigation, and intersection safety (PreVENT, 2011).

The SAFESPOT project (Brignolo, 2006; SAFESPOT, 2011), co-funded by the European Commission, develop solutions for enhancing the driver’s perception by gathering information from VRC and IVC through cooperative network interaction. This effort focuses on development of a Safety Margin Assistant (SMA) which provides a safe distance for car following in the network for the prevention of traffic accidents. The goal is to provide road safety improvements through the cooperation of intelligent vehicles and road networks to extend the response time and distance for drivers by alerting them of potentially dangerous situations through the SMA application (SAFESPOT, 2011). The SAFESPOT architecture includes on-board vehicle computers, central transportation safety service providers, and intelligent infrastructure components available as roadside devices. Like PReVENT, SAFESPOT encompasses many subprojects covering topics such as infrastructure sensing, business models, architecture, and in-vehicle assistance.

WATCHOVER, also funded by the European Commission Information Society Technologies and concluded in 2008 (WATCHOVER, 2011), sought to prevent accidents involving participants such as pedestrians and bicyclists in transportation networks through the use of an on-board computer and sensor module at-risk travelers carry to alert drivers through short range communication and vision sensors including laser range finders and microwave radar (Andreone...
et al., 2006; WATCHOVER, 2011). WATCHOVER summarized that the application of such technology could be accomplished using either IEEE 802.15.4, available within the network sub-components of TrAITS, Radio Frequency Identification (RFID), or Ultra Wide Band Radio (UWB) (WATCHOVER, 2011).

Additional projects relate to securing and maintaining the privacy of user-specific information in VANET environments and authentication of messages received by any component of the network (Guo et al., 2007). The European Union funded SeVeCom (Secure Vehicular Communication) (SEVECOM, 2011) focuses on developing a definition and implementation of security considerations for vehicular communications. The SeVeCom goal is to provide improved road safety and optimized traffic through IVC and VRC. For its part, SeVeCom focuses on security aspects of such an environment. Its objective is to define the security architecture of VANETs to combat threats such as identity cheating or denial-of-service (Leinmueller et al., 2006, SEVECOM, 2011).

As detailed in this section, many projects exist under funding from both commercial and government organizations to assist in the development of transportation and infrastructure components to improve traveler safety with varying degrees of interaction with individual users. This research leverages the analyses of these projects in the definition of the TrAITS simulation platform and travel modeling framework to allow verified of similar actions within the simulation space at a fraction of the implementation cost.

### 2.2 Communication Considerations for ITS

Most research into the aspects of IVC concern the use of communication reliability in VANET environments and the efficient creation and maintenance of virtualized ad hoc network infrastructure. This includes aspects of routing (Fan, 2007), protocols (IEEE, 2011; Talk, 2008; ASTM, 2010; DSRC, 2010), and clustering (Chatterjee et al., 2002; Haran et al., 2005; Fan et al., 2005; Fan et al., 2006). To achieve optimum reception reliability, safety services (Schaufer et al., 2006) in these networks general use single-hop communication, often with connectionless broadcast protocols (Yin et al., 2004; Xu et al., 2004) over dedicated wireless bands, an
unfortunate implication of the trouble establishing and maintaining multi-hop connectivity with high mobility in time critical situations. Multi-hop communication may be used under some circumstances to enhance the range of message distribution (Wu et al., 2005b; Korkmaz et al., 2004) and is of significant interest for communicating over large distances where cellular and VRC relays may be too difficult or costly to implement. Along these lines, (Kutzner et al., 2003; Ott and Kutscher, 2004) discuss deployment of Internet gateways along roads networks and the concerns with establishing such an environment. The performance of both IVC and VRC and methods for evaluating it are discussed in (Singh et al., 2002), specifically the performance of IVC 802.11b communication. Note that more recent analysis of IVC and VRC communication focuses on the WAVE-specific 802.11p (Fisher, 2007; IEEE, 2011) or 802.11a (Cottingham et al., 2007) which provided the foundation for its development.

The communication performance under three operating environments such as suburban, urban and highway is evaluated independently. (Gass et al., 2006) investigates the performance of 802.11 at speeds up to approximately 120 km/h with varying packet sizes and wireless traffic. This analysis, however, was based upon an idealized, but unrealistic, environment with a vehicle traveling down a straight road with no traffic past an access point and neglected the obvious problems presented by urban environments where line-of-sight and flooding overhead become problematic. The wireless field analysis detailed in chapter 9 discussed these concerns in more detail.

(Cottingham et al., 2007) evaluates 802.11a at 5.2 GHz in an VRC scenario of a moving vehicle and fixed roadside station under realistic speeds and details low speeds network performance degradation results from wireless null zones, a definite concern for implementation of VANET applications with VRC relays or communication between vehicles with relatively slow relative motion. As noted, the recent 802.11p standard for dedicated short-range communications (DSRC) derives from 802.11a and has been shown to have similar properties (DSRC, 2010; IEEE, 2011).
TCP and UDP performance is also investigated under an environment with roadside access points by (Ott and Kutscher, 2004). As a contrast, (Wellens et al., 2007) find UDP and TCP performance at high speeds (120 km/h) does not degrade significantly.

### 2.3 Motivation from Mobile Ad Hoc Networks

These studies detailed within this chapter explored many of the current and recent research projects for ITS applications as well as many worthwhile applications which have been developed under the governance of various bodies.

The impact communication parameters (e.g., packet size, signal level, and speed) on the reliability and performance of wireless networks is also significant to consider (Fan, 2007), but the main focus of this research is on the development of a platform within which the applications and scenarios discussed by these authors can be thoroughly investigated.

The projects discussed primarily concern the development of real-world components for improving the safety of vehicular travel. The networking components crucial to implementing these concepts such as reducing communication overhead through the establishment of cluster-based network hierarchies is the focus of this and other research. These methods of clustering, the problematic test environments available, and the implementation concerns noted for each project motivate the design of the TrAITS environment as a platform for rapid verification and refinement of ITS algorithms.

Chapters 3 and 4 introduce the concept of vehicular ad hoc network clustering and detail the early studies which ultimately resulted in the development of the TrAITS environment and simulation insights noted throughout the rest of the document.
3 Clustering Algorithms

3.1 Mobile Ad Hoc Networks

Mobile Ad Hoc Networks (MANETs) facilitate the deployment and configuration of networks without dedicated infrastructure and are ideal for scenarios in which rapid network establishment is necessary and wired or cellular networks are not available such as in battlefield or emergency operations or vehicular communication (Kahn et al., 1999; Wan et al., 2002, Haran et al., 2005).

The basic flat topology model for MANET environments (Guptar and Kumar, 2000; Fan, 2007) the nodes within the network are distributed across a two-dimensional plane and form ad hoc connections with neighboring nodes. The aggregation and interconnectivity of these connections forms the basis of a well-connected network as a result of high link redundancy but suffers from efficiency and reliability problems due to the overhead of maintaining and verifying network links.

Other studies have shown flat ad hoc network configurations are not conducive to good performance for tasks including routing and broadcasting (Guptar and Kumar, 2000; Hong et al., 2002) and is prone to scalability problems compounded by mobility characteristics (Fan, 2007).

The advantages MANETs provide with rapid deployment and dynamic reconfiguration must be evaluated with respect to their reliability and scalability for reasonable deployment scenarios. Although MANETs provide a feasible means for efficient communication and information access (Fan, 2007), this is not without significant concern relating to the optimal methods for organizing the network without loss of bandwidth.

As outlined in (Fan, 2007) for background into development of the Directional Stability-based Clustering Model (DISCA) model, many approaches have been taken to reduce the routing complexity in MANETs.

Flooding (Ni et al., 1999) is a method in which a sender node broadcasts a message to all neighboring nodes which forward the message to their neighbors. This procedure continues until the destination node receives the request and replies with a response which continues this same model of inefficiency leading to significant messaging overhead.
Destination Sequenced Distance Vector (DSDV) (Johansson et al., 1999; Perkins and Bhagwat, 1994) is a distance vector routing protocol which utilizes local tables within each wireless node such that communication can be directed to the appropriate neighbor to route throughout the network. Each table contains the appropriate neighbor to which to forward the message and the number of hops for each destination. The routing tables are maintained through periodic route request broadcasts which add some network overhead.

Ad Hoc On Demand Distance Vector (AODV) (Perkins, 2001a) is similar to DSDV (Fan, 2007), but adds features of reactivity (Perkins, 2001a) and on-demand route determination to the routing table maintenance and route discovery. AODV finds routes when needed and has an efficiency advantage because it disregards nodes which are not communicating and does not require the periodic route beacons of DSDV. Route discovery in AODV involves sending the request through all intermediate nodes between source and destination and waiting for the reply from the receiver. Intermediate nodes update their routing tables during this process to ensure an efficient path is available for relay.

Like AODV, Dynamic Source Routing protocol (DSR) (Johnson and Maltz, 1996; Johnson et al., 2003), is a reactive, on-demand scheme which allows the network to be completely self-organizing and self-configuring. The route discovery and route maintenance procedures of DSR aid discovery and maintenance of routes to arbitrary destinations and provide the foundation for the protocol’s efficiency. Like AODV, intermediate DSR nodes will automatically update their own routing table during normal activity (Fan, 2007).

These routing protocols carry a communication overhead of $O(n^2)$ (Belding-Royer, 2002) such that increases in the number of network nodes trigger quadratic increases in the network’s overhead. The overhead of maintaining up-to-date routing tables for high-mobility networks and the delay caused by initialization of all node’s routing tables can become unacceptable for larger networks. A hierarchical structure, or clustered model, is efficient and essential for achieving a desirable performance guarantee in large MANETs (Perkins, 2001b). Aided by clustering, efficient routing and broadcasting (Stojmenovic et al., 2001) and improvement in resources utilization, especially bandwidth, can be achieved. In this model nodes are partitioned into
clusters with each cluster’s nodes nominating a representative or clusterhead to coordinate and manage the cluster.

Cluster-Based Routing Protocol (CBRP) (Jiang et al., 1999; Santos et al., 2004) divides the wireless network into a number of clusters for routing efficiency and maintenance of topology information. Within each cluster a node assumes the role of clusterhead. These clusterheads communicate with one another as well as the members of their cluster. Packets sent in CBRP are routed at the cluster hierarchy level and not across all nodes. CBRP incurs lower overhead (Sucic and Marsic, 2004) compared to other protocols during topology updates and benefits from quicker convergence (Stojmenovic et al., 2001).

The main benefit of the CBRP approach is that existing routing protocols can be directly applied to the network by substituting nodes with clusters in network abstraction (Krishna et al., 1997). Different clustering algorithms generate different cluster structures for the same network with varying stability characteristics which significantly affect the performance of CBRC or broadcasting. This research designs and analyzes different clustering algorithms for VANETs and details various algorithms for the establishment and maintenance of clusters based on mobility characteristics of vehicles.

The clustering technique partitions a network into groups (clusters) as a single-depth hierarchy in the network. The multi-level hierarchy approach (Shacham and Westcott, 1987; Lauer, 1995) builds upon this model to further abstract the network topology and gain organizational and communication efficiency in larger networks.

3.2 Graph Representation

Clustering partitions the network into logical groups based upon some predefined rule (Fan, 2007). Within a cluster structure nodes are classified as clusterhead or member. A clusterhead serves as a local coordinator for its cluster and performs data forwarding, transmission management, and other network activities for its cluster members. A non-clusterhead member or ordinary node does not maintain any links to neighboring clusters or members of neighboring
clusters. A clusterhead maintains links to other neighboring clusters such that it may access neighboring clusters and forward information between clusters.

A cluster structure guarantees performance and connectivity in a MANET with a large number of mobile nodes as detailed in (Perkins, 2001b; Belding-Royer, 2002) and is an effective topology control scheme (McDonald and Znati, 1999; Lin and Gerla, 1997; Hou and Tsai, 2001). Many benefits of clustering have been noted (Fan, 2007) leading to an overall efficiency gain based on five key factors. First, hierarchical structures facilitate the reuse of resources which increase the system capacity (Lin and Gerla, 1997). Second, clusterheads can coordinate a cluster’s network events to save retransmission and reduce contention and collisions. Third, clusterheads routing reduces the information discovery and dissemination process to a possibly much smaller subset of nodes (Kozat et al., 2001; Pealman and Haas, 1999). Fourth, clustered networks appear smaller and more stable relative to each mobile node (McDonald and Znati, 1999). When a mobile node changes clusters, only its former and current clusters need respond to this update (Iwata et al., 1999; Chen et al., 1999). Finally, clustering readily scales to large numbers of nodes. However, this scalability into very large collections of clusters or superclusters raises obvious needs for data aggregation and distribution and efficient resource utilization. Clustering has also been shown extremely effective in multiple message communication schemes such as one-to-many, many-to-one, one-to-any, or one-to-all (Younis and Fahmy, 2004; Fan, 2007).

As explained in (Fan, 2007) and modeled in the clustering simulation implementation of (Haran et al., 2005), communication in networks is modeled by an undirected graph $G = \{V, E\}$. The vertex set $V$ represents the wireless-equipped nodes. The edge set $E$ contains pairs of nodes which may directly communicate with one another. These edges represent the possible recipients of messages throughout the network. The ‘$\epsilon$’ symbol denotes that a member exists within the set. For example, ‘$a' \epsilon \{ 'a', 'b', 'c'\}’. Edge $(u, v) \epsilon E$ if and only if between nodes $u, v \epsilon V$ transmissions can be received.

Neighbors$(v)$ is the set neighbors of node $v \epsilon V$. The cardinality of Neighbors$(v)$, or number of neighbors, is denoted as degree$(v)$. Every node $v$ in the network maintains a unique identifier
id(v) which is most often associated with the node radio’s MAC. Within this model, all neighbors can receive messages broadcast by the node v.

A dominating set (DS) of G is a subset D of V such that every vertex within V is either in D or adjacent to a vertex of D (Wan et al., 2002). Given this, a dominator of a vertex u is any v in DS such that the distance between u and v, d(u, v), is at most one hop away. A connected dominating set (CDS) is a DS that induces a connected subgraph of G.

Clustering uses each dominator as a clusterhead (Fan, 2007) (as a node located in the overlapping region between clusters). Many existing CDS finding algorithms in MANETs have a common objective of constructing a minimum connected dominating set (MCDS) to facilitate routing rather than to provide a robust hierarchical structure. This problem of finding a MCDS in a general graph is NP-complete (Garey and Johnson, 1979). The performance of existing CDS constructing algorithms (Das et al., 1997; Wu and Li, 1999; Wan, et al., 2002) shows maximum message and time complexities of $O(n^2)$ with the size ratio between the computed CDS and optimum (MCDS) $O(n)$. Clustering in CDS has been performed by using every dominator as a clusterhead with nodes assigned to the clusterhead that dominates them. It has been shown that within the CDS-based clustering algorithms (Wu and Li, 1999; Chen and Liestman, 2002) local network topology updates may require global adjustment of the structure of CDS and cause large communication overhead for the maintenance. Thus, CDS-based clustering is more appropriate for static networks or networks with low mobility (Yu and Chong, 2005).

It may be undesirable to have neighboring clusterheads within each other’s transmission range (Basagni, 1999) so gateway nodes are often used to bridge communication between clusters with clusterheads scattered out of range of one another. This requirement helps alleviate the interference caused by reachable clusterheads and maps to the concept of an independent set problem. Given graph G, an independent set (IS), I, is a subset of V such that no two nodes in I are connected by an edge in G. A maximum independent set (MIS) is therefore an IS containing the maximum number of vertices. Finding a MIS in a general graph is NP-hard (Garey and Johnson, 1979). A MIS is also a dominating set (DS) such that every node in the network is either in DS or adjacent to a node in DS. MIS construction algorithms guarantee a constant
approximation ratio to MCDS and a linear upper bound message and time complexities. Therefore, many clustering algorithms focus on identifying a set of nodes that form a maximal independent set with the dominating property in the network.

The topology of wireless mobile ad hoc networks is often modeled as a unit disk graph (UDG) (Clark et al., 1990), in which there is an edge between two nodes if and only if their distance is at most one. By first constructing an MIS (Fan, 2007) and then identifying a set of clusterheads, clustering provides a sub-network for efficient message routing, forwarding, etc. This set should consist of a small number of nodes that are well distributed between any pair of nodes. The size of any MIS in a UDG has been shown to be at most five times the size of the minimum dominating set (MDS) as each node is adjacent to at most five independent nodes (Marathe et al., 1995). (Alzoubi et al., 2002) established a tighter bound on the size of any maximal UDG IS as at most $4\text{opt} + 1$, where opt defines the number of nodes of the optimal solution for the MDS problem in the UDG. Therefore, by constructing a MIS, a cluster structure with a high quality dominating set, i.e. a very good approximation of the optimum size, while keeping the independence property is obtained.

### 3.3 Maintaining Cluster Stability

Communication network clustering organizes the network nodes into a hierarchical arrangement. Figure 8 provides an example of the organization of twelve nodes into three clusters. The basic communication capability between the twelve nodes is outlined as connections between the bottom tiers of the hierarchy (in green with more frequent dashes). These nodes are grouped into clusters using the same algorithm. The three clusterhead nodes are displayed with connections between them (in blue) representing the possible message paths under the cluster-constrained network (Krishna et al., 1997; Ramanathan and Steenstrup, 1998; Bettstetter and Konig, 2002; Bettstetter, 2004).
This clustered architecture reduces the communication relay points for each node to a small subset of the total network either using gateway nodes, nodes that are in the communication range of two or more clusterheads, or directly communicating with other clusterheads by switching to long-range transmission mode. Clusterheads aggregates local member topology and act as a relay point for communication between its members and members of other clusters. This reduces the messages exchanged between individual network nodes and the overhead of information stored within those nodes (Garg and Shyamasundar, 2004).

The main goal of clustering algorithms is to establish and maintain an efficient cluster structure which provides a connected path and tends to cover the entire network. So a distributed clustering algorithm usually consists of two phases: cluster setup and cluster maintenance.

Attention on clustering in MANETs has increased considerably as wireless technologies improve and MANET theories become practice (Fuessler et al., 2007). Most of these approaches embrace the role of a clusterhead that maintains the cluster and provides the entry point of that cluster into the broader network. In addition, a low cost clustering method should be able to partition a wireless network quickly with little control messages overhead, usually periodical beacons (HELLO messages) broadcast by each node which signals its existence to all neighbors. As a result of receiving the HELLO messages from all the nodes it can hear, each node is able to dynamically build up its latest neighbor list.
Many approaches proposed for clusterhead election in MANETs are based on node ID or degree. Linked Cluster Architecture (LCA) (Baker and Ephremides, 1981) is one of the earliest ID-based clustering algorithms intended for networks of less than 100 nodes. LCA was later revised in (Ephremides et al., 1987) to reduce the number of clusters in the network. In the revised version, a node could declare its leadership role only if it had the lowest ID among the undecided nodes in its neighborhood. This method is more commonly known as Lowest-ID clustering.

In (Parekh, 1994) the connectivity degree, instead of node ID, is used for clusterhead election. This method, known as Highest-Degree clustering, selects the node with the highest connectivity degree (maximum number of neighbors) as clusterhead. The neighbors of a clusterhead become member nodes. The general idea is that choosing nodes with the highest degree as clusterhead candidates tends to uncover larger clusters. In MANETs, however, small movements of nodes can usually lead to a large number of degree changes throughout the network. As a result, the stability of such clusters degrades over time (Amis and Prakash, 2000). So clusterheads of this algorithm are unlikely to maintain clusterhead status for long. Put simply, Highest-Degree has large but unstable clusters because degree is more prone to topology changes than ID.

Besides using either ID or degree to elect clusterheads a generalized clustering algorithm (GCA) is proposed in (Basagni et al., 1997) which provides a generic method of preferences for choosing clusterheads. A detailed weighted clustering algorithm (WCA) is presented in (Chatterjee et al., 2002) by taking multiple parameters such as node degree, speed and power into consideration for optimizing clusterhead election. In addition, a distributed version of GCA, namely distributed clustering algorithm (DCA) is discussed for in (Basagni, 1999) which describes the dominance and independent properties desirable for any ad hoc clustering algorithm. DCA, however, unrealistically assumes that the network is static or quasi-static and as a result does not have a cluster maintenance procedure to deal with mobility-induced topology changes.

In response to the problem mentioned above, an enhanced version of DCA, distributed mobility adaptive clustering (DMAC), is designed in (Basagni, 2001) to relax the no or low node mobility constraint on DCA. This method includes a mobility-adaptive cluster maintenance
procedure. Each node reacts accordingly when it detects a topology change in its neighborhood. In DCA and DMAC the clusterhead election procedure is similar to the Lowest-ID and Highest-Degree methods. A node has an undecided status when joining the network and it decides its role by its own weight and the weights of its neighbors. DMAC utilizes a continuous periodic beacon protocol within each node to exchange and update neighboring knowledge and react to topology changes quickly and correctly. For instance, if a member node detects a connection loss with its clusterhead it will quickly determine its role again by executing a subroutine called ‘link-loss’ in the DMAC maintenance procedure. Similar to DMAC, (Basu et al., 2001) propose a mobility weight based clustering algorithm (MOBIC) which uses a new mobility metric called aggregate local mobility (ALM) to elect a clusterhead other than ID. ALM is computed as the ratio between the received power levels of successive transmissions (periodical beacons) between a pair of nodes, which indicates the relative mobility between neighboring nodes. The motivation of MOBIC is that when nodes are not equipped with their own speed and direction information network condition such as reception ratio may serve as a good indicator for certain mobility features.

The cluster stability, as noted above, is critical to avoid the prohibitive overhead incurred during clusterhead changes. In (Chiang et al., 1997), the Lowest-ID algorithm was further improved by making an additional requirement: if a member node with a lower ID moves into the range of another cluster with a higher ID, no cluster reorganization is required. This approach, known as least clusterhead change (LCC), drastically reduces the number of cluster changes as the results of forming a cluster structure. However, there exist some cases where a sequence of cluster changes in LCC could be triggered by a topology update. For example, when two clusterheads come into their mutual communication range, according to LCC (Lowest-ID and Highest-Degree) the one with a lower ID will resign and become a member node of the other one requiring some nodes in the resigned cluster to search for new clusters. Therefore, one or more of those nodes could form an ID-based cluster. If the newly formed cluster has a lower ID than one of its neighboring clusterheads, a cluster reshuffling will happen again. Such effects may propagate across the network causing a chain reaction of clusterhead changes.
3.4 Lowest-ID Clustering

The Lowest-ID algorithm involves the selection of cluster heads by means of an absolute ordering of a fixed vehicle ID attribute which is typically modeled as the wireless MAC ID. Cluster formation is performed using node-level election of cluster heads. During clustering, each node broadcasts its ID. Each node then chooses as its cluster head the node with the lowest ID. This method has been discussed in great detail in a number of studies (Ephremides et al., 1987; Gerla and Tsai, 1995; Jiang et al., 1999) and is well known for its stability in general MANET applications. In each cluster, the in-range, lowest ID node becomes a cluster head and maintains the cluster membership information of all other nodes.

The simulation platforms of this research randomly assign the MAC ID values to the radio components of each wireless node. This approach approximates real-world situations in which the ID attribute relates to the MAC ID of the network hardware. MAC ID can be assumed to have a random distribution among all wireless devices. The ID attribute of a vehicle is fixed for the lifetime of that vehicle. This property explains why repeated cluster head selection from a local set of vehicles tends to reselect previous cluster heads and why the studies of the next chapter show the Lowest-ID algorithm to be an ideal clustering model for VANET applications.

3.5 Highest-Degree Clustering

The Highest-Degree algorithm uses the degree of the nodes within the network to determine the cluster heads. The general idea is that choosing high-degree nodes as cluster head candidates tends to uncover larger clusters. In MANET implementations, however, small movements in network nodes can often lead to a large number of degree changes throughout the network as nodes move in and out of range with one another. This, understandably, has a detrimental effect on the stability of the clusters over time (Gerla and Tsai, 1995) and therefore it has been shown that cluster heads in Highest-Degree implementations are not likely to maintain cluster head status for long.
3.6 Overhead and Efficiency

Apart from the instability of the structures established by algorithms mentioned, clustering may incur substantial overhead for cluster reorganization if a mobile network consisting of individual nodes lacks a common organizational structure. Therefore, being able to cluster the nodes sharing a certain degree of group mobility pattern is more appropriate than static clustering and can eliminate many unnecessary cluster changes.

(McDonald and Znati, 1999) propose a probability based clustering algorithm, $(\alpha, t)$-clustering, based on node mobility patterns (Fan, 2007). This algorithm predicts the mobility of each node in a cluster and quantifies the stability of a cluster based on path availability between all cluster members over a certain period of time and includes a procedure to dynamically organize the cluster topology. Furthermore, the probability of path availability in this approach is determined by an unrealistic mobility model, the Random Walk Based Mobility Model (Jardosh et al., 2003), which requires all cluster members be aware of each other’s existence and current mobility profile to evaluate path availability. Although not practical in MANET environments, this model may have some merit in a VANET environment when a base assumption of predefined or preselected vehicle routes exists. In this special case, if nodes within a VANET environment were to register their paths of travel it is feasible that the probability of path availability could be significantly increased.

Unfortunately, many of the clustering algorithms suggested by other research do not satisfy the needs of a VANET environment. (Gao et al., 2001) suggests a constant approximation of the optimum but has analyzed extensively for static cases without establishing merit in mobile networks. Many algorithms have been developed for energy efficiency to accommodate limited battery life for most wireless devices. For example, (Bandyopadhyay and Coyle, 2003; Younis and Fahmy, 2004) focus on saving energy through hierarchical clustering in sensor networks and (Ryu et al., 2001) aims to minimize transmission power with a distributed clustering algorithm.

Recent work has simulated the performances of these algorithms with random placement in a square grid with multi-directional node movement. As detailed throughout this document, this mobility model does not relate to any typical traffic movement or translate into VANET
environments. Thus the performance of these algorithms should be reevaluated under realistic traffic environment such as those resulting from transportation microsimulation.

In vehicular networks, as a result of high node mobility in the presence of cluster-based services network performance metrics such as throughput, delay and effective management are highly correlated with the frequency of cluster reorganization. Therefore, stability of cluster formation is an important metric for determining the merit of any clustering algorithm and is covered in more detail in chapter 4.

3.7 Desirable Features for VANET Algorithms

VANETs have significant advantages over MANETs. Vehicles can easily provide the power required for wireless communication devices and will not be seriously affected by the addition of extra weight for antennas and additional hardware. Furthermore, it can be generally expected that vehicles will have an accurate knowledge of their own geographical position, e.g., by means of GPS. Thus, many of the issues making deployment and long-term use of ad hoc networks problematic in other contexts are not relevant in VANETs.

To support the highly dynamic nature of VANETs, clustering must be updated or reorganized quickly to reflect topological changes and node movements. Thus, ensuring stability is a very important challenge for clustering algorithms in VANETs. In summary, a good clustering algorithm should focus not only on forming a minimal number of clusters, as most of the existing algorithms, but also on dynamically maintaining the cluster structure without incurring a high communication overhead.

Association with and dissociation from clusters as a result of the mobile nature of VANET nodes (vehicles) perturb the network and cluster selections. Cluster reconfiguration and cluster head changes are unavoidable. Therefore, a good VANET clustering algorithm should not only focus on forming minimal number of clusters by most of the existing algorithms, but also dynamically maintain the cluster structure without incurring a high communication overhead. In other words, algorithms should also maintain cluster stability as much as possible during vehicle velocity and acceleration changes and/or traffic topology shifts. Otherwise, the overhead of
cluster re-computation and the involved information exchange will result in high computational cost and negate the benefits of VANET communication. The ideal VANET cluster will maintain its clusterhead and members over the long time range.

### 3.8 Summary

This chapter highlights the significant amount of research into clustering algorithms in MANETs. VANETs, however, pose new challenges in cluster head selection and network stability. VANETs must follow a tighter set of constraints than MANETs and can benefit from specialized clustering algorithms such as those which will be discussed in the next chapter.

Nodes, as vehicles in a transportation network, cannot randomly move within the physical space and must instead follow constraints set in place by the real road network topology, driver-based car-following and lane-changing models, and the signs and signals present in every transportation environment. Vehicle movements follow well-understood traffic movement patterns which are not reflected in MANET simulators. Each vehicle is constrained by the movements of surrounding vehicles. Vehicles generally travel in a single direction and are constrained to travel with two-dimensional movement. Given these restrictions and the knowledge of position, velocity, and acceleration commonly available to on-board vehicle systems it is possible to approach clustering more intelligently and possibly discover a better clustering methodology for VANET environments.
4 Utility-Based Clustering in VANETs

A survey of current MANET research highlights the need for a transportation-specific review of clustering methodology and the discovery of traffic-optimized clustering algorithms. Preliminary research chose to design a utility-based methodology for clustering. In this approach each vehicle implements some form of utility analysis of each proximally located possible clusterhead then chooses a clusterhead by evaluating the utility of each potential clusterhead. The node with the highest utility value is selected as the clusterhead. The objective of this approach is to obtain optimal results by adding transportation specific information to the clustering logic.

4.1 Introduction of Traffic Simulation Model

The constrained environmental conditions of VANETs warrant an equivalently constrained simulation environment. Many simulation tools and environments have been designed for MANET implementations. These tools, however, fail to adequately model the needs of VANETs. Compared to the random movements modeled in MANET environments, VANET simulations must behave according to traffic patterns in terms of car-following, lane-changing, directional movement, velocity, and acceleration among others. Therefore, transportation traffic simulators are considered to be one of the solutions for VANET simulations.

Many simulators exist for testing MANETs-related algorithms, but unfortunately most only simulate random node movements in a fixed area (Basagni, 2001; Chatterjee et al., 2002). Another drawback is that vehicles do not interact with one another and there is no traffic control, which implies the absence of a traffic mobility model, such as car-following, leading to very unrealistic vehicular movements. At least fifty microsimulation models (Smartest, 2009; CORSIM, 2006) are in existence as commercial or academic tools. Many of the existing models, however, have been developed for use on specific research tasks. Even though a number of general purpose micro simulation tools, such as CORSIM, TransSim, VISSIM and Paramics, are suitable for use, most are commercial and do not have a friendly application programming interface. Therefore, as a preliminary approach, an open-source simulator (Treiber et al., 2000) consisting
of simplified key components of a micro simulation model was modified to provide basic wireless simulation capability for the purpose of modeling clustering algorithms. This simulator, however, proved very simplistic lacking real world traffic network and communication model. Nevertheless, it provided valuable insight into VANET clustering algorithms for the next research phase.

This study originally modified Traffic Simulation 3.0 (Treiber, 2005), an Intelligent Driver Model (IDM) (Treiber et al., 2000) micro simulation tool built to monitor traffic flow under various basic highway configurations. This environment simulated accelerations and braking decelerations of drivers (i.e., longitudinal dynamics), and uses the Minimized Overall Braking Induced by Lane changes (MOBIL) lane change model. All model parameters and the initial simulation source code are available for modification (Treiber, 2005) as are the adjustments made for verification of the algorithms discussed in this research.

Well known within MANET studies is that different simulation scenarios and traffic parameters may influence the simulation result dramatically (Sivavakeesar and Pavlou, 2002). Therefore, the key functions for validating the designed algorithms were implemented under six different traffic configurations in the Traffic Simulator 3.0 microsimulation model thus making it capable of demonstrating the clustering algorithms under different scenarios and parameters. The six traffic configurations available in this model are:

- On-Ramp: On-ramp acts as a stationary bottleneck to a traffic on the main road
- Ring Road: Two lane vehicular traffic in a closed system
- Lane Closing and Speed Limit: An open system with a lane closing and speed limit
- Uphill Grade: Shows the uphill gradients effects
- Traffic Lights: Shows the traffic light on the roadside effect
- Lane Change: Shows the effect on a ring road with obstacles on both lanes, and vehicles are forced to perform four lane changes

Although other parameters are available, the main parameters manipulated during the study are:

- Main Inflow: The volume of main traffic flow pouring into the system
• Ramp Inflow: The volume of traffic flow through ramp pouring into the system
• Average Density: The total number of vehicles within the closed ring road system
• Politeness Factor: The waiting patience degree of moving from ramp onto the main road
• Time warp factor: Speed up or slow down a simulation run

The author’s additions to this transportation simulation model not only provided a general framework for implementing the most frequently used functions for clustering algorithms but also leveraged statistical functions that are unique to vehicular networks. As a result, this model enabled the design and implementation of clustering algorithms by making use of basic characteristics of transportation and vehicle movement. The available vehicle and wireless simulations parameters include transmission range, velocity, vehicle ID, degree, position, clusterhead, duration, neighbor, position, neighbor, and velocity. The statistics gathered include acceleration, average velocity within a period time, average acceleration of all neighbors, closest velocity to itself, and average position of all neighbors. The availability of these values for use in the clustering model as utility function parameters enabled the development of new clustering algorithms based upon the combined metrics of wireless and vehicular simulation (Chatterjee et al., 2002; Haran et al., 2005; Fan et al., 2005; Fan et al., 2006).

4.2 Utility-based Approach

A utility-based approach (Haran et al., 2005) to clustering requires the creation of a vehicle specific agent model which is implemented by augmenting each vehicle in a traffic micro simulation platform to periodically determine and store clusterhead information. The clusterhead algorithm was implemented first in a single weight method that calculates a weight value for each vehicle with which the current vehicle can communicate. After implementation of this method, the Lowest-ID and Highest-Degree methods were implemented and tested. Once validation of these algorithms was complete, four other algorithms were designed and implemented to harness vehicle state information. Rather than overcomplicate this initial stage of investigation with
compound weighting logic, the algorithms were chosen to use single parameter weighting based on a) closest velocity to the current vehicle, b) closest position, c) closest velocity to average, and d) closest acceleration to the average of all proximal vehicles with the belief that these traffic specific algorithms would be better predictors of the common traffic situations that lead to cluster dissociation.

The four new clustering methods are summarized as follows:

1. Closest Velocity: A vehicle attempts to join with other vehicles in a clusterhead to member relationship in order of closest velocity to itself.
2. Closest Position to Average: A vehicle attempts to choose as its clusterhead in order of the absolute difference of candidate’s position to the average position of all proximal vehicles.
3. Closest Velocity to Average: A vehicle attempts to choose as its clusterhead in order of the absolute difference of candidate’s velocity to the average velocity of all proximal vehicles.
4. Closest Acceleration to Average: A vehicle attempts to choose as its clusterhead in order of the absolute difference of candidate’s acceleration to the average acceleration of all proximal vehicles.

These steps outline the procedure for implementation of this type of utility function:

1. Each vehicle determines the vehicles within range by polling the neighboring region and tracking the candidate clusterhead set \( C \). All vehicles within transmission range are considered candidate clusterheads.
2. Based on the candidate set and the state information of neighboring nodes, each candidate is evaluated using the utility function.
3. The clusterhead is chosen in decreasing order of utility. In other words, the one with highest utility value will be selected as the clusterhead, and then its neighboring node could join this chosen vehicle’s newly formed cluster. The petition for cluster membership is notified to the chosen vehicle, namely the clusterhead. If the chosen vehicle denies the request, the vehicle with the next highest utility is selected and this step repeated.
A vehicle may deny the selection as clusterhead if it has reached its maintainable limit of cluster members or if the vehicle has already chosen to join with another clusterhead. In all of the algorithms except a) a vehicle may elect itself as its clusterhead. Random selection of vehicles simulates asynchronous cluster selection at fixed time intervals.

The simulation tool was modified to perform all of the aforementioned algorithms: Lowest-ID, Highest-Degree, Closest Velocity, Closest Position to Average, Closest Velocity to Average, and Closest Acceleration to Average.

Note that in this study the implementation of a utility scheme followed a general design for agent behavior but did not reflect true characteristics of agent-based software. Rather, the utility model simulated the interaction of a broadcast clustering method in a typical wireless routing scheme. As research transitioned to the SWANS framework the implementation of these routing table maintenance algorithms was more appropriately placed within the network routing components of the wireless stack implementation.

### 4.3 Simulation Metrics

To measure the system performance, two metrics were identified: (i) the average clusterhead change per step and (ii) the average cluster size. Metric (ii) alone did not accurately depict system performance, so the relative measurement (ii)/(i) was introduced to provide a reasonable comparison metric between the evaluated algorithms. A method is considered relatively better if it has either better stability reflected as a smaller metric (i) or larger average cluster size (ii).

Primary focus was placed on these two metrics during implementation for a number of reasons. First, due to the dynamic, yet constrained nature of vehicular networks, the member vehicles as well the clusterheads tend to move in semi-related motion on the road network. This motion destabilizes the network clusters warranting periodic cluster reformation. Re-clustering may result in the transition of nodes from one cluster to another, split of a cluster into more than one cluster, or convergence of multiple clusters into a single larger cluster. The frequency of cluster formation and cluster change is thus an important consideration in algorithm evaluation. Next and equally important is the size of each cluster. Resource and relay algorithm performance
considerations may limit the manageable size of a clusterhead’s cluster. For simplicity, this research used a common fixed upper bound on all vehicles’ cluster sizes. The implication is that vehicles may reject nodes within range due to resource exhaustion. The delicate balance between cluster size and coverage has major implications in network communication latency and throughput. Each vehicle communicates with vehicles in other clusters through the selected clusterheads. Care must be taken to ensure that the head selection algorithm does not have the unfortunate result of adding network transmission bottlenecks. Alternately, algorithms that yield too many clusterheads may result in a computationally expensive system.

An important area of study is the selection of cluster algorithms that balance high throughput and low latency. The performance of the new algorithms must be measured relative to previously analyzed MANET algorithms. Therefore, the objective of this preliminary research was to evaluate the number of cluster changes and the cluster size for simple combinations of the Highest-Degree and Lowest-ID algorithms with some traffic logics to determine whether traffic specific augmentation can improve stability in VANET environments.

To assist in visualizing the algorithm processing in traffic networks, the graphical display of the simulation tool was modified to display each vehicle cluster with a color representing the current cluster members while following basic mapping algorithms such that neighboring clusters never shared the same color. In addition to display changes, periodic state logging (bookkeeping vehicular related information, e.g., position, degree etc.) was implemented as the basis for simulation results analysis. These features helped provide comparison points for the modeled algorithms.

As discussed above, the utility functions studied are single-dimension weighting methods considering only one attribute of each candidate vehicle. MANET research covers many compound or multi-dimensional clustering algorithms. In general, these methods are presented to overcome certain disadvantages of general MANET models such as power consumption, low mobility, or random multi-directional movement. These algorithms have not been modeled because their contributions to VANET implementations are not immediately apparent. Instead,
general purpose compound algorithms using traffic specific information have been implemented to obtain better overall clustering performance.

**4.4 Preliminary Experiments and Discussion**

The simulation results represent the performance of each algorithm across various wireless transmission range values (0-300 meters) and maximum vehicle speed (40-140 kilometers/hour) while holding the simulation time duration constant across all tests. To minimize traffic flow variability between simulations and enable repeatable test results, the randomized features of the model were seeded with the same value at each simulation run.

![Cluster changes vs. transmission range.](image)

Figure 9 summarizes the variation of the average number of clusters with respect to the transmission range. It illustrates the performance of all six algorithms for a reasonably standard traffic flow environment with a fixed maximum speed of 100km/h. Notably, the Lowest-ID and Closest Position to Average algorithms show rapid initial increase of clusterhead changes as a result of transmission range increase. These algorithms quickly converge, however, in line with the uniform distribution of the randomly generated ID values and vehicles in the driver model. For
small transmission ranges, most vehicles remain out of each other’s transmission range. This leads to a severely disconnected network. For the other four algorithms, the likelihood of change in either of the metrics as a result of increased transmission range results in a steady increase in the number of clusters with transmission range. The Lowest-ID algorithm clearly performs better than the other five algorithms and shows a convergence to a stable cluster count. The Highest-Degree, Closest Velocity to Average, and Closest Acceleration to Average algorithms show almost equivalent performance characteristics. Finally, the Closest Position to Average and Closest Velocity algorithms show similar performance; a result of common traffic patterns wherein similarly located vehicles are more likely to share similar velocities.

![Figure 10. Average clusterhead changes vs. speed limit.](image)

Figure 10 shows the effect of varying the maximum speed on the average number of clusterhead changes with a fixed transmission range of 150m. Algorithm performance is consistent with those of Figure 9. Speed limits are only useful in heavy traffic situations.
In Figures 11 and 12, the Lowest ID algorithm is omitted because it significantly outperforms the other algorithms and makes the relative performance among the other five algorithms...
indistinguishable. Figure 11 displays the performance of all but the Lowest ID algorithm over various transmission ranges. Higher curves indicate better overall performance.

Highest-Degree, Closest Velocity to Average, and Closest Acceleration to Average again show similar performance and better overall results than the Closest Position to Average or Closest Velocity. Figure 12 shows the overall performance across various speed limits for this same algorithm subset. Note that the Closest Velocity to Average algorithm outperforms the Highest-Degree and Closest Acceleration to Average as the maximum speed nears 100km/h. At this speed, the overall traffic flow performs optimally without any noise (traffic slowdown or bottleneck).

4.5 Compound Utility-based Clustering

Based on the initial investigation of a single-dimension utility methods, (Fan et al., 2005; Haran et al., 2005) proposed an advanced compound weighting method utilizing multiple attributes and taking advantage of aggregate utility for these characteristics. In this approach, the Lowest-ID and Highest-Degree attributes were augmented with traffic specific metrics with the goal of improving the basic utility-based approaches by weighing different domain specific attributes to aid clusterhead selection.

![Figure 13. Clusterhead selection.](image)

Previous research illustrated that traffic related information such as vehicular position may help choose a clusterhead with more desirable stability. In Figure 13, for example, vehicle N has two candidate clusterheads, A and B within its transmission range. This weighted method chooses clusterheads based on degree and proximity. Clusterhead A has a degree of 10, i.e. 10
vehicles are within its transmission range, and is 30 meters from vehicle N. Clusterhead B, however, has a degree of 9 and is 10 meters away from vehicle N. In this case, N’s preference would be clusterhead B over A due to the distance measurement. Because of B’s close proximity to N, N is expected to be within B’s communication coverage area at a great possibility even with significant velocity changes. This results in an improvement of stability. On the contrary, if N chooses A as its head, the stability measurement would decline.

The compound utility-based algorithm calculates a utility weight for each potential clusterhead using normalized results from each of the component algorithms. Using a weighted sum of component algorithm ranks, the overall algorithm rank is determined. The weights applied to each algorithm sum to one. After each vehicle’s compound utility is determined, the vehicles are selected as clusterheads in order of highest utility.

4.6 Simulation Study of Utility-based Algorithms

As an initial test, the utility functions Closest Position to Average and Closest Velocity to Average were combined with better performance results than their standalone versions to exhibit the superiority of weighted clustering. The results discussed in section 4.4 detailed clear predictable performance for single-value utility-based algorithms. Therefore, the non-weighted versions were used to benchmark the performance of their weighted counterparts.

The final weighing combination (Degree/ID 85%, Pos/AvgVelocity 15%) detailed in the resulting charts outperformed all other combinations during initial tests.

<table>
<thead>
<tr>
<th>Highest Degree / Lowest ID</th>
<th>Position / Average Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>15%</td>
<td>85%</td>
</tr>
<tr>
<td>30%</td>
<td>70%</td>
</tr>
<tr>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>70%</td>
<td>30%</td>
</tr>
<tr>
<td>85%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Table 1. Experimental settings for weighted clustering.
The performance of six algorithms for a reasonably standard traffic flow environment with a fixed maximum speed of 100km/h was evaluated. Notably, Figure 14 shows the algorithms based on the Lowest-ID clustering exhibited rapid initial increase of clusterhead changes as a result of transmission range increase. These algorithms quickly converged, however, in line with the uniform distribution of the randomly generated ID values.

For small transmission ranges, most vehicles remained out of each range with each other which led to a severely disconnected network. For the other algorithms, the likelihood of change in either of the metrics as a result of increased transmission range resulted in a steady increase in the number of clusters with transmission range.

The three Lowest-ID algorithms perform better than the three Highest-Degree algorithms. The simple Highest-Degree method proved to have the poorest performance. The addition of traffic specific information to the Highest-Degree clustering enabled more stable clusters illustrating the potential for the utility-based clustering algorithm aided by a vehicular mobility information (simulated using a traffic microsimulation mobility model) to enhance the clustering stability in
VANET situations. The three Lowest-ID algorithms show similar performance with no significant improvement with the utility-algorithms.

The traffic specific logic did not significantly outperform the already well-performing Lowest-ID algorithm based on the average clusterhead change metric, but did help reduce the initial peak values seen with the simple method at a 50m range. Traffic patterns show that similarly located vehicles are more likely to share similar velocities and in turn remain relatively located for large travel distances. As a result, the algorithms weighed with Position and Closest Velocity to Average clustering components have strong correlation, but different overall performance.

Figure 15. Clusterhead changes vs. speed for utility-based algorithms.

Figure 15 shows the effect of varying the maximum speed on the average number of clusterhead changes with a fixed range of 150m. Algorithm performance is consistent with Figure 14 illustrating that speed limits only tend to affect performance in heavy traffic scenarios, a property studied in detail in (Treiber, 2005).

Figures 16 and 17 detail the performance of all algorithms using the previously defined clustering ratio. Higher clustering ratio values indicate better algorithm performance measured by cluster stability. The chart illustrates that the Lowest-ID based algorithms dominated for ranges.
greater than 50m with noticeable Lowest ID improvement in the utility-based form and both algorithms benefits from the utility-based clustering enhancements.

Figure 16. Clustering ratio vs. range for utility-based algorithms.

Figure 17. Clustering ratio vs. speed for utility-based algorithms.
4.7 Summary

The analysis performed in this chapter details the performance of some well-known clustering algorithms under the constrained MANET environment provided by VANETs. In addition, algorithms incorporating traffic specific heuristics were proposed. These compound algorithms have the flexibility of assigning different weights and consider a combination of the degree, ID, position, velocity of vehicles. The compound algorithms perform better than their original versions under the rudimentary simulation environment presented in this chapter. Additionally, this research highlighted the performance of the Lowest-ID clustering algorithms as optimal. As in MANET studies, Lowest-ID based clustering provides a stable cluster topology over long time durations due to its nature of being an unbiased, uniformly distributed clustering methodology.

The study of the augmented Highest-Degree methods did, however, show some noticeable improvement when the traffic specific heuristic was added. This result shows the potential for improvement of well-known and trusted MANET clustering algorithms when used in VANET environments and the advantage of applying domain knowledge.

When measured using the clustering ratio defined in this chapter a clear advantage to weighting using vehicular characteristics of velocity or position is shown. This contribution shows that the quality of MANET clustering algorithms within a VANET environment may be improved through use of weighted clustering scenarios.

Although the micro simulation in this chapter lacks many aspects of real world environments (key features of traffic scenarios are abstracted), these results aid the design of more efficient methods by establishing the basic intuition of how each attribute affects clustering stability under various circumstances. The insight gained into gathering meaningful visual and quantitative representations of the traffic and wireless simulations during active simulation and as historical evidence provided the foundation for a survey of available models for the simulation of wireless ad hoc networks leading to the eventual selection of the SWANS simulator. The TrAITS simulator detailed throughout this document has been designed to better reflect these scenarios and provide a platform for further analysis of these algorithms.
5 Simulation and Platform Evaluation

The efficiency and performance of any proposed system can be evaluated in one of two ways, either through field operational testing or computer simulation. While field testing can be judged as the best approach to evaluate the performance of the system in the real world, it can be very expensive and time consuming and in intelligent transportation scenarios may require the involvement of hundreds or thousands of people to gather meaningful data. Alternatively, the analyst can utilize a simulation based evaluation to monitor the performance of the system.

Traffic simulation models are computer programs that use mathematical models to conduct experiments with traffic events on a transportation facility or system over extended periods of time (HCM, 2000). Simulation models can be used to replicate both discrete and continuous systems. While discrete systems present the state of the system at discrete points in time, continuous simulation systems replicate the state of the system continuously over time. Continuous simulation systems are typically considered more suitable to replicate characteristics of the traffic flow in the system, but it is well known that sufficiently frequent discrete modeling of the system can adequately approximate the continuous form.

5.1 Traffic Simulation Model

Traffic flow simulation models can be roughly classified into the categories of macroscopic, mesoscopic, and microscopic, depending on the representation of the traffic flow within the model.

Macroscopic simulation models consider system wide representation of the traffic flow and characteristics rather than individual vehicles. These models can be either deterministic or stochastic. Fundamental traffic speed flow relationships are utilized in these models to describe aggregate traffic condition at any given time. Several traffic simulation packages like TRANSYT, FREQ, and META are examples of such models (Smartest, 2009). These models have limited applications in responding to changes in traffic condition and are therefore not well suited for the study of VANET networks. In the macroscopic model, system-wide parameters such as traffic
density and flow are configured and used to distribute the traffic within the road network. This approach neglects individual behavior and has no foundation for analyzing the transportation effects of changing the behavior of an individual vehicle or person as is common in VANET applications. Macroscopic modeling typically uses a broad set of rules for defining the flow on network segments based on calibration data collected for zone boundaries. Finally, modeling of pedestrian behavior is typically not available. Methods for modeling the public transit are often provided.

Mesoscopic simulation models analyze platoons of vehicles and aggregate their characteristics as a single unit to study traffic flow. These models are capable of analyzing minor changes in the traffic condition over a short period of time but do not offer transparency into individual vehicle or driver behavior. Examples of these models include DYNASMART, VISTA, METROPOLIS, DynaMIT, SIMNET, and SATURN (Smartest, 2009). In these models aggregate statistics such as speed and travel time are used model the traffic flow throughout the network and the effect of network zone on one another.

Microscopic simulation models simulate the individual interaction of the vehicles and thus are more suitable for simulation studies of traffic systems. Generally based on traffic behavior theory, these models simulate the time-space trajectory of individual vehicles in the network and vehicle to vehicle interactions. Furthermore, microscopic models tend to be more flexible in replicating the real traffic condition due to their ability to account for vehicle and driver characteristics. They are capable of associating different classes of vehicles with different attributes like speed, acceleration and driving characteristics. Moreover, the driver behavior, as in stress free or more aggressive behavior, can be accounted for in such simulation platform. The foundation for these models is typically based upon psycho-physical research into user behavior under various traffic and driving situations. Microscopic models allow the researcher observe of modify the state of any vehicle within the network in simulation time such that impact analysis of such a change may be performed. Microscopic models often provide a wide variety of vehicle and transit types as well as options for users to define new types based on preference. More advanced microscopic models extend their simulation to multiple modes of transportation including the movement of
people and allow the segregation of transportation modes such that rail, automotive, pedestrian, and bicycle traffic, for example, may have both shared and separated routes of movement. Within microscopic models, key aspects of transportation affecting the efficiency of the overall network, such as the number of people in a vehicle, can be modeled.

The popularity of these models is associated with the availability of application programming interfaces (APIs) that allows users to include custom-tailored functionality. These APIs are often coupled with powerful visualization engines. Users can retrieve data from the simulation for analysis and provide interactions between the simulation processes and control parameters such as signal controllers, driver behavior and vehicle characteristics. Additionally, these models are capable of accounting for driver behavior at the micro scale. They can also model all elements of the transportation network including interactions between system elements and junctions such as various aspects of signal timing and sensing.

It has been shown that microscopic simulation models can generate more accurate results (e.g., in terms of travel time, speed, delay, or queue length) compared to mesoscopic or macroscopic simulation models (TEC, 2005). However, the challenge is in providing good vehicle and driver characteristics and calibrating the simulation process. Evidence suggests that microscopic models face calibration difficulties due to their underlying car-following and lane-changing behavior assumptions that may not fully represent real life driving condition. Therefore, model calibration is an important and challenging task in these models.

Preliminary studies utilized a simple micro simulation model for the purpose of investigation of clustering in VANETs. This simulator, however, lacked a wireless communication model and realistic environments for traffic flow and road network. Building upon this research, a more robust simulator platform with a realistic network simulator and more robust transportation model was assembled for follow-up studies and is detailed as a primary contribution of this work.

5.2 Wireless Simulation Model

A challenge in the simulation of wireless network characteristics of VANETs is that the majority of existing MANET simulation models available model random movements in fixed field areas and
do not follow movement trajectories based upon vehicle travel routes. This is clearly not suitable for VANET models because VANET nodes move within a predefined transportation network with movement patterns following well-known rules from transportation science as reflected in micro-simulation models. This major limitation comes from the fact that models which deal with wireless communication techniques do not adequately model V2V interactions in traffic flow. The absence of traffic mobility models results in unrealistic movement of vehicles in the network.

Additionally, traffic control options (e.g., traffic signals, speed limits, etc) are lacking such that nodes moving within the network do not model even the most basic behavior of stopping for signals, signs, or other vehicles. Consequently, many of these models provide results based movement patterns which are unfortunately unrelated to real-world traffic scenarios. The result is that the wireless communication between vehicles and the outcomes of these models can be impractical and unreliable. Therefore, the robustness of these models must be improved with respect to transportation network features. Note that some simulators (e.g., the NS-2 network simulator and STRAW) can partially address this concern by incorporating real network map data into the model (Saha and Johnson, 2004) either through trace file uploads or following network based on road survey Topographically Integrated Geographic Encoding and Referencing system or TIGER files (TIGER 2011) but still lack the robust modeling features of a transportation network simulator. The absence of multi-lane modeling in STRAW, for example, leaves much to be desired by the intelligent transportation scientist.

Alternatively, one can utilize a traffic microsimulation model and incorporate wireless communication capability into it. This can provide the flexibility to model data communication among vehicles in the network and between vehicles and traffic control systems. For instance, some researchers have accounted for traffic constrained motion using real road scenarios (Wu et al., 2005a) in which CORSIM, a widely used micro simulation model, provides a highly accurate movement model that has been validated against observed traffic patterns. Besides the well-known limitations of the CORSIM model (e.g., lack of friendly APIs, not being open source, etc) which can inhibit research and development (Bloomberg and Dale, 2000), it also behaves as
a separate entity from the wireless network simulator which may limit its application for such research.

5.3 Integrated Simulator for VANETs

Initial research into simulation, as described, had obvious limitations from two aspects: 1) lack of network communication model and 2) unrealistic traffic flow data. In order to overcome these limitations, an open source simulator, SWANS++ (Choffnes et al., 2005), was leveraged to integrate both wireless network and traffic control models. Incorporated with freely available US Census Bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) data files (Miller, 2002; USCB, 2006) and traffic flow simulation, the SWANS++/STRAW model is implemented as an extension to JIST/SWANS (Barr et al., 2005), a scalable, high-performance and scalable discrete event wireless network simulator featuring low memory consumption and fast run times and utilizing the Java virtual machine.

SWANS++ incorporates a car-following model with traffic control to introduce vehicular congestion modeling real traffic conditions. The integration of a vehicular traffic mobility model and network communication model makes possible to experiment with applications where the traffic information data is used to dynamically alter the routes taken by nodes. SWANS++ uses TIGER data files as the source of street plans. The TIGER data is provided as packages and organized by county, containing files that provide information about various geographic features, including roads, railroads, landmarks, census statistical boundaries, zip codes for streets, etc.

SWANS++ extracts the names, locations and shapes of roads, their corresponding street addresses and their road classes, which can be used to determine the speed limit and capacity of each road. The design however is not able to model multiple lanes of travel on individual road segments effectively and lacks the sophistication necessary to detail key network features such as dedicated bus lanes, tuning lanes, traffic signals, and on/off ramps.

To investigate more realistic scenarios and gain more reliable performance results, SWANS++ was modified in the following areas.
1) Implementation of the origin-destination (OD) placement model which allows configuration and assignment of vehicles traveling from origin to destination zones with different starting times.

2) Implementation of the flow dispatching model which allowing vehicles to leave their origins based on each predefined traffic rate.

3) Improvements on the OD mobility model by dynamically considering up-to-date traffic conditions into the path generation procedure instead of using static condition. As a result, vehicles traveling from same origins to destination may choose different routes based on the traffic condition when departing.

4) Implementation of the zone traffic placement model which first superimposes the traffic zones onto the corresponding road network layer and then imports realistic zone-to-zone traffic flow data. This organized each map following the traffic zone format and allowed import of realistic traffic data based on the widely accepted zone format.

5.4 Testing Scenarios

Both clustering algorithms and broadcasting protocols were examined on the highway network portion of Cook County, Illinois including a portion of the city of Chicago. According to vehicular mobility models of the simulator (Choffnes et al., 2005) a simulation map (Figure 18) showing the study area and demonstrating major highway/arterial network traffic was carefully designed. The simulation network represented current conditions of the highway system where the potential for many future VANET applications exists. Having various traffic constraints, such as frequent traffic lights and stop signs, low speed limits and number of surrounding objects, or local traffic, however, makes it more complicated to exercise the clustering without the accuracy provided by true transportation simulation. To accommodate these shortcomings of this model, all clustering algorithms in SWANS++ were accommodated and evaluated under highway/arterial network traffic conditions in the study area with modeling of traffic control interaction left as a future task.

The network simulator includes two modules which determine the travel path taken by each vehicle for the duration of the simulation: simple inter-segment mobility (Simple) and mobility with
origin-destination pairs (OD). In the Simple module, the next road to which a vehicle will move is determined stochastically at each intersection. In the OD pair module, the routing decision is based on the shortest path between the vehicle’s specified origin and destination computed for each vehicle before leaving the origin point. In this study, the OD module was implemented to improve the accuracy of the vehicular movement representation in the network. Origin and destination locations are assumed to be the centroids of Traffic Analysis Zones (TAZ) in the region and vehicles are assigned to travel paths between pairs of OD zones.

Figure 18. SWANS++/STRAW modeling Cook County, IL.

Neither of the available models in SWANS++/STRAW provide a clear foundation for modeling the dynamic effects of route change decisions which would be the natural result of well-informed drivers in VANET environments. Nevertheless, the models allowed for more robust simulation of
the clustering algorithms previously analyzed and provided the foundation for research into alternative clustering strategies (Fan et al., 2007).

### 5.5 Realistic Traffic Data

In this study, actual counts of vehicle flows between pairs of OD pairs were extracted from the regional travel demand model and assigned to travel routes between zones in the regional highway network. The data available is based upon zonal centroids which can be positioned outside of the available transportation network grid. In addition, many TAZs can be external to the simulation network. To accommodate the features of this model, centroids of TAZs were positioned adjacent to the closest highway and all trips originating from or destined to TAZs that were located outside of the simulation network were aggregated into external super zones.

While input data in a simulation framework plays a pivotal point, it is important to consider the effects of a random traffic flow on performance of the algorithm. Hence, to enhance the reliability of the simulation, realistic traffic count data was imported to the simulation model. This dataset was extracted from typical weekday OD trip matrices of the Chicago Metropolitan Agency for Planning’s (CMAP) regional travel demand model.

The zone data file obtained from CMAP contains a zone ID, and TAZ centroid longitude and latitude information while the OD trip data file contains a list of OD pairs and trip volume data between those TAZs.

It should be noted that although the algorithm is tested in a simulation model with a real highway network with actual traffic data, the simulation process used in this study has some inherent limitations including the assumption of uniform temporal distribution of travel demand in a typical weekday, the restriction of origin and destination points to centroids of TAZs in the network, and some simplifying assumptions concerning lane changing, gap keeping and other related driving behavior factors in the simulation process. Furthermore, the movement patterns followed a wireless simulation parameter based upon random waypoint mobility rather than that of typical vehicle movement. The paths of travel may be adequately modeled as a
highway road network, but the movement within that network still maintained a random waypoint movement pattern (STRAW, 2011).

### 5.6 Summary

This chapter detailed investigation into utilizing a wireless simulation platform with a customized pseudo-traffic mobility model built upon transportation data sets. This work highlights the challenges of finding an adequate combined transportation and wireless simulator for the purpose of modeling VANET interaction while maintaining the integrity of the underlying transportation movement patterns.

Initial approaches into VANET simulation by this and other authors neglected the wireless simulation model by adapting a basic transportation simulator to manage idealized radio communication between vehicles neglecting the complexity of even the most basic fading and shadowing models.

Later research added transportation parameters to a wireless simulator, SWANS, or more appropriately SWANS++, but failed to adequately model traffic movement patterns and did not offer the level of control necessary for monitoring the control settings of a true transportation network. Its successor, STRAW (STRAW, 2011), showed improvement, but also lacked true transportation modeling as it still relied on the random waypoint assumption rather than vehicle microsimulation movement. Furthermore, statistics gathering was limited to only the few items of interest from prior research. Although generally acceptable as a platform for verification of clustering algorithms, a much more robust system is needed to meet the challenges of ITA simulation. The following chapters detail the challenges, requirements, and design of a robust VANET simulation platform as well as the necessary components for modeling the behavior of users of ITS applications, such as the ITA, in a dynamic network environment.
6 Traveler Modeling

6.1 Foundation

The VANET simulation environment envisioned for this study was based upon previous research into the development and planning of an Intelligent Traveler Assistant (ITA) platform (Dillenburg et al., 2002) which is best understood as the integration of a dynamic multi-modal transportation network, wireless communication, handheld and vehicular computing devices, and software services and applications that provide real-time access to transportation resources and facilitate the efficient coordination of activities. Real implementation of such a device requires significant investment in transportation infrastructure, improvement to current wireless hardware devices, development of collaborative software systems, and integration of thousands of disparate transportation sensors. Recent advances in the technology such as smartphones have made the availability of relatively low cost, yet powerful, handheld and mobile consumer communication devices commonplace providing a robust platform upon which developers can deploy applications. In addition, software and hardware services provided by automobile vendors as standard features and through GPS navigation devices have increased the demand for intelligent transportation services.

Each device, application, or service poses a new series of challenges for the designers as they seek to plan and verify the use and characteristics of such applications before deployment into a real world VANET scenario. Even with the ability to rapidly deploy developed applications to thousands of smartphone users, the researcher is constrained by the need to recruit users, essentially research volunteers, and the ability to target rollout and data collection to key geographic areas of study. Such ready deployment across large user bases may generate a significant flow of data, but whether that data is immediately valuable remains a challenge.

Given these challenges and the prohibitive cost of implementing a real-world large scale test environment for ITS applications, the focus of this research is to provide a simulation platform for
the rapid prototyping and analysis of ITS services using proven methods for simulation from the fields of wireless networking, transportation science and traffic modeling, and computer science.

The desire is to support a traveler equipped with a number of technologies within a device supported with applications and services which increase the efficiency the transportation network and more efficiently use of available network resources by leveraging advance knowledge systems to adjust driving behavior or decisions, and overall reduction in the detrimental outputs of our transportation networks such as lost time, accidental deaths, fuel consumption, and carbon emissions. Built upon new and existing wireless and cellular device technologies, these services will include such features as dynamic ridesharing, spatio-temporal database management, cellular and wireless communications, transportation monitoring and alerting, automatic scheduling and fare payment as well as a general management framework for the monitoring of transportation network health.

An Intelligent Traveler Assistant (ITA) device, described in (Dillenburg et al., 2002), includes features for planning multi-modal travel routes for users and assisting in general transportation activities. Originally proposed as a standalone dedicated hardware device, the ITA will likely instead leverage the advanced cell phone hardware platforms currently available and integrate into platforms already commonplace. In addition, features such as electronic ticketing and payment, common to some smartphone applications today, will help the user navigate the transportation network. The device software capability may be an individual micro-application behaving independently or collaboratively or as a single all encompassing platform. The former is preferred because of its architectural simplicity by providing each application as a both an independent functioning software unit and as a pluggable service interface for other applications and integrates well into modern smartphone application design.

Multi-modal planning in an intelligent transportation system can accomplished through many means but the use of spatio-temporal databases with traveler location awareness and route planning algorithms designed to accommodate user-specific parameters such as privacy concerns, number or type of transportation mode changes, accessibility features, or safety measures. In general, the services will accept a destination from the user and, using its current
location as a starting point, suggest multiple route plans with tradeoffs identified and tuned to the user’s preference or decision history. The devices will make use of wireless and cellular technologies to interact with transit schedules, user profile databases, taxis, vehicles within the transportation network, and other devices to determine the best possible route for the user and arrange and pay all necessary fees leading to a fluid flow throughout the network and reducing the negative aspects of transit for the user.

Another common feature of most smartphone devices, Global Positioning System (GPS), provides location awareness and facilitates the monitoring of progress through the transportation network or the development of location-aware services for the traveler to consume. This information, when coupled with travel times, aids dynamic rerouting of travelers when traffic situations occur or are predicted. Traveler path information and travel time collection may provide further insight into the state of the transportation network and areas for infrastructure modification or improvement. User-centric services based on geographic location will aid the user in navigating local environments or finding key options for dining, travel, or shopping. The data collected and integration with localized marketing will provide many routes of commercialization of the services fueling more advanced development and deployment of similar platforms. GPS is also crucial to determining the traveler’s position and providing periodic updates of to central servers and distributed services for possible use in ride sharing scenarios and localized consumer exchange.

In addition to the options already common to many smartphone systems, an advanced ITS service or ITA device is anticipated to provide at least these features as detailed by (Dillenburg et al., 2002):

- Dynamic rideshare matching
- A multi-modal route planner
- Congestion monitoring and reporting
- Construction alerts
- Incident alerts
- Special event notifications
Transportation schedules, bus routes, ferries, taxi availability, parking options, traffic conditions, etc. are available in forms today and will be leveraged to enhance the traveler’s navigation through the network. ITA services may interact with a centralized infrastructure such as weather or scheduling sites or as an agent connecting to other ITA agents for the advantage of all in the transportation network. Interaction between the ITA and other systems will occur in both an on-demand form as a result of user interaction as well as an automatic form in which the ITA constantly monitors the user’s activity and interacts with other systems to determine better transportation options for its user and assist in queries from other users.

The key feature of the ITA (Dillenbourg et al., 2002) is the planning component which determines a multi-modal route from the traveler’s current position to the desired destination while considering the traveler’s preferences and knowledge of current and predicted future traffic conditions. Prediction of the future state for an individual is important because of the dynamic nature of transportation networks such that anticipated routes and travel times may need to be adjusted based on more recent updates from the transportation network or other ITA users.

In addition, the device must provide for dynamic rerouting of travelers as traffic characteristics change or a user deviates from a defined route. The ITA will continually update the time and cost of the remaining trip by computing the route plan periodically with current information on the transportation network. If a significant improvement is found in a new route prediction, this may be offered to the user as an alternative to the current route. Likewise, this information will integrate with preference characteristics and scheduling services for the user such that travel recommendations may be made based on likelihood estimates of arrival time based on active and predicted delays.

By integrating mass transit schedules and routes of buses, trains, and ferries into the ITA’s planning capability, the device will help guide users to alternative transportation when it may be beneficial to the user. This integration may be accomplished using wireless communication and data mining to consolidate mass transit fares, schedules and routes already freely available on the internet for the benefit of the user.
Such a device will likely collect and distribute current traffic conditions to help itself and other users determine and remain on the optimal route to a destination. Knowledge of traffic conditions such as congestion, accidents, or construction zones as well as special events such as concerts or sporting events will be critical for dynamic route adjustment. The necessary event, planning, and status information is currently available in different forms on the Internet.

Using a location management framework, the device may query a database for the current location of ITA devices, buses, trains, and other equipped vehicles. These queries must accommodate location-specific and time-specific features to provide the resolution necessary to truly assist in planning and routing. Predictive queries may utilize historical information and advanced algorithms to provide estimates on future states in the transportation network. Triggers such as those alerting users to network state changes will integrate into other services to minimize the user interaction required to benefit from the services. The next section of this chapter highlights the conceptual implementation of location monitoring as they relate to the original ITA concept as outlined in (Dillenburg et al., 2002) and devices in use today.

6.2 Location Management

A necessary feature for any location management implementation is the ability to model location information which changes regularly during travel such as is common for many devices today including cell phones.

A straightforward point-location management approach outlines in (Dillenburg, 2002) and supported by other research is to model each moving object as a location-time point \((l, t)\) with periodic updates. These updates provide additional points based on the new location and time. The location may be of many forms, but is typically a \((\text{longitude}, \text{latitude})\) coordinate pair (or triplet when altitude is considered). Any standard GPS device is capable of producing such traces coupled with precise satellite-based timestamps. A trace of this state information over time provides the traveler's path and specific travel times.

Point-location management does not enable extrapolation to predict travel paths and requires additional data manipulation to produce meaningful results for analysis. Another drawback is that
the updates periodicity may be different for each reporter such that a query performed at any specific time is unlikely to have fully up-to-date information.

Another serious problem with the point-location method is a tradeoff between precision and resources. Put simply, to obtain a precise model of the traveler’s location more frequent updates are necessary. These updates, however, consume resources such as bandwidth and processing power and in some cases battery life.

An alternative, trajectory location management, introduced in (Dillenburg et al., 2002), directly relates to the methods employed by many transportation simulation platforms. In this approach, location is stored as a collection of travel trajectories which form the route traveled. Each trajectory constitutes a path within the transportation network.

Trajectory location management when coupled with detailed information on the transportation network provides a mechanism for determining precise vehicle locations within the transportation network. This information may also be used to determine current traffic situations and predict arrival times (Dillenburg et al., 2002) as an approximation of the expected motion of the vehicle.

This method is not without problems either, however, and also suffers from a tradeoff between update frequency (and location precision) and the number of line segments which must be managed for interpretation. Thankfully, the number of line segments on each trajectory may be adjusted and fine tuned using the line simplification method (Barrett et al., 2004; Ben-Akiva et al., 2001) familiar from computer graphics research.

By collecting the trajectory location information from each vehicle, the server can compute the expected location of the moving object at any time for extrapolation of the vehicles current location, while the moving object may maintain its course without unnecessary updates from a the server (Gianotti et al., 2007).

This information may also be coupled with uncertainty thresholds on each line segment (Dillenburg et al., 2002) to define an agreement between the server and traveler on the stated location and travel times. If changing trajectory the traveler notifies the server and an updated travel plan may is provided. This agreement between the vehicle and the server avoids the correlation between bandwidth consumption and precision. The resulting data allows good
precision estimates of the location of a vehicle using few location updates, or no updates at all. If
the moving object is on schedule, i.e. does not deviate from its prescribed trajectory by more than
the uncertainty threshold, no resources are consumed for updates.

6.3 Preference and Activity Modeling

A key component of user behavior is the modeling of user preferences such that decisions
made through the processing of a simulation reflect realistic travel choices made by users of
transportation and, more appropriately, ITS systems.

This research coincides with a series of similar developments to design and implement a
framework for user activity planning and scheduling. This Agent-based Dynamic Activity Planning
and Travel Schedule, or ADAPTS, model (Auld and Mohammadian, 2009a, 2009b) seeks to
dynamically simulate the decision-making process of individual travelers in a vehicular
environment with respect to time-dependent activity scheduling. The goal of ADAPTS is to model
the behavior of an individual user as they make everyday travel adjustments based upon updated
travel information.

6.4 Summary

One goal of this research is to provide a user modeling framework within which researchers
can define and simulate behavioral models representing how a user of ITS-enabled devices may
interact with a transportation network without significant investment into developing the
infrastructure components of a simulation platform. The SWANS network simulator components
of the VANET simulation platform designed in this research accommodate communication among
wireless devices. The movement and transportation network characteristics affecting user
behavior are modeled using the TrAITS environment. Within the TrAITS environment,
researchers can build and run simple applications which can take advantage of all characteristics
of the simulation environment and model the behavior aspects of travelers. A basic design of
such an application is discussed in Chapter 8 as a sample application based in part upon the
research of (Auld and Mohammadian, 2009a) and intended as a foundation for further implementation of ADAPTS within this simulator.

The VISSIM route representation provides a vector-based format which may readily be mapped into the trajectory location management format desired for ITA device modeling by providing travel duration measurements to each sub-path of the route with a certainty measurement. For simplicity, this certainty measure may be represented as an error range on the time duration value as a function of the sub-path duration or distance. Regular travel time and route adjustments available as a component of the VISSIM transportation simulator provide a sufficient proxy to modeling the real-work updates of travel time as a result of loop sensor information and other methods. The data structures defined within the SWANS components of the network simulator provide a basis upon which trace information, similar to the trajectory management structures, and also meeting the requirements of point location management can be maintained. This information can then be used within a TrAITs simulation service to accurately estimate of any vehicle’s location between VISSIM’s time steps or modeled traveler updates. These items form a solid foundation for modeling travelers and ITA devices within TrAITs using familiar techniques from previous research.
7 System Design and Architecture

7.1 JiST / SWANS / SiDnet Architecture

SWANS, the Scalable Wireless Ad Hoc Network Simulator, acts as an application framework built upon the JiST, Java in Simulation Time, discrete event simulator (Barr et al., 2005). Both JiST and SWANS are written in the Java language and provided as open source tools for use in research. JiST is a discrete event simulator which leverages the tools of reflection and bytecode engineering available in the Java language and runs within the Java Virtual Machine (JVM).

In simplest form, the JiST application incorporates simulation method calls into the Java bytecode generated by a standard Java compiler such that native Java applications are capable of running within a simulated environment without source code modifications. This design significantly reduces the test cycle timeline for researches by allowing use of standard language libraries within the simulation environment. The core of the JiST architecture consists of a Rewriter component, which rewrites Java bytecode before loading it into the JVM, and a Kernel Event Loop which manages the creation and processing of discrete events in simulation time. This approach enables verification of simulation event processing in a model allowing concurrent execution of events by applications running upon JiST for independent or coordinated activity.

The application processing flow for the JiST simulator is summarized in Figure 19. Applications running within the JiST environment execute within the standard JVM with modified bytecode which incorporates the JiST simulator as a base for execution processing. This modified bytecode is run within the standard JVM while accessing the core libraries of the JiST framework.
The SWANS application library (Figure 20) sits atop the JiST framework and runs within the simulation time environment provided by JiST. The core SWANS library includes features for modeling and testing aspects of the wireless network stack and protocols which is why the framework proved ideal for initial research into the characteristics of node discovery and clustering in wireless networks. Importantly, SWANS is a separate application and is itself subject to the bytecode reengineering of the JiST model. Independent analysis of the SWANS simulator (Kargl and Schoch, 2007) has shown the validity of the SWANS model in simulating wireless ad hoc networks by proving that the protocols and models implemented in SWANS produce results comparable to the ns-2 simulator with better scalability to simulations with a larger number of network nodes as is common in VANETs. These characteristic of the SWANS simulator and the clearly delineated architectural components for implementation of new network elements and protocols made it an ideal candidate for integration into the VANET simulation environment. The process of installing and adapting a network simulation to SWANS is significantly less involved than that of ns-2, as well, reducing the research time lost configuring and running simulations.
Although open source and freely available for download, the SWANS project has suffered from a lack of major upgrades or maintenance since initial release. Without commercial adoption or a dedicated sponsor, the SWANS project, like many open source projects, has not been actively maintained by the original authors or a community of volunteers. Many small scale patches and updates have been made to these libraries, but in general these have proven incompatible with one another leading to strange simulation behavior. Further, many large-scale add-on project such as STRAW integrate poorly with the base SWANS framework causing difficulty when attempting to model or verify new components. Initial attempts at this research leveraged the SWANS++ and STRAW models for integration with the VISSIM traffic model. Each of these models, however, suffered flawed network stack implementations that hindered development of application components modeling the agent framework. Furthermore, the visualization components within these models proved obsolete when compared to more modern
Java GUI development paradigms causing a serious performance degradation from the core SWANS models.

The SiDnet (SN) project has developed a series of enhancements to the SWANS library to aid in other research (Ghica, 2010) which correct flaws in the wireless stack implementation, add an energy / battery life model for wireless devices, and providing a more robust visualization framework upon which pluggable components may be implemented. SN also implements a series of remote query tools which provides basic geographical region query information to identify the set of wireless nodes in a particular region. These libraries are based upon the simulation world coordinate system which does not directly relate to real-world GPS but provides a simple proxy in some scenarios. In addition, SN adds a series of tools for visualizing the wireless communication among nodes using a standard body and outline coloring approach. Each of these may be of use to researches interested in studying wireless components, but do not significantly aid this research. More information on the SN library is available in the referenced user manual. SN, rather than the base SWANS has been utilized as the foundation for the wireless simulation platform because it provides a series of code fixes to the underlying SWANS platform and significantly improves the TCP stack implementation such that integration of library components may be more readily integrated. The relevant components of the SN implementation are illustrated in the execution models shown in Figure 21. The simulator designed in this research leverages the SN node implementation and base sensor model and integrates with the SN GUI for debugging purposes.
SWANS and SN follow a model in which the applications essentially stack upon one another and in turn upon the JiST simulator. Applications can be written to run within the SN environment using native Java and accessing the modified stack elements of SN and SWANS.

SN is not perfect or complete, however. Analysis into the libraries and recreation of previous SWANS experiments using larger scale simulations showed a loss of scalability when compared to the SWANS implementation. This loss of scalability was due to the enhanced visualization kit available in SN, dependency on code generation units from the NetBeans IDE as the foundation for GUI development, and known limitations of the Java GUI library. To accommodate running
SN on larger scale simulations, the framework allows users to disable GUI updates during simulation processing. To further aid simulations this research has enhanced the SN framework to completely disable the GUI components and allow running of simulations without any visualization.

Even with the enhancements provided by SN, the base wireless simulation platform suffers from a number of restrictions which made integration into the VISSIM environment difficult and error-prone. The most significant issue the SWANS environment, and in turn the SN environment, is an underlying assumption that the number of nodes used in the wireless simulation will be constant throughout the simulation time. This restriction is incompatible with the model of microscopic traffic simulation which allows for the entry and exit of vehicles during the simulation timeline as is reasonably expected in any realistic scenario. Consequently, the initial focus of this development of an event triggering model within SN allowed the dynamic creation and destruction of wireless simulation nodes in an active simulation timeline.

### 7.2 VISSIM Traffic Simulation Model

A shortcoming of many VANET research projects is the failure to model traffic simulation and control in a fashion which follows well-known transportation algorithms and provides results consistent with real-world traffic behavior, including the movement and car following models of individual vehicles or drivers. Many well known algorithms for modeling transportation have been developed since research began more than 60 years ago such that failure to consider these algorithms neglects proven methods for traffic behavior and leads to unrealistic simulation vehicle movement. To accommodate this, a commercial microsimulator, VISSIM (Figure 22), has been chosen as the foundation for the movement model in the hybrid simulator. VISSIM provides a number of control features including facilities for manipulating vehicle parameters during simulation, multiple modes of transportation, and most importantly a modern API allowing dynamic control of vehicles either through a windows based dynamically linked library (DLL) driver model or through a Windows Component Object Model (COM) interface. Both API options provide access to the characteristics of an individual driver, but the COM interface also provides
access to many additional features of the simulation environment including signal timing, OD matrices, routing, and road and network configuration settings.

A traffic simulation model uses mathematical methods to conduct experiments with traffic events on a transportation system over extended periods of time (TRB, 2000) using either a discrete or continuous system. The state of the simulation in a discrete system is adjusted at specific intervals while the state of the system in a continuous simulation varies constantly over time. Continuous simulation systems can characterize the traffic flow system more effectively, whereas discrete models typically allow better stepwise configuration and adjustment. As with any simulation platform, high frequency discrete simulation is a sufficient approximation to continuous activity. VISSIM’s model provides a configurable simulation time setting with a frequency limit of 10 steps per second which is sufficient for modeling driver behavior.

Microscopic models simulate the individual interaction of the vehicles and thus are more suitable for the simulation studies of traffic system. Developed generally based on traffic
behavior theory, these models simulate the time-space trajectory of individual vehicles in the network. These models tend to be more flexible in allowing association of different classes of vehicles with different characteristics like speed, acceleration and driving characteristics. Vehicle state within these models is typically available for data collection on an individual basis in one form or another. Some simulators, including VISSIM, also allow modification of vehicle state outside the simulation through methods such as API interaction or scripting environments. Microscopic models are known to generate more accurate results (e.g., in terms of travel time, speed, delay, etc.) when compared to mesoscopic or macroscopic simulation models (Smartest, 2009).

This research augmented the VISSIM API with a set of methods designed to offer an outside controller application access to the traffic control mechanisms necessary for VANET, and more importantly, ITA simulation. These modifications expose vehicle and control device parameters such as speed, direction, and travel route and allow modification of these parameters in a real simulation timeline. To aid the scalability and performance of the simulation environment the design implements a service oriented architecture layer upon the VISSIM API which allows the traffic and network simulators to run on different machines with a network and offers specialized API options for tasks such as rerouting a vehicle back to its starting location or prompting traffic jams in the live simulation.

The movement model for VISSIM (Figure 23) is based upon the Wiedemann car following model (VISSIM, 2010; Wiedemann, 1991; Fellendorf and Vortisch, 2000), a psycho-physical model for the conscious and subconscious behavior of a driver when interacting in a transportation network. Within the Wiedemann car-following model, driving behavior is classified into four general states:

1. **Free**: The driver is in an unconstrained state of driving and can travel along the road without impact from other vehicles.

2. **Approaching**: The driver is nearing another vehicle or set of vehicles and must begin making adjustments based on car following rules.
3. Following: The driver is following another vehicle either actively or subconsciously as a result of relative positions.

4. Danger: The driver is in danger of hitting another vehicle and must avoid collision by braking.

When following another vehicle the Wiedemann model considers the driver is in a state of oscillation controlled by individual preference and vehicle parameters such that the driver may advance or withdraw from a vehicle in front of the driver. Each vehicle within a VISSIM simulation follows a specific set of car-following rules within such that a driver may perform a unique implementation of the Wiedemann model. The simulation of traffic within VISSIM is the result of hundreds or thousands of vehicles performing implementations of this model based upon simulated traffic which is itself the result of the same car-following behavior. This approach is the basis for the microsimulation model and lends itself to the augmentation of driver behavior based on other factors such as vehicle state or sensors.

Multi-modal transportation simulation involves the simulations of multiple types of transportation and their interaction. Typical multi-modal traffic studies evaluate the effectiveness
of transportation features when considering pedestrian, vehicle, and mass transit activity.

Standard modes of transit available for study in micro simulation tools include individual vehicles, buses, trains, trams, pedestrians, wheelchairs, bicycles, and airplanes. VISSIM also allows a rich API which enables users to specify their own vehicle types and set vehicle parameters based on predefines features such as maximum speed, acceleration, number of passengers, or even automobile color.

### 7.3 General Design

As previously discussed, ITS research is inhibited by the lack of adequate simulation platforms for modeling the complex interaction of both wireless and vehicular network activity while following the theory and principles of both communication and transportation engineering. The platforms highlighted as part of this research displayed the typical inadequacies of all-in-one VANET simulation solutions including unrealistic traffic movement, lack of transportation controls such as signals and loop sensors, lack of pedestrian or traveler capability, and poor modeling of road networks. Furthermore, attempts at modifying existing wireless networks to accommodate traffic simulation (Choffnes et al. 2005; Fan et al., 2006) may represent special purpose situations adequately, but are wholly insufficient for true representation of the intricacies of transportation science.

Previous work on VANET simulation focused on the development of integrated network and transportation simulation environment primarily through modification of an existing network of transportation simulator (Wang et al., 2007; Piorkowski et al., 2007; Sommer et al., 2008; Leung et al., 2006; Gorgorin et al., 2006; Choffnes et al., 2005) with some research attempting hybridization of separate simulators into a common simulation platform (Lochert et al., 2005; Killat et al., 2007; Schroth et al., 2005).

One of the most recent efforts to augment a network simulator to perform transportation tasks is the Street Random Waypoint (STRAW) (Choffnes et al., 2005) project which builds upon the existing SWANS framework to provide a constrained movement environment such that transportation networks can be adequately modeled. Although analysis (Schoch and Kargl,
2007) supports the maturity of the SWANS network simulator, the accuracy of the transportation model implemented by STRAW is left unproven (Gorgorin et al., 2006; Choffnes et al., 2005). The application of random waypoint movement features to a transportation network neglects well-studied methods for modeling the movement of vehicles within a traffic simulation environment. In addition to the implementation of an unproven technique for modeling driver behavior, the STRAW model fails to consider basic transportation network components such as multi-lane roads or traffic signals. Furthermore, the difficulty of configuring transportation-specific parameters within STRAW leaves its value as a tool for the transportation scientist in question. The major advantage of the STRAW/SWANS project is the availability of complete source code for all components of the simulator and the continued development of new models and components to enhance the product.

TraNS (Piorkowski et al., 2007) builds upon the Network Simulator 2 (ns-2) (NS2, 2011) framework to provide transportation modeling in a VANET environment. NCTUNS (Wang et al., 2007), like STRAW, attempts to build both a realistic transportation environment within a wireless communication platform. Although these models provide a more realistic transportation network than STRAW, they fail to provide an adequate platform for the verification of ITA models.

Recent hybrid studies have focused on the integration of the commercial transportation micro simulator VISSIM (VISSIM, 2011) with the network simulator ns-2. These studies benefit from proven network and traffic models and provide a realistic platform for the modeling of VANET applications. In addition, a commercial effort is underway to develop wireless communication feature modeling within the VISSIM environment. This feature as a research platform for wireless analysis on protocols or routing algorithms has yet to be studied in detail but preliminary analysis highlights that the basic focus of the model remains traffic simulation.

This research leverages the knowledge gained during study and simulation of VANET clustering to build a hybrid simulation platform based upon the coupling of the SWANS wireless network simulator and commercially available VISSIM traffic microsimulator into the TrAITS Traveler Agent Intelligent Transportation Simulator.
The basic architecture of TrAITS consists of the VISSIM application, SWANS simulator, and an application interface which maintains simulation time step between the two environments (Figure 24) and provides efficient collaboration between the component simulators. In addition, the framework enables the collection of combined data such that various models may be quantitatively evaluated considering aspects of both wireless and transportation simulation. This model generally follows the approaches of (Lochert et. al., 2005; Killat et al., 2007; Schroth et al., 2005) but expands the goal of modeling to the coordination of multiple modes of transit, including pedestrian movement and behavior, rather than vehicular traffic only. In addition, this model allowed for the VISSIM API to be augmented with additional elements to aid the triggering of changes from the wireless and agent environment.

![Figure 24. TrAITS as a VISSIM-SWANS hybrid simulation platform.](image)

![Figure 25. Basic Object Model for VISSIM API.](image)
VISSIM follows Microsoft Windows architecture and provides three methods for external control, a COM interface, an integrated scripting engine, and a set of external DLLs that allow developers to replace or modify various features of driver behavior. To accommodate the architecture of VISSIM and provide for the most flexibility in interfacing with the simulator, the VISSIM COM API has been wrapped with a web services interface, thus abstracting the interface from its Window's core to a callable open interface for integration by external applications with HTTP connectivity. In addition to wrapping common simulation interfaces, however, common patterns of access for VANET and ITS simulation were implemented, thereby extending the native VISSIM interface beyond its initial purpose or providing access to read or modify the state of individual network elements. Figure 25 highlights the key components of the VISSIM API object model in simplified form. Within a simulation, Links represent the network elements upon which vehicles travel. These are typically roads, but may also reflect railroad lines, rivers, etc. Links are connected to one another by Nodes which typically reflect intersection points but may also connect special areas within a single road such as standard and turning lanes. To avoid confusion, the term Node will not be used to represent the VISSIM entity elsewhere in this document. Parking Lots in VISSIM serve two purposes, first they help represent traffic zone input and output locations, second they may represent true parking locations. In each case they act as entry and exit points for vehicles within the network. Paths and Routes represent the available series of Links through the network. The OD matrices used in dynamic assignment resolve to Paths during dynamic assignment processing and these paths are chosen as Routes by the travelers. During simulation a vehicle periodically reconsiders the chosen route at a series of decision points. An important distinction should be made between routes in an ad hoc network and unrelated vehicular routes. Many forms of signals and detectors are also available within the model.

SWANS provides an open source Java framework with a light resource footprint and good scalability. In previous research, the ns-2 network simulator was integrated with VISSIM (Lochert et. al., 2005; Killat et al., 2007; Schroth et al., 2005) for single mode single purpose ITS
These efforts, however, did not expand into interaction between vehicles, people, and infrastructure components. The more modern architecture of SWANS provides a solid base upon which to integrate with VISSIM using readily available libraries for integration via COM, network, or HTTP connections. Through implementation of a Java web services client application interface, the SWANS simulator is able to integrate with VISSIM's web services.

This simulation environment design couples the VISSIM traffic microsimulator with the SWANS wireless network simulator into a hybrid simulation platform which allows the components of wireless ad hoc networking and transportation planning to be modeled individually or run in a coupled simulation. This form enables the development of VANET / ITS applications which leverage the underlying mobility and wireless communication environments to interpret sensor and state information and action changes with the goals of enhancing the efficiency of the transportation network. The VISSIM environment is a microsimulation platform within which the properties of individual vehicles such as speed or destination may be altered to represent the unique aspect of that participant in the overall transportation network. VISSIM also provides access to the OD matrix for the simulation and dynamically reroutes vehicles based on changing traffic scenarios. Similarly, the SWANS wireless simulator allows individual positioning of the nodes within the location plane of the wireless network. To accommodate the transportation model as the driver of the node positioning, each VISSIM time step triggers a full update of the position information within the SWANS environment. The models available within VISSIM account for action as a result of changing traffic scenarios affecting the local traffic for the vehicle. The model components added as part of this research provide the tools to proactively modify driver behavior before the local traffic has been affected. This core capability provides one foundation for ITS networks based upon the intelligent driver / traveler paradigm of previous research (Haran et al., 2008; Haran and Nelson, 2010).
7.4 Initialization and Synchronization

The basic processing model for the TrAITS hybrid simulator is shown in Figure 26. The JiST/SWANS controller serves as the basis for the initialization and processing of the simulation. Initialization of SWANS simulations requiring the VISSIM mobility model triggers SWANS to
connect into the proxy services and initialize the VISSIM environment. This loads the VISSIM application and preconfigured transportation network and returns the physical characteristics of the environment to SWANS. SWANS uses these characteristics to configure the field element within the simulation. After configuration of the field, the VISSIM simulation is initialized and the set of vehicles available in the network is returned to SWANS. SWANS then creates nodes reflecting the vehicles and configures network elements to support communication between the nodes. During node configuration the agent application model is loaded to support the interaction of nodes with one another through various wireless interfaces. Finally, event processing is started within JiST to process the SWANS simulation and maintain lockstep with the VISSIM environment.

In this model, the JiST controller has been modified to support the synchronized stepping of time across both environments. At each JiST time step the event loop will evaluate the events added during SWANS processing and when necessary trigger a step event in the VISSIM simulation. The assumption under this model is that the time steps within the transportation environment are no more frequent than the duration reflected by JiST simulation time. Upon stepping the VISSIM environment, the current position for each simulation node is updated as is other vehicle state information. This capability has been built to either allow the simulators to move in lockstep with one another or maintain a rate of simulation steps between the simulators based upon configuration options within SWANS and VISSIM. JiST/SWANS may be configured to any processing frequency value using the standard configuration for the simulator. VISSIM provides the ability to configure the time step to a minimum frequency of 10 step per second. To accommodate the potential differences in steps between the environments, the proxy services created to support communication automatically determine a time multiple to keep the timelines synchronized.

The interaction between the two applications is handled using a set of custom interfaces which translate location and vector movement information from VISSIM to SWANS at each time step and in turn allow the simulated ITA wireless application running within SWANS to trigger changes within the VISSIM environment through vehicle property changes. This effectively alternates
control between the vehicle and wireless simulation platforms providing a realistic traffic simulation using the VISSIM transportation movement models and utilizing the wireless communication models within SWANS as an add-on communication layer for each vehicle.

### 7.5 Transportation Network Proxy Architecture

The core components of the integrated environment reside within the VanetSimulation class implemented within the Java environment. This class and the associated member classes provides a proxy representation for the VISSIM objects within the Java simulation environment and helps minimize the calls between the environments by providing a local state of the transportation network elements within the JVM.

VanetSimulation and its member data elements represent the VISSIM network within SWANS. It also relates the changes made to the SWANS proxies to the real network elements by calling the web services API, thus abstracting the complex interaction between the two environments. Access to the web service is handled by a Simple Object Access Protocol (SOAP) proxy which handles service location and connectivity and maps the exposed transportation network services into Java method calls. The SOAP proxy performs the communication with the web service component which interacts with VISSIM. The configuration of the VanetSimulation and SOAP proxy is such that the VanetSimulation and associated members process within the JiST / SWANS environment and is subject to the JiST Rewriter, but the SOAP proxy processes in the underlying JVM across the real network in native Java.

The runtime processing environment of the TrAITS is illustrated in Figure 27. The two separate component simulators each provide for a visualization interface which is synchronized through VanetSimulation bookkeeping tasks. Each vehicle within the VISSIM environment has an equivalent SWANS node. The SWANS environment recognizes and adapts the field configuration to match the geometry of the VISSIM simulation field and road network. The geometry of the roads themselves is not modeled in SWANS since it has no impact on the available wireless communication models. The SWANS and VISSIM environments communicate through the standard TCP/IP network and may be either in separate processes on the same
machine or on different machines / virtual machines. The Java implementation of SWANS may be run in a standalone or server JVM.

The web service component in this VISSIM implementation runs on a Windows-based machine within an ASP.Net Web Server. The Windows environment is required to allow processing of COM and the VISSIM application. Implementations using different transportation simulators may implement the web service interface on any operating system.

Synchronization of the two environments is managed by VanetSimulation which communicates through the SOAP proxy to call the web services (Figure 27). These web services process requests to make basic or more involved calls to the VISSIM COM API which updates the VISSIM environment. The results of these updates are read through the COM API, formatted within the web services, and returned VanetSimulation through the SOAP proxy.

Figure 27. Example runtime processing of the TrAITS environment.
The singleton VanetSimulation handles all elements of interaction with the transportation simulator (VISSIM) including the initialization stages and management of VISSIM time steps on behalf of the JIST controller. The member classes of VanetSimulation reflect the state and methods of the underlying VISSIM simulation elements. The components of this class replicate the transportation network in the context of SWANS. Figure 28 provides a partial class diagram for VanetSimulation and its member classes and highlights some key data elements and methods used to model the associated transportation environment.

Upon first request of a VanetSimulation instance, the class constructor and initializer processes call the SOAP proxy to start VISSIM, load the transportation network, and initialize the vehicle traffic. The VISSIM field geometry, time step frequency, and other factors are then
returned and used to initialize the SWANS simulation field. After processing this initial sequence, all required transportation elements are gathered from VISSIM, mapped into their local equivalents, and cached within VanetSimulation. During this initialization sequence, and when necessary afterward, the nodes within SWANS are created using the callback method. This event driven approach to initializing and maintaining network elements between the environments is one of significant processing model improvements made to the SWANS / SN architecture as part of this research.

VanetSimulation contains a callback member interface which allows dynamical addition and removal of Node elements within the SWANS / SN environment based on updated information from VISSIM. Simulations simply implement this interface to enable this capability instead of the standard static node count of the underlying SWANS. With each request of a VISSIM step, VanetSimulation gathers elements from VISSIM, resolves them to their previous SWANS instances, and updates any local state information. Elements without SWANS representations have new instances created. SWANS nodes no longer reflected in VISSIM are removed from SWANS by disabling the node and node GUI. In addition, the mapping between IDs representing the same object in the separate SWANS and VISSIM accounting is also resolved by VanetSimulation.

The VanetSimulation member classes constitute a series of hardware APIs when used in conjunction with the SWANS / SN Node implementation. Each SWANS Node created as a result of this component is associated to a specific instance of a member class. These member classes in turn reflect an element in the transportation network, i.e. a simulated hardware element or machine. Mapping a device to a SWANS Node effectively provides it a wireless radio, thereby enabling its participation in ITS interaction.

State requests from the VanetSimulation member classes retrieve the most recent VISSIM state data from the local cache. In addition to state access, certain methods also allow modification of certain the parameters, such as driver route, for the underlying transportation elements. Calls made to these methods trigger the appropriate calls to the proxy, web service, COM API, and VISSIM such that the transportation simulation is adjusted appropriately. Where
certain calls require more complex adjustments to the VISSIM model, they are bundled into complex web service calls to reduce network traffic and simulation lag. Other methods relate only to the local implementation of the transportation element and allow VanetSimulation and its member classes to augment the available transportation option, providing a mechanism by which future research may adjust transportation-specific behavior, implement additional features, or model more specialized actions upon the existing interface.

Vehicle state information is available at each time step to provide current information on various state elements such as speed, direction, number of passengers, and safety controls as available in VISSIM. This information has applicability to monitoring the global state of a traffic network as well as individual characteristics for the purpose of ad hoc networking or emergency response. Sensors and actuators are modeled as needed using elements relating to items in the VISSIM network. Additional wireless devices may also be modeled without transportation network equivalents.

Figure 29. TrAITS architecture detailing interface and control components.
Figure 29 details the architecture elements of transportation and wireless network simulation. The transportation simulation interface provides a complete abstraction of the underlying transportation network within the SWANS environment and manages all interaction with the underlying simulator, in this case VISSIM.

### 7.6 Simulation Control and Agent Modeling

The SWANS simulation platform includes a basic infrastructure for building applications upon the simulation environment. This framework is enhanced somewhat in the SN environment with the availability of an improved network implementation for use in application modeling. This research utilizes application built upon the SWANS/SN application interfaces to model the interaction of transportation and ITA agents over a wireless network infrastructure under the mobility constraints of the VISSIM simulator. Through the resulting framework, each agent is aware of the underlying characteristics of the transportation element it represents and able to leverage this sensor data while interacting with other network elements. Travelers are modeled within this design as intelligent agents which not only have access to state elements from the transportation network, but have a framework upon which to make informed decisions and enact changes through communication with other network elements or the transportation simulation interface. Each component of the environment is accessible through some form of application interface or service layer such that any connected component in the network may interact to both sense and actuate.

To facilitate simulation control and modeling the behavior of components and travelers in the network the SWANS / SN application driver and Node interfaces were used to build an ITS simulation controller application. Along with this controller a number of modifications and additions were added to the SWANS and SN infrastructure to tie in with transportation network services and enable awareness of vehicle positions and paths. In addition, the ITS application controller provides a member class which implements the callback interface, thus connecting the transportation and wireless simulation components with a single driver.
Finally, a series of agent components (Figure 30) were built to facilitate the autonomous and intelligent interaction of wireless devices within the network. These agent implementations were built upon the SWANS application interface to provide basic capability available to applications running within SWANS. Simple autonomous agents implement AutonomousAgentInterface and respond to state requests, signal alerts, and requests for state modification. They are used to model wireless enabled signals and sensors within the network. Intelligent agents implement IntelligentAgentInterface and provide a mechanism to adjust respond to information and coordinate activity with other network elements. A basic ITA GUI application was built within the SN net framework to provide access to the state information of nodes within a simulation and allow the simulation user to modify the parameters or travel path of individual travelers.

Figure 30. The ITSAgent intelligent and autonomous agent class model.
The resulting architecture shown in Figure 31 provides insight into the interaction of elements within a simulation and the conceptual model of the simulation across the key components of SWANS / SN, the transportation simulation interface, an application simulation controller, and a series of agent components upon which to model device and traveler behavior.

The mobility model used within the combined architecture implement the mobility interfaces of the SWANS architecture to provide a consistent simulation framework for all simulations running on the platform. Other wireless elements may implement the many mobility interfaces available within SWANS / SN as well. Transportation network elements may be mobile or stationary. The immutable StaticMobility class represents stationary elements, such as signs and parking lots, and is initialized with a specific geographical location at simulation startup and not updated throughout the simulation. The mutable VanetMobility class tracks the most recent location of moving VISSIM objects such as vehicles and pedestrians. VanetMobility locations are updated whenever the VISSIM simulation is stepped and locations have been adjusted. For convenience the VanetMobility class stores the most recent and previous locations. These models are compatible with all existing SWANS features.
To accommodate the modeling of GPS devices within the framework, a general GPS implementation was added to the SN framework which reflects the latitude and longitude information from VISSIM and provides a reasonable substitute for the output of a real GPS receiver. Those wishing to model the physical characteristics of a GPS device may wish to build upon this base to add components of availability and error to the exact values currently available from the core VISSIM model.

Finally, adjustments were made to facilitate the development of broadcast and routing algorithms based upon vehicular models such as those discussed in Chapter 4. Within the SWANS and SN MAC layers for the 802.11 and 802.15 standards, features were added to support awareness of the vehicle position, heading, speed, and acceleration as well as travel path. This provided the foundation to discount or ignore wireless signals from parallel roads, vehicles traveling in opposite directions, or different travel paths as well as to implement broadcast or routing algorithms which consider transportation network location rather than simple position for more efficient routing or broadcast. Although not practical for a general MANET use
these additions enable many practical applications for VANET modeling and algorithm development.

The VISSIM and SWANS applications currently each provide some form of aggregate statistics based on transportation parameters such as emissions and fuel or network parameters such as bandwidth or latency. To facilitate common logging and statistics gathering across the environment, the transportation network data gathered from VISSIM through VanetSimulation was combined with the SWANS data into additional elements built upon the SN logging module. Additional log parameters following the basic architecture of SN were designed as needed to facilitate easy analysis of both wireless and transportation statistics in atomic or aggregate form following a common interface and provide a more robust library of observation elements to ease the analysis of VANET network performance.

### 7.7 Traffic Simulation Service Library

The transportation simulation service library facilitates the abstraction of the transportation network simulator into the generalized model represented by the VanetSimulation library. As the lowest level interface to the transportation simulator API, the traffic simulation service library translates calls from the wireless simulation platform into API calls and control changes in the transportation simulation and returns the results of these calls to VanetSimulation through the SOAP proxy. This library also defines a structure upon which other transportation simulators may be integrated as an alternative to VISSIM.

The traffic simulation service library implements a number of key simple and complex methods upon the VISSIM COM library to help synchronize the data and activity of VanetSimulation. A sample set of the web service methods is listed in Table 2. The most basic of these method calls simply steps the simulator a single time step forward by triggering the equivalent call in the VISSIM API. Other services provide more complex functions such as initializing the simulator or triggering traffic scenarios such as a broken down vehicle using techniques for the represented transportation simulator.
<table>
<thead>
<tr>
<th>Service Call</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stepOnce</td>
<td>Triggers a call to the VISSIM API step method to advance the simulator one time step.</td>
</tr>
<tr>
<td>initialize</td>
<td>Opens the associated transportation simulator and loads the desired simulation network. Various other steps of transportation network initialization are also performed including priming the network with initial traffic flow. After initialization, the geometry and initial settings of the network are returned to allow equivalent geometry in SWANS and synchronization of time steps with the equivalent SWANS steps.</td>
</tr>
<tr>
<td>addPassenger</td>
<td>Adds a passenger to a specific vehicle.</td>
</tr>
<tr>
<td>changeSpeed</td>
<td>Changes the vehicles desired speed.</td>
</tr>
<tr>
<td>changeColor</td>
<td>Changes the vehicles color (used for debugging).</td>
</tr>
<tr>
<td>detourAll</td>
<td>Closes a specific traffic lane on a link such that all vehicles must detour around it.</td>
</tr>
<tr>
<td>setAttribute</td>
<td>Changes the value of a specific attribute on a network element. Various translations are performed to adjust VanetSimulation equivalents to VISSIM.</td>
</tr>
<tr>
<td>getVehicles</td>
<td>Collects and returns the set of vehicles from the network.</td>
</tr>
<tr>
<td>getParkingLots</td>
<td>Collects and returns the parking lot elements from the network.</td>
</tr>
<tr>
<td>getLinks</td>
<td>Returns all the road and intersection elements.</td>
</tr>
</tbody>
</table>

Table 2. Subset of services in the TrAITS Traffic Services library.

Alternate transportation simulators may be swapped into this platform by implementing the TrafficServices interface defined for the traffic simulation library.

The actual implementation of the web services may be in any language or run on any operating system or server provided the interface is followed and the SOAP proxy is configured to recognize the implementation. For compatibility with the implementation of VISSIM’s COM API, the VISSIM web service library has been implemented using the Microsoft .Net framework and runs on the ASP.Net web server on the Windows platform.

### 7.8 Traffic Simulation Model

This research primarily used two traffic networks for simulating VANET behavior in an urban environment. Additional models available as sample sets from the VISSIM vendor were utilized
for testing basic components of the implementation. In addition, real-world traffic trace data was used to verify the SWANS components of the simulator based upon the movement of vehicles in the characteristic study of a later chapter.

The first network modeled was of Boise, ID and covered a section of downtown Boise with detail into the roads and signals throughout the network as well as special considerations such as traffic slowing areas. The Boise network provided a well defined simulation zone with a pre-calibrated traffic model upon which various features of the simulator could be tested. This model covered the roads and highways in downtown Boise and many types of vehicles, but was limited in coverage of multiple modes of transportation. This disadvantage was not a result of the model itself, rather the reality of the transportation network in Boise. Unlike larger cities, Boise does not have a standard train or tram line running in this section of the city.

Figure 34. Aerial photo of Boise, ID (© 2011 Google).
For comparison purposes the aerial view of Boise, ID is detailed in Figure 34 using the Google Earth view of the location. The network section modeled within VISSIM is shown in Figure 35 which illustrates the accuracy of road network configurations in VISSIM versus real world views. The actual model conforms to the latitude and longitude configuration of the road segment surveys available through multiple sources. Each intersection within this model accurately depicted the signals and lane configurations of the real Boise network.

The Boise model reflects a static flow of transportation in the section. VISSIM has the capability to run dynamic traffic assignment (DTA) models which act upon OD matrix input to model the flow of vehicles between zones following multiple routes based on traffic scenarios. This sort of model converges to a steady flow state after a number of iterations, so VISSIM offers an additional option to store the traffic state into static models with tightly defined vehicle input and output locations along the network without the dynamic adjustments. Unfortunately, VISSIM static models do not allow adjustment of vehicle parameters or traffic rerouting.

Figure 35. Boise, ID VISSIM Network.
VISSIM static network models provide an accurate and repeatable view of the vehicle mobility within the network and are a very good option for modeling the mobility of vehicles using the simulation platform when adjustment of vehicle state or dynamic rerouting is not under study.

A Chicago, IL model was provided by Mygistics / PTV America (Mygistics, 2011; VISSIM, 2011) to accommodate the need to verify the dynamic assignment features of the integrated simulator. The network provided covers the critical road and highway sections of the Chicago area with signal timing and vehicle zone count information at 10 minute intervals. Figures 36 and 37 compare the aerial view of Chicago to the provided network. The network is currently under expansion by Mygistics. In addition to the road and expressways of the Chicago area, this model provides details into the Metra and CTA train lines servicing the city, thus providing a platform upon which the interaction of a traveler with multiple modes of transportation can be modeled.

Figure 36. Aerial photo of Chicago, IL (© 2011 Google, © 2011 TerraMetrics).
This Chicago model provided the basis for the verification of the dynamic elements of the VISSIM simulator and enabled the verification of ITS applications which are not only aware of the state of the transportation network but may also take action on any particular vehicle to improve an individual driver’s objective.

Figure 37. Chicago, IL VISSIM network.

7.9 Smartphone Simulation

In addition to the simulation of basic wireless and VANET applications, a critical test of the integrated simulation environment will be its ability to simulate typical client-server style applications such as those used by next generation cellular devices such as Apple’s iPhone or Android devices.
The operating system powering the Apple iPhone and iPad, iOS, is built upon traditional cellular networks but also capable of WiFi communication providing an ideal platform for building applications for intelligent transportation (Apple, 2011). Apple’s iPhone is built using a proprietary hardware platform with advanced control features for navigation and user input. The software platform for the device utilizes the Objective-C language and a series of API interfaces for network connectivity and server interaction. The iPhone does not provide native support for the Java language upon the iPhone, but this does not hinder our ability to model the client-server interaction typical of the device using the designed simulator. It has also been shown (XMLVM, 2009) that applications built using standard libraries may be converted to Objective-C using byte code translation such that they may be ported to the iPhone. The model offered by iPhone and iPad devices, as a result of manufacturer restrictions, allows client-server communication only. Many applications exist, however, which enable peer-based communication using server-based rendezvous or sharing.

The Google Android operating system is installed on a number of consumer devices including smartphones and tablets. Android is based upon a Linux operating system utilizing a customized Java Virtual Machine which offers tighter integration into the underlying hardware and provides a robust application development environment (Android, 2011). Applications are developed using a combination of Java and XML following a framework defined within the operating system. As with devices running iOS, Android devices typically have multiple wireless interfaces allowing communication across Bluetooth, cellular, or 802.11 radios. Android devices enable both client-server and peer-to-peer based networking.

The simulation environment provided by the TrAlTS agent model provides a solid foundation for testing various client-server applications that interact with the infrastructure of the platform. Although not directly modeled against the iPhone or Android APIs, the simulator can provide results which are consistent with the interaction of smartphone users and provide results in line with true use of these applications in a client-server paradigm. Server interfaces within the simulation environment are modeled as using the standard network interfaces of SWANS using predefined access points. Access points may be modeled using wireless node interfaces such
that communication from any simulation node to a wireless base station is modeled using the same framework as peer-to-peer communication but may use different parameters for fading, radio configuration, network protocol, or error rates. Peer-to-peer applications, such as those becoming more common on smartphone platforms, are easily implemented using standard Java language constructs.

7.10 Summary

The goal of this research was to design and implement a framework for the verification and analysis of coordinated traffic activities across multiple models of transportation and aided by wireless enabled devices for tasks such as ride sharing, multi-mode origin to destination mapping, and traffic prediction. The TrAITS simulation environment designed in this work provides a foundation upon which to model ITS applications and the behavior of travelers in multiple forms including those integrated with vehicle hardware or provided as applications on user devices. Many such applications are already under development by other research teams.

As detailed in this chapter, simulation evaluation must be performed for vehicle traffic conditions in different times, areas, communication parameters, etc. Additionally, more accurate real-time traffic data should be employed for the purpose of reliability between simulation results and actual transportation conditions. This research defines a framework which enables advanced analyses following the well known patterns of both transportation and network science. Without significant investment in building or verifying the simulation environment, a user may leverage the design of this chapter to model a user traveling through and interacting with a multi-modal environment similar to the real-world in which applications may be deployed but at a fraction of the investment.

A principle component of this research is the documentation of the developed simulation platform and architecture components. The desire is to provide a basis for further research leveraging in part or whole the results of this research. As such, components of the TrAITS architecture are documented in various forms in the Appendix. In addition, source code, architecture diagrams, quick start guides, and integration tips are available upon request from the
author. Sample applications for simulations within the environment are detailed in the next chapter and available in the source bundle.

The platform detailed within this chapter provides the foundation for development and validation of services following a cooperative model which depends on the contribution of participating vehicles, device users, and infrastructure components. The resulting simulator models the complete interaction of ad hoc wireless and transportation simulators into a single ITS network simulator which and provides features upon which to model traveler behavior thereby providing a robust long-term research platform for further development and analysis of ITS components.
8 Simulation Controller and Sample Applications

8.1 Controller Application

Simulations run upon the TrAITS are initialized using a driver application which initializes the simulation environment, and starts the processing sequence. In typical SWANS or SN simulations the driver accepts the simulation input parameters and initializes the SWANS simulation field, a fixed set of wireless nodes, and a duration for simulation execution. Within TrAITS a simple driver application is coupled with a VanetController class which builds upon the driver model from JiST, SWANS, and SN and incorporates the initialization and configuration of combined ITS simulation.

The initialization of the VanetSimulation singleton and SOAP proxy occurs upon entry into the driver application. Once initialized, VanetSimulation triggers the load of a VISSIM simulation and gathers the field dimensions for the transportation model. These dimensions are used by the driver to initialize the SWANS field. Unlike standard SWANS models, the duration of simulation and number of nodes are not initially configured. The simulation duration is instead controlled within the JiST Controller in combination with the VanetSimulation utility methods. Node creation is deferred to the callback interface implemented by the VanetSimulationController which allows addition and removal of nodes based on transportation simulation activity. All wireless nodes are created using this standard callback interface such that a single point of configuration is available for simulation setup. VanetController encapsulates all simulation parameters and acts as the simulation controller. This model improves the setup and configuration when compared to SWANS / SN equivalents.

VanetController has two main purposes. First, the constructor of the class initializes the simulation field as well as any radios, protocols, routing algorithms, etc required by the simulation. This initialization also configures the SN GUI, when enabled, and defines the statistics to gather during simulation. Second, the class implements the callback methods which enables the dynamic creation and removal of SWANS nodes in response to transportation simulation activity.
Figure 38. The VanetController class which initializes and manages simulation.

Figure 38 shows an abbreviated class diagram for the VanetController class. The external interface provides the basic capability to initialize a simulation and add or remove elements from the simulation.

createNode(…)

<Instantiate Node Instance>
  <Instantiate GPS>
  <Initialize Radio Noise Model>
  <Initialize Battery Model>
  <Initialize Energy Consumption Model>
  <Create Radio>
  <Create MAC>
  <Assign Routing Protocol>
  <Assign Network Protocol>

<Instantiate Agent Instance>
  <Link Agent to Wireless Node>

<Start Agent>

Figure 39. Pseudocode for createNode method of VanetController.

The createNode method configures the SWANS wireless node options and initialized the agent capability for the node. The key actions performed by this method are outlined in the pseudocode of Figure 39. Note that once created, an individual wireless node’s communication and activity is handled by the agent model assigned to the node by this method. The agent is a
Java wireless application which runs atop the SWANS simulator and handles the activity for an individual wireless node.

8.2 Preference-Action Model for Behavioral Characteristics

As an illustration of options available for implementing advanced user behavior models in the TraITS simulation environment, an agent was created to model the basic interaction of an intelligent traveler with the wireless and transportation networks (Figure 40). This approach is intended to illustrate the implementation of more advanced user modeling such as that described for the ADAPTS system (Auld and Mohammadian, 2009a). The ADAPTS model works on the principle that a user’s driving behavior, path, and schedule is the result of individual preferences and activities which affect their trip. The ADAPTS model does not require the implementation of a wireless network for all aspects, but certain tasks require the coordination of external devices to gather up-to-date transit information.

To model the preference of an individual traveler in the ITS environment an agent was implemented based on a preference threshold response model. In this model, the behavior of an individual user is the result of a series of individual tolerances for aspects of the transportation model. The breach of any predefined tolerance threshold triggers the user to perform some transportation activity. In this example two actions were outlined for the agent based on common traffic situations for which users may adjust their behavior:

1. Emergency Reroute: This behavior is the result of a notification that a portion of the traveler’s route has slowed or shut down because of some other activity. Once a
certain travel time limit is reached, the traveler’s response is to either change paths or return home (represented by the starting zone).

2. Alternate Parking Lot: In this behavior the traveler is notified that the parking lot they prefer has become full. The action by the traveler is to choose a new parking lot and reroute itself to that lot.

Figure 41. Agent processing flow for preference-action model.
Figure 41 details the processing flow of the agent’s reasoning. Each agent defines a listener which becomes the entry point for radio messages. Messages first travel through the SWANS simulated wireless network from their originator. Once received a message is passed up the network stack into the appropriate layer. Messages intended for the agent propagate to the agent layer and are processed by the receive method.

The receive method is the entry point for all agent wireless communication and acts as a handler to disseminate the source, type, and other characteristics of the message. Within the BehavioralAgent, the receive method starts the agent’s processing flow. Once a message has been received and decoded into a readable format, the agent determines whether it is affected by the contents by checking preference thresholds specific to the message type and contents.

In the case of a parking lot notification the agent searches for the closest alternative lot and reroutes the vehicle by changing the destination lot through the vehicle instance reference. This triggers VanetSimulation utility calls to modify the VISSIM equivalent of the vehicle and direct the vehicle to a new destination lot.

In the case of an accident alert or travel time update, the agent will determine if the travel time threshold will be breached and search for alternative routes. If a valid alternative is discovered the vehicle the VISSIM network is notified to change the vehicle’s path. If no alternative is found, the driver’s threshold to cancel the journey is evaluated which may trigger a request to route the vehicle back to its starting location within the VISSIM simulation.

### 8.3 Active Driver Alert System

Another common model for ITS implementation is the creation of active alert systems. These systems are typically implemented as information signs alongside or above the roadside which display messages based on traffic situations downstream from the driver. Another implementation is as painted road markings or rubble strips embedded within the road to alert the driver to the changing condition. To benefit from these messages a driver must often read the sign or acknowledge the rumble strip and decide whether to adjust their driving. In the simplest form, these messages may warn of a traffic slowdown or stop, but many other situations may
follow the same model such as alerting a driver of a road repair crew, slippery surface, or accident. An alternative to this approach is to provide an active response system within the vehicle. In this model, signs and signals periodically broadcast notifications of their state. The vehicle receives and interprets these notifications and determines the best course of action for the driver.

The motivation for the active driver alert system is that 20.8% of traffic fatalities are intersection related. This percentage has not decreased significantly despite additional signage, road markings, rumble strips, or flashing lights (FHWA, 2009) so a new approach may be needed to further aid the safety of intersection traffic.

The model of Figure 42 illustrates the components of a vehicle’s stopping distance along the vertical axis at various speeds along the horizontal axis. The standard stopping distance is composed of a number of factors but primarily results from vehicle speed and driver reaction time. This reaction time factor represents the distance traveled during the time from which a
driver first perceives the need to stop to when they have moved their foot from the accelerator to the brake pedal. A further component of stopping distance may be weather related slippage as a result of slippery road surfaces.

The horizontal lines within Figure 42 represent the perception thresholds under various conditions. Under normal visual perception the driver has an unobstructed, fully visible view of the sign and may react normally. As weather or nighttime driving affect visibility the perception distance decreases and thus decreases the range of safe travel speeds at which a driver can adequately stop the vehicle.

The goal of the Active Driver Alert System is to increase the perception distance for the vehicle by providing wireless notification to the vehicle and driver such that automatic and driver initiated adjustments may be made to stop the vehicle in with a consistent stopping distance despite visibility or driver alertness conditions.

Figure 43. The active driver alert agent classes.

To model an active driver alert system, an implementation is suggested in which low-cost wireless transmitters embedded in roadway signs and signals alert drivers of the state of a signal or the existence of a sign. It is assumed that the wireless communication would leverage dedicated bands for communication as is standard for traffic safety devices and thus the interference from other signals would be limited.

To implement this model, three agent devices (Figure 43) were created modeling a signal, sign, and vehicle within the TrAITS environment. In this implementation, the autonomous signal agent provides a broadcast of its location and state and allows incoming connections from drivers to query its state. As an alternative, the autonomous sign agent sends periodic broadcasts of its type and location to alert drivers and cannot receive communications. The intelligent vehicle
agent models both vehicle and driver and responds to the signal broadcasts appropriately based on type.

The initial reaction to a signal is automatic such that the vehicle's agent receives the signal and reacts by decelerating the vehicle. The implementation assumes that the driver of an enabled vehicle would perceive the deceleration and react accordingly or the vehicle would provide an audible alert to notify the driver of the impending stop.

Three scenarios are modeled in the current environment:

1. Active-Braking for Stop Signs: A stop sign broadcasts its position. A vehicle receives a broadcast, determines the relevancy of the signal, slows the vehicle, and notifies the driver or the action.

2. Red Light Alert: A Red-Amber-Green (RAG) signal broadcasts its red status and position. A vehicle receives the broadcast and slows the vehicle while notifying the driver.

3. Intelligent Signaling: A RAG signal broadcasts its state. A vehicle receives the broadcast and communicates with the signal to determine the signal timing. Based on the current color and time to color change, the vehicle determines whether the driver should maintain the current course or slow down to stop. In this more complex example, the vehicle communicates with the signal through the SWANS network to determine the change timing and recommends the best approach to the driver. Note that a similar implementation could be provided by broadcasting the signal state and change time, but the intent is to illustrate the basic capability for more advanced options which may build upon it.

Each of these scenarios assumes the agent is location-aware through GPS devices or programmed locations. Each vehicle also is able to determine the relevancy of the signal / sign broadcast based on the information within the broadcast packet, specifically type, state, location, and heading. Figure 44 illustrates the general processing flow for the three agents.

Signal state and location information is taken from the VISSIM network. The vehicle actions resulting from this notification trigger changes in the VISSIM vehicle state. The VISSIM vehicle
still follows the vehicle movement model, but this system adjusts speed proactively to in response to signs and signals.

8.4 Summary

The examples within this chapter and associated source code illustrate the ease of modeling complex ITS applications within the TrAITS simulation environment. Within TrAITS, complex wireless network and transportation models have been abstracted to allow rapid prototyping and verification of agent-based ITS applications and services. The underlying SWANS architecture components allow modeling of various network protocols and algorithms to provide a complete
environment for modeling VANET activity. The movement and actions of each agent triggers the appropriate calls within the VISSIM network to maintain consistency with the transportation simulation model.
9 Wireless Simulation Field Analysis

9.1 Overview

The simulation of IVC in a VANET requires the modeling of traffic movement patterns and the simulation of wireless communication between the vehicles which act as mobile nodes within the wireless network. This chapter discusses the fitness of the JIST/SWANS network simulator for use in VANET simulation by comparing its results to a series of tests run with readily available wireless hardware devices. Although this study involves the creation of a hybrid VANET simulator coupling wireless and transportation environments, this chapter studies the characteristics of the wireless environment modeled within the SN/SWANS simulator components. Although the ultimate goal of this research is to model the behavior of ITS devices in a VANET environment, a primary concern of this study was the validation of the ad-hoc wireless portion of the SWANS simulator.

This chapter discusses the calibration and comparison characteristics between the wireless simulation model and ITA test devices. To better understand the tradeoff of simulation versus a real-world implementation, a basic verification experiment was devised which highlights differences between the two platforms. The sections which follow detail the experimental setup, procedure, and results of a test between two mobile vehicles in a simple urban environment under the 802.11 wireless network protocols.

Real-world ITA (Haran et al., 2008) device or application testing requires investments into the development of hardware and software platforms (Choffnes et al., 2005; Haran et al., 2005; Fan et al., 2006; Fan et al., 2005; Killat et al., 2007), the development of infrastructure and services for the support of ITA users, as well as a sufficiently large adoption of the device usage before metrics on the device become valuable in understanding the characteristics of device reliability and response in a live environment. This model leaves the refinement of communication protocols and routing algorithms as a reactive exercise within which development teams must adapt to the information gathered from devices and manage software and hardware updates.
before carrying out additional tests. As tests become more complex, repeatability of experiments becomes increasingly difficult and experimental turnaround time grows prohibitively long. These challenges make the simulation of these devices in a software environment a necessary step in the design and development of experiments related to ITA devices. Through the use of simulation, the hope is to adequately model the behavior of multiple users interacting with each other and infrastructure under the protocols of various network protocols, clustering, and routing algorithms with the goal of reducing network bottlenecks, increasing connectivity, and aiding the predictability of a future hardware-based implementation.

The TrAITS platform provides an integration API which aids integration between SWANS and VISSIM platform through a series of shared services and event triggering framework. This API provides the framework for simulation based on real-world node location data point traces rather than the standard SWANS mobility models and is the input mechanism for validating the real world data collected in this experiment within the simulation environment. Leveraging the open architecture of the hybrid integration components, the data gathered from real-world experiments were easily input into the SWANS environment for comparison to simulated results.

Previous research (Kotz et al., 2004) highlighted a series of six axioms in common use across most wireless simulators. These axioms, numbered from zero for consistency with the original presentation, are summarized as:

0. The world is flat. This assumes that all radios are positioned at equal height and distributed across a flat terrain with no earth curvature. This idealized form facilitates the modeling of radio transmission using a simple two-dimensional plane and reduces computation within the simulator.

1. A radio’s transmission radius is circular. The transmission of a radio’s signals is affected by many factors including the design of the transmitter and environmental factors such as weather conditions or obstacles. Actual signal transmission, as shown in the results of this chapter’s experiment, is much more geometrically complex than a simple circle. Most simulators, however, has a base assumption that the signal propagation will follow well defined stochastic models with underlying assumptions of
circular transmission and an inverse relationship between signal strength and the distance from the transmitter. An implication of this assumption is that most wireless simulation models assume a cliff-effect within the communication such that the reception has a zero-probability outside a communication radius and a probability of 1, or some other stochastically determined value inside the radius.

2. All radios have equal range. This assumes that the range of the radio is a well-defined value specified by a manufacturer and fixed across all versions of the radio. Similarly, it assumes that all radios manufactured follow the exact same specification and have identical properties. The reality is that radios and the components of radios, transmitters, receivers, etc, are electro-mechanical devices which are prone to manufacturing and differences as well as many other conditions which affect their capability. Simple characteristics, such as the height or angle of position of the sender transmitter or receiver antenna can significantly affect the likelihood that communication will be successful.

3. Radio receipt is symmetric. This assumes that if radio A can receive a broadcast from radio B, then radio B can in turn receive the broadcast from radio A. This is challenging in VANET environments because both sender and receiver are moving at the time of communication. In the ideal sense, the devices have not moved or have stayed in the same relative position during the communication timeline. Even so, however, it’s unlikely, that sender and receiver will always mutually communicate. This has implications for the use of wireless devices in connection-oriented layouts requiring message receipt acknowledgement, but may be less of a concern in broadcast-only or connectionless communication such as emergency notification.

4. If a signal is heard, it is heard perfectly. Signals of any kind experience some loss under certain scenarios. This loss may be of the entire signal or of individual components or bits of the signal’s packets. Simulators rarely consider data loss within a packet but may provide a framework for assuming packet loss. In the case of SWANS, it provides models for zero packet loss (ideal communication) as well as
uniformly distributed random packet loss in addition to facilities for adding bit error rate (BER) considerations into simulations.

5. Signal strength is a simple function of distance. Signal strength is affected by many factors including attenuation as a result of weather or environmental conditions or obstacles in path of propagation.

To accommodate the gathering of experimental data to analyze the effects of these assumptions on experiments performed using the VANET simulator designed in this research and how simulated results may relate to the interaction of real wireless devices, a basic VANET experiment was designed (Haran and Nelson, 2010). Two vehicles carrying netbooks running Ubuntu Linux 9.10 and utilizing the Atheros wireless chipset running IEEE 802.11g with a built in antenna positioned in the netbook screen (Atheros, 2011) were equipped with a series of basic applications to log and monitor the wireless communication with one another and position at one second intervals. The wireless cards on the devices were each modified to run in managed mode and utilize the same wireless channel. For this test, a survey of the area local to the test site was performed which revealed that wireless channels 6, 7, and 8 were not in use in the local area. Wireless channel 7 was chosen based on this analysis to limit interference from viewable or hidden wireless access points or devices. The goal of this configuration was to limit the outside interference from wireless noise and allow focus on the results related to the broadcast reachability of the devices from one another in a mobile situation across a relatively small, flat test area. This method aids verification of the simulator’s reliability within the theoretical flat world environment (Choffnes et al., 2005).

Although recently ratified WAVE/DSRC recommendations warrant the use of the 802.11p standard for VANET applications, the use of 802.11g should not have significantly affected the results of this study since the analysis relied upon MAC-level reachability among notes in the network. For more extensive analyses, the recently ratified 802.11n standard (IEEE, 2009), offering more reliable connectivity at the same 2.4 GHz band as 802.11g, or the 802.11p (IEEE, 2011; Talk, 2008; ASTM, 2010; DSRC, 2010) standard which operates at the 5.9 GHz band
dedicated to ITS applications, may be more appropriate. Hardware implementations of these standards were not readily available at the time of study, however.

Each device in this study was paired with an i-Blue GPS receiver for the purposes of tracking actual location using GPS traces logged at regular intervals. The GPS traces, coupled with the result of the broadcast testing, provided a map of the locations where each node could receive the broadcast beacon from the other node. This setup required the use of Bluetooth to stream the GPS traces to the computer which is unfortunately known to sometimes interfere with 802.11 devices (Kamerman et al., 2000; Zyren, 1999; Lansford et al., 2001). This potential interference is however mitigated by Bluetooth’s frequency hopping model which ensures that interference, although likely, will be rapidly resolved as the frequency shifts. Therefore, the interference resulting from Bluetooth was expected to be minimal. Note that use of the 802.11p (ASTM, 2010) standard as recommended for ITS solutions would significantly reduce or entirely eliminate this potential source of interference.

The ITA devices were mounted on the vehicle dashboard to limit the signal interference from vehicle components with antennas positioned perpendicular to the path of travel. The vehicles were driven at a 20 mph in opposite directions around two city blocks such that multiple passes were made on the path and the same approximate GPS location provided multiple checks of the reachability. Using the wireless configuration utilities, the wireless devices of this experiment were each set at 20 dB, reasonable signal strength for any handheld wireless device. This same signal strength would later be replicated in the SWANS experiment.

Figures 45 and 46 highlight the signal loss as the vehicles traveled around corners in the real world environment using images reflecting the local environment. In the real-world experiment, the vehicles were able to communicate when in clear line of sight and heading toward one another. Results from the experiment were processed into trace paths and time-stepped simulations within Google Earth (Google, 2011) to provide a platform for visual analysis of the data elements with relation to the surveyed images of buildings and other structures.

The second portion of the analysis step involved recreating the live experiment within the simulation environment. This experiment did not require the use of the VISSIM simulator for the
interaction of vehicles within the network. Instead, the GPS trace information from the two vehicles was translated into point locations within the simulation coordinate system following the architecture discussed in previous chapters. As would be modeled in the hybrid TrAITS environment, the vehicle locations at each time step were updated within the SWANS using the position of the vehicles gathered from the GPS receiver. Once positioned within the network, each vehicle beaconed the other vehicle utilizing the UDP node discovery protocol. As with the real-world version, the results of the simulation test were logged to determine the visibility of each node within the simulation environment.

Figure 45. Vehicle A path and signal activity (© 2011 Google).
9.2 Propagation and Reachability

The results of the wireless simulation experiment showed very different wireless signal traces between the two environments (Figures 48 and 49) as the vehicle traveled through the path traces from the real world experiment. SWANS doesn’t provide a platform for modeling the line-of-sight obstructions like houses, trees, and parked vehicles which impacted the real world experiment. Instead, the SWANS simulator models signal noise using an additive noise algorithm based on signal propagation and path loss approximations. Although this model provides a form of testing the reliability of network routing algorithms within uncertain environments and signal-loss situations it doesn’t accurately model the expected behavior of the signal in a live production experiment.
In addition, the simple test of signal or beacon reachability performed during this study did not accurately model the use of devices in a connection oriented wireless scenarios. The expected use of these devices incorporates use of wireless services to relay transportation information among travelers within a network in addition to broadcast transmission and receipt. Previous studies (Haran et al., 2005; Fan et al., 2006; Fan et al., 2005) showed the necessity of wireless routing and clustering models in providing stability to ad-hoc communication paths, this test of simple reachability shows that the unpredictability of the network, especially under highly mobile circumstances, may be well under-represented by simulation and should be a primary concern when interpreting simulated results.

SWANS provides a propagation model based on freespace path loss with little consideration to the interference or fading effects of environmental conditions. Additional facilities are provided within SWANS to model random packet loss using the uniform packet loss model, but it’s apparent that more robust models for these characteristics are needed within the simulation environment to accurately predict real world behavior.

9.3 Environmental Effects

A well-known constraint of wireless simulations is the flat earth approximation used to model the paths along which nodes travel (Kotz et al., 2004; Choffnes et al., 2005). In true MANET environments, the flat earth, or Cartesian, approximations to node movement is entirely unrealistic. Nodes within a MANET environment simply do not follow two-dimensional paths. Interaction of the MANET nodes with their environment forces the path into three dimensions as wireless devices travel through buildings and other structures. In the VANET environment, the simulation of vehicles using a two-dimensional model under small areas may be adequate for modeling the behavior in simple urban environments. These simulations, however, will quickly break down for hillier real-world terrain. A flat terrain was intentionally selected for this experiment to reduce the simulation differences as a result of this approximation.

Simulators like ns-2 and SWANS have adjusted the flat earth model (Wang et al., 2007; Leung et al., 2006; Choffnes et al., 2005; Schroth et al., 2005; Wang et al., 2007) to consider ground
reflection and signal shadowing and but still do not adequately reflect the VANET implementation environments. Their implementations allow ground interference to provide signal reflection but take no account of the material structure of the ground at different locations which may affect the signal as a result of surface absorption and reflection. A configuration parameter also allows radio-specific heights which somewhat aid the propagation model. Any real-world wireless network environment will reflect the interaction of hundreds of material properties such as pavement or soil composition, vehicles, construction materials, and structural design each having significant impact on the propagation of the wireless signal. Modeling of these environments within SWANS must account for these characteristics or at least acknowledge that the models in use will not truly reflect the deployment environment.

With the large deployment of wireless-enabled devices and potential for interference across protocols sharing the same bandwidth, the potential is for wireless devices to have more interfering than communicating neighbors. It has been shown in similar studies that hidden neighbors also have a high potential for interfering with wireless devices (Walke et al., 2006). Initial surveys of the test area across multiple channels were performed to limit the effect of this in this experiment, but nevertheless this is an important consideration when interpreting simulated results.

9.4 Transmission Radius

The SWANS environment, like many other simulators, models the transmission of wireless signal using a circular propagation centered on the antenna. Simple factors such as ground reflection may be considered in this model, but the core assumption remains that a signal propagates equally in all directions as an inverse of the distance from the sender and signal loss is a result of random noise occurrences and excessive distance between sender and receiver.

This simplification is known to be inadequate for modeling the signal propagation within common wireless devices. In addition, simulators often assume that signal strength is a fixed property easily set using wireless configuration options. This simply does not hold true in real-world experimentation. Wireless device signal strength varies considerably between locations in all
environments and may experience spontaneous signal bursts of strength and range as a result of weather and materials in use in the environment or battery strength.

Signal propagation from an antenna is far more complex than a simply circular radius. The typical antenna in a wireless device provides signal strengths which vary in different directions based on the device's component design. It has been shown that antenna orientation is often as important as antenna positioning in determining the success of wireless communications. Simulation studies simply do not capture the intricacies of the wireless antenna positional effect in mobile environments. In the typical laptop or netbook computer, the device antenna is most often built into the screen such that the screen positioning has an effect on the signal strength. These devices are more likely to have somewhat stationary antenna than handheld devices with which the antenna is more frequently repositioned as the device is moved by the user.

The manufacture and factory calibration of various devices differs considerably by vendor as does the implementation of these wireless cards and antenna in various computers, vehicles, and handheld devices. The signal strength and propagation characteristics are directly tied to the antenna design implemented within the computer. Various factors such as computer construction material, battery strength, and laptop position affect the behavior of the wireless signal in a real device. Early attempts at performing these same experiments even showed widely inconsistent results from two identical laptops from the same vendor each with the same wireless card. The results of this experiment, however, used a new platform and did not appear to have this trouble.

An interesting characteristic of the wireless behavior was the position relationship between the vehicles. In many scenarios the vehicles were unable to communicate even at distances of only 100 meters. This, however, appeared due to the position of the wireless device in the front of the vehicle. Under other situations the devices communicated reliably at distances greater than 250 meters when traveling toward one another. Mounting the devices upon the dashboard in this experiment limited variability of the radio position relative to the moving vehicle, but did not limit the variations caused as vehicles passed one another or were positioned as various relative angles. Similarly, the materials used to manufacture the cars may have also contributed to signal loss.
Figure 47 displays the subset of vehicle paths through during which the vehicles are not able to communicate. At closest, the vehicles are 100 meters apart illustrating vehicle distance and antenna configuration have significant impact on wireless reliability. The arrows show the path of travel. Note the vehicles regain communication as they come within 100 meters of one another.

In this experiment, vehicles sent simple beacon signals at one second intervals. The analysis of the data gathered for each vehicle pairs the data sent by one vehicle with the signals received and responded to by the other vehicle. This resulted in a simple 4-tuple for each beacon message reflecting a <request, receive, acknowledge, receive acknowledgment> set of timestamps and GPS locations for each vehicle. This provided the necessary data to analyze the traces for symmetry of communication. As shown, the signals did not reflect symmetric communication. Each vehicle often experienced situations in which its signal could be received by another but
joint communication could not be established. This model may work well enough for the simple beaconing of status information and emergency alerts common in some VANET applications, but would prove troublesome in connection-oriented communication.

9.5 Cliff Effect

The experiment showed that the cliff effect common to simulation is not an accurate representation of the wireless model when compared to the real measurements. Although signal reliability and strength clearly lessened as the vehicles moved further from one another, the connectivity was not simply a factor of distance (when comparing Figures 48 and 49) it did not exhibit the exact distance related signal loss characteristic of the cliff effect as illustrated in the simulation results of Figure 50.

The factors affecting the propagation of wireless signals are considerably more complex than simple distance measurements. Weather related attenuation, simple fading, and shadowing, among other factors, interrelate to provide unpredictable and often unrepeatable signal effects in real world experiments.

In this experiment, the distance across between the two vehicles never exceeded 325 meters which is consistent with expectations for reliable connectivity in 802.11g configurations and the manufacturers stated limits. The connectivity, however, was much less than reliable when simply attempting to connect vehicles that were positioned around corners.

9.6 Summary

The results of this experiment showed the reliability of the wireless connections in a simple urban experiment covering two city blocks. The results show that communication becomes unpredictable even when simple line-of-sight obstruction occurs as a result of vehicle parts, houses, trees, and parked vehicles. As anticipated (Schoch and Kargl, 2007; DSRC, 2010; IEEE, 2009) and illustrated in Figures 45 and 46 the connectivity between vehicles degrades significantly when rounding corners or when in the presence of multiple obstacles such as parked cars, trees, and buildings.
The real-world experiment showed that the communication could reliably occur at distances nearing 300 meters, the approximate outdoor range provided by the hardware vendor (Atheros, 2011; Acer, 2011) when no line-of-sight obstructions occur. This was consistent with the idealized simulation environment within which no obstructions are modeled. The simulation did not, however, model the interaction of vehicles as they moved around corners or to different sides of the block leaving them susceptible to interference from trees and buildings.

Figure 48. Relative distance measurements between vehicles.

Figure 49. Successful mutual vehicle connectivity.
Figure 48 displays the distance measurements between the two vehicles used in the test based on the vehicle GPS positions. The characteristic flower pattern illustrates the movement of vehicles away and together during the simulation following a steady series of test loops within the test area with time reflected as clockwise rotation around the center from the top. Figure 49 displays the subset of points at which the vehicles were able to successfully connect to one another. Note the significant difference from the ideal connection activity of Figure 48. Figure 50 extends the study to the SWANS environment and represents a simple connectivity test using wireless nodes following the same vehicle paths from the real world study. The results display the cliff effect pattern common to wireless simulators.

Within the simulation environment the vehicles have reliable communication to distances within the 300 meter signal radius configured to match the manufacturer specifications (Acer, 2011). The interference and signal loss models in use did not model the pattern exhibited within the real world experiment. Communication between the two vehicles was most unreliable as the vehicles traveled away from one another indicating a reliance on antenna orientation and positioning within the vehicle is also not considered by the SWANS simulation (Choffnes et al., 2005; Schroth et al., 2005), nor is the interference resulting from obstacles such as a vehicle’s body or windshield. Although this analysis shows that the wireless simulation may not accurately
model the effects of real-world interference on wireless simulation, it does highlight some key configuration options which should be investigated further as part of wireless simulation studies as well as qualitative aspects of analyzing the results of any wireless simulation, especially those on ITS devices. In these results, the signal was somewhat reliable over longer distances when line-of-sight was not a factor.

The experiments conducted as part of this analysis highlight many of the differences between real-world experimentation and simulation and provide insight into some of the additional configuration options necessary in understanding the interaction of users and wireless devices in an ITA environment. As detailed throughout this chapter, simulation of wireless networks provides a sufficient model for analysis of protocols and route discovery algorithms but does not reflect real-world noise or interference and, therefore, may not be a good basis for studies requiring more realistic models for radio design, interference, or propagation. Simulation also fails to address interference from hidden radio sources which may be distributed throughout the network without the researcher explicitly adding these hidden nodes as art of the experiment configuration.

The result of the study detailed in this chapter provides insight into calibrating the TrAITS simulation platform and analyzing the results from VANET simulations to consider how well they may relate to real world results both quantitatively and qualitatively. The framework for recreating the experiments described within this chapter, as well as the data sets of all results, is available with the TrAITS simulator source code on request from the author.
10 Summary and Proposed Future Work

10.1 Summary of Research Contributions

The goal of this research was to design and implement a framework for the verification and analysis of coordinated traveler-based traffic activities across multiple models of transportation and aided by wireless enabled devices. The TrAITS simulation environment designed in this work provides a foundation upon which to model ITS applications and the behavior of travelers in multiple forms and thereby provides a foundation upon which future research may further investigate the properties of ITS environments.

As detailed throughout this document, the contributions of this research are the:

1. Design of the TrAITS integrated multi-modal simulation environment.
2. Creation of an agent-based simulation model as a foundation upon which to model traveler behavior across multiple modes of transportation.
3. Detailed analysis of VANET simulation and comparison to real environments.
4. Design of alternative VANET algorithms based upon the characteristics of vehicular movement.

Calibration and seeding of models with results gathered by observation of real traffic networks and wireless activity should also be a primary step in any use of simulation. This research leveraged calibrated transportation networks to provide a realistic simulation environment whenever possible. In addition, the field analysis allowed calibration of the wireless model based on observation of small-scale wireless tests. TrAITS leverages proven architectures of wireless and transportation simulators to provide a platform with known quality and performance and consistency with the state of the art of modeling in both fields. The quality of any simulation is determined by the capacity for the simulation models to represent the behavior of real environments. The simulation of wireless activity should consider that advanced characteristics of wireless networking such as signal interference may not be fully reflected in the model but do not impair the ability to gain insight into network algorithm behavior.
Simulation provides a tool with which researchers may model and refine their work while reducing or eliminating the variability experienced in real environments and is a valuable component of any analysis into the characteristics of ITS environment behavior. TrAITS provides a mechanism with which researchers may provide support for the implementation of devices and algorithms described by the statistical results of TrAITS simulation.

A principle component of this research is the documentation of the TrAITS platform and architecture components such that others may leverage in part or whole the results of this research. The TrAITS platform provides the foundation for development and validation of services following a cooperative model which depends on the contribution of participating vehicles, device users, and infrastructure components. This models the complete interaction of ad hoc wireless and transportation simulators into a single ITS simulator with features upon which to model traveler behavior.

Within TrAITS, complex wireless network and transportation models have been abstracted to allow rapid prototyping and verification of agent-based ITS applications and services. The underlying SWANS architecture components allow modeling of various network protocols and algorithms to fully model VANET activity. Similarly the VISSIM microsimulator provides a proven model for characterizing driver behavior in a complex transportation network.

This work explores several issues concerning the analysis and design of algorithms for use in VANETs and details key considerations for any researcher seeking to analyze the aspects of ITS through the use of simulation tools. This research serves as a foundation for others seeking to perform additional work in the interesting aspects of ITS. As a basis for further research the resulting source code, examples, test cases, and data sets are available from the author.

### 10.2 Transit Scheduling Elements

Transportation infrastructure falls into a number of different categories with various characteristics and connectivity options for the end user. For the purposes of this research, the infrastructure components necessary to simulate the driver behavior and integrate vehicle
sensors into an intelligent traveler application framework were implemented based upon the capability available in the SWANS and VISSIM environments. An additional series of infrastructure components may be implemented to allow interface of the driver with static schedules, stop locations, and other services available for existing transit lines. These services may provide additional information in for various transportation elements. States such as whether or not the train or bus is on-time, capacity of the transit element, and current location are readily within the VISSIM environment for integration with the wireless and agent platforms of the TrAITS environment.

The various transportation infrastructure components may be modeled using techniques to distribute and access the necessary services for real-time ITA interaction. These include the wrapping of readily available internet resources such as the Chicago Transit Authority (CTA, 2011), Metra Rail Service (METRA, 2011), or the Regional Transit Authority (RTA, 2011). In general, transit schedules may be modeled using two high level abstractions; those with fixed arrival and departure frequencies for various stations and time intervals, and those with fixed arrival and departure times. These abstractions allow efficient storage of multiple schedules. From a traveler perspective, public transportation may generally be understood with a functional basis. The traveler or ITS application services desire a predictable public transit network for the purpose of routing and arrival / departure time and alerting of possible delays or interruptions. In general planning situations, knowledge of all potential schedules within a various time frame is essential.

10.3 More Robust Profile Information

The model implemented within this research is intended as a proof-of-concept for the elements of a user profile information database. Additional elements may be built upon this framework to better model the historical decision making of travelers and enhance the series of decisions available within the network. The original model for an ITA device (Dillenburg, 2004) included the use of a centralized travel planning and profile database with which each traveler would regularly interact to store location information and receive update travel plans. The model
implemented in this research is based on a peer-to-peer communication model with a basic assumption that a gateway is provided to centralized services, but the implementation of a single intelligent central services is left as a future task.

Modeling of the ITA profile information will require development of a custom database interface to access and store state information from various ITA users. This ITA central server framework will also provide data for the management of vehicle trajectory information such that vehicles may travel independent of the central server interaction except when necessary to update route information based on knowledge from the vehicle such as slow traffic situations or from the server based on changes in other areas of the traffic network.

10.4 Additional Infrastructure Components

Certain infrastructure sensors and actuators were modeled in this research as necessary to implement the detailed agent applications. These infrastructure elements were based on components available within the VISSIM network and modeled using the VISSIM state information but implemented using a customized sensor framework. The VISSIM environment is capable of modeling many of additional infrastructure elements. These may be integrated as needed using the approach outlined within this document.

10.5 Expanded Traffic Network

The environments modeled within this research included a network for the Cook County / Chicago, IL highway system, the Boise, ID downtown urban environment, and a basic network for the Chicago, IL system of roads, expressways, CTA, and Metra lines. Microsimulation accuracy hinges on the robustness of the underlying transportation network model. Any continued research into the behavior of ITS applications in a VANET environment should seek to develop and maintain robust transportation network models based on real-world sampling of traffic data and up-to-date maps of roads routes of various transit elements. To aid this research the author was able to obtain a more robust transit network from research team at Mygistics (Mygistics, 2011), specifically Dr. Jingtao Ma. The stated goal of Mygistics is to build the world’s largest
contiguous transportation model by detailing the transportation networks of the United States, Canada, and Mexico in a single VISSIM model through mesoscopic and microscopic modeling (Mygistics, 2011). At the time of this research the Chicago area transportation model was in its early stages. It is recommended that users of this software integrate the latest models available from Mygistics as well as other sources to continuously improve the breadth and accuracy of the modeled environment.

10.6 Pedestrian Modeling

One reason the VISSIM transportation microsimulator was selected for use in this study was because it provided a basic model for pedestrian movement within the simulation environment. As this research has developed, so has the model for pedestrian movement in the VISSIM environment. The framework developed provides for the integration of pedestrians into the VANET simulation and provides the basis for modeling a traveler’s route across multiple models of transportation aided by a set of ITA services. It is recommended that future researchers leveraging this platform develop and test services which utilize the robust pedestrian model provided by VISSIM.

10.7 Additional Transportation Simulators

The VISSIM simulator used in this research is a commercial tool requiring an annual license for a specific set of licensed users. This poses obvious challenges to larger research groups. To accommodate these challenges the architecture for TrAITS has been designed so that the transportation microsimulator may be substituted with any available microsimulator which supports a basic set of driver or vehicle capability. As an alternative to using VISSIM, users are encouraged to integrate alternative microsimulation tools by providing a web service implementation conforming to specifications outlined.
11 Appendix

11.1 Modification to JiST / SWANS / SIDnet Libraries

This section is not intended as a full detail of the changes made to the JiST, SWANS, or SIDnet libraries during this research. Rather, the hope is to outline some of the key components changes to support features of the TrAITS hybrid microsimulator which have not been covered in the text to date.

To support the dynamic creation and deletion of nodes within the wireless simulator during runtime and provide features consistent with VISSIM, the following changes were necessary:

- **Field**: The SWANS field library was modified to allow nodes within the simulation field to appear or disappear at any point in the simulation. This required handling of null exceptions and considerations when radio signals received originated from radio sources no longer present.
- **GroupSelectionTool**: This tool was modified to consider adapt to node addition or removal.
- **TopologyGUI**: This library was modified to support referencing a shared node collection to reduce memory consumption and copying resulting from copying arrays of nodes throughout the application. These changes also facilitated the event triggering logic added to support interaction between the simulators.
- **Node**: A number of methods were added to support the audit of node life and the killing of nodes. These methods are used throughout the framework to reduce the overhead of the GUI components and a mechanism for disabling nodes after their lifetime in the traffic simulation has expired. Note that disabled nodes are still maintained for statistics collection purposes.
- **NodeFinder**: This class provides utility methods for storing and finding nodes located within the simulation field at user request. To accommodate node deletion the data
structures within were modified and better performance was achieved by using HashMap data structures to rapidly add and delete nodes.

These changes were made to the simulator to support additional messaging options based on VANET scenarios:

- **Route**: The Routing Protocol libraries were modified to allow broadcast messages to trigger response in the application layer of the receiver. Previously these broadcast messages were discarded when sent on the same MAC as other broadcast messages but not related to routing.
- **MAC**: The MAC interface libraries were similarly modified to support the bubbling of alert broadcast messages, such as those in an emergency warning system, up to the application layer. In addition, methods for originating and handling these messages were implemented following standard protocol implementation. Finally, additional logic was added to support multi-hop broadcast relay as has been proposed in certain strategies to void routing maintenance overhead.

Additional changes were made to support statistics gathering and tie the statistics framework into the StatsCollector component of the SiDnet framework. This enabled the centralized maintenance of both wireless and transportation statistics.

These changes were made to support the development of an ITA sample application which provides access to the features for a specific node within the network and allows any user of the simulation GUI to modify the wireless or vehicular settings for that node.

- **NodeGUIImpl**:
  - A significant number of changes were made to this library which enhanced the handling of node interaction with the GUI user.
  - A right-click menu was added to support access to the ITA device.
o A number of handlers were added to accept changes made in the GUI and trigger event processing within the wireless simulation which eventually prompted changes in the VISSIM vehicles through the service interface.

o Changes were also made to allow the user to open multiple instances of the Terminal and ITA windows rather than the single instance previously allowed.

o A double-click handler was added to support opening of the ITA tool.

- SimGUI:
  
o An additional menu was added to support interfaces for the ITA device.
  
o The incredibly complex layout logic resulting from code generation were redesigned and simplified to provide faster GUI creation and repaint response.
  
o A number of unused class creations and references were removed to speed GUI creation.

To support the triggering of events between the simulation models and facilitate interaction with vehicle parameters from the ITA GUI device, changes were required to the Node and NodeGUIImpl classes to allow interaction with the event handling components of the ITA device and transportation network proxy elements.

In an effort to tune and stabilize the GUI components of SIDnet significant changes were made to the GUI libraries which removed redundancies and excess references resulting from code generation techniques used in the NetBeans IDE. This allowed the GUI to scale to a significantly larger visualization set and resulted in more rapid GUI processing.

A number of changes were also made across the full code base to migrate the data structures used within Jist, SWANS, and SIDnet to common Java data structures and leverage these previously tuned and proven techniques which offer better integration into iterators and methods helpful to any developer. An additional effort was made to migrate the use of Java class data types to primitive datatypes whenever possible. This step aids the code execution by reducing
conversions, memory utilization, and synchronization issues. Primitive data types within Java are
natively concurrent.

11.2 Java Framework Upgrade

The JiST, SWANS, and SIDnet framework were developed against the Java SDK v1.4.2 (JDK)
and incompatible with the features of later version of the JDK. To ensure compatibility with the
integration components and improve application performance on modern hardware the three
applications were modified and tuned to run on JDK v1.6.0. This effort involved a significant
number of changes to bring the code in line with more modern Java development practices.

11.3 Bytecode Engineering Library Changes

During the upgrade to JDK v1.6.0 a series of issues related to the Bytecode Engineering
Library (BCEL) v5.2, the foundation for the JiST Rewriter, were discovered. The BCEL library
was unable to process variables of Type.CLASS resulting in incomplete JiST-rewritten bytecode.
When run in the JiST environment, the bytecode modifications performed by JiST would be in an
invalid, but executable, form. When process within JiST this led to unexpected behavior. To
adjust for this issue the BCEL library v5.2 ExecutionVisitor class was modified to correctly rewrite
instances of the CLASS type.
12 Cited Literature


(Fan et al., 2006) P. Fan, J. Haran, P. Nelson and J. Dillenburg, Traffic Model for Clustering algorithms in VANETs, IEEE CCNC 2006.


(Morsink et al., 2002) P. Morsink, C. Cseh, O. Gietelink and M. Miglietta, Design of an application for communication-based longitudinal control in the CarTALK project. in IT Solutions for Safety and Security in Intelligent Transport (e-Safety), (2002).


(RTA, 2011) Regional Transportation Authority. www.rtachicago.com


(Santos et al., 2004) R. A. Santos, R. M. Edwards and A. Edwards: Cluster-based location routing algorithm for inter-vehicle communication. VTC Fall (2) 2004: 914-918.


(Sommers et. al., 2008) C. Sommer, Z. Yao, R. German, F. Dressler, On the Need for Bi-Directional Coupling of Road Traffic Microsimulation and Network Simulation, ACM Mobility Models May 26, 2008, Hong Kong SAR, China.


(Zang et al., 2005) Y. Zang, L. Stibor, G. Hiertz and H. Reumerman. Vehicular wireless media network


NAME: James Gabriel Haran


B.S., General Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1997.

HONORS Computational Transportation Science IGERT Associate, University of Illinois at Chicago, 2006-2011

PROFESSIONAL LICENSE Engineer Intern, State of Illinois (No. 061027728)

PROFESSIONAL MEMBERSHIP: Association for Computing Machinery (ACM)
Institute of Electrical and Electronics Engineers (IEEE)

PUBLICATIONS:


