APPROVAL SHEET

Title of Thesis: StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's

Name of Candidate: Sandor Dornbush
Master of Science, 2006

Thesis and Abstract Approved: _____
Dr. Anupam Joshi
Associate Professor
Department of Computer Science and
Electrical Engineering

Date Approved: _____

Curriculum Vitae

Name: Sandor Dornbush.

Permanent Address: 813 E. Fort Ave, Baltimore, MD 21230.

Degree and date to be conferred: Master of Science, May 2006.

Date of Birth: June 10, 1979.

Place of Birth: Denver, Colorado.

Secondary Education: International Baccalauriate George Washington H.S., Denver, Colorado 1997.

Collegiate institutions attended:

University of Maryland, Baltimore County, M.S. Computer Science, 2006.
University of Colorado, Boulder, 2001, B.S. Computer Science, 2001.

Major: Computer Science.

Minor(s):

Professional publications:

Sandor Dornbush, Zary Segall, Kevin Fisher, Kyle, McKay and Alex Prikhodko
XPod a human activity and emotion aware mobile music player
Mobility 2005

Professional positions held:

Research Assistant, CSEE Department, UMBC. (Jun. '05 - Jun. '06).

Graduate Assistant, OIT Department, UMBC. (Aug. '04 - May. '05).

Software Engineer, Convera Inc. (Dec. '03 - Aug. '04).

Research Software Engineer, NeoCore Inc. (Jan. '02 - Mar. '03).

Software Engineer, KBKids.com. (Sep. '99 - Jan. '00).

Software Engineer, University of Colorado, Mechanical Engineering. (Dec. '98 - Jun. '99).

Software Engineer, Wyndemere Inc. (Jan. '98 - May. '98).

ABSTRACT

Title of Thesis:

StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's

Author: Sandor Dornbush Master of Science, 2006

Thesis directed by: Dr. Anupam Joshi, Associate Professor
Department of Computer Science and
Electrical Engineering

Automobile traffic is a major problem in developed societies. We collectively waste huge amounts of time and resources traveling through traffic congestion. Drivers choose the route that they believe will be the fastest; however traffic congestion can significantly change the duration of a trip. Drivers that know the location of areas of slow traffic can choose other, more efficient routes. We could save significant amounts of time if traffic congestion patterns could be effectively discovered and disseminated to the general public. Currently most people use a centralized system that is over 50 years old. This system is fairly effective, but it has significant problems. We propose a system that uses a standard GPS driving aid, augmented with peer-to-peer wireless communication. This system could provide more accurate and complete traffic monitoring than existing systems, and do so at almost no cost to the service provider. StreetSmart has been be evaluated in a simulation. The system uses a combination of clustering and epidemic communication to find and disseminate traffic information. This system is designed to accommodate dynamic traffic patterns. We ensure the privacy of the participating drivers so drivers will be willing to disclose their driving paths. This project could become a very useful system, saving millions of human hours and dollars.

StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's

by

Sandor Dornbush

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2006

DEDICATION

Annie the support of my life. My parents who started and nurtured my academic interests. Dr. David Yager for pushing me back into school, giving me the inspiration to start this project and guidance to finish.

ACKNOWLEDGMENTS

I have deep gratitude for Dr. Joshi's advice and guidance throughout my Master's education. This work would not be possible without the guidance of Dr. Hillol Kargupta and Dr. Tim Finin in the fields of distributed data mining, statistics and machine learning. Dr. Tim Finin and Dr. Zary Segall lead my exploration of the world of mobile and ubiquitous computing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Automobile traffic is a major problem in modern societies. Throughout the world millions of hours and gallons of fuel are wasted everyday by vehicles stuck in traffic. The Texas Traffic Institute estimates that traffic congestion costs the US $68 billion dollars a year [26]. A system that could deliver an accurate map of traffic to drivers in real time could save huge amounts of money. If such a system could be deployed cheaply it would be very profitable and decrease the environmental impact of automobile traffic.

This paper details an investigation of automobile traffic monitoring using mobile peer-to-peer networks. Several companies provide live traffic information [2], and give drivers suggested routes over congested roads. Finding the optimal route through congested roads can be found using use Dijkstra's algorithm [43] weighting the edges using congestion information. The main deficiency of current systems is the lack of accurate source data. This is classic problem of *"Garbage In Garbage Out"* . None of the approaches so far have sufficiently accurate measurements of the traffic conditions to provide a useful service to consumers[35]. Any system that effectively monitored and rerouted drivers would balance the load on roads.

StreetSmart's main contribution is a new method for collecting accurate real time congestion information. There is a great need for accurate real time traffic

information. There have been several centralized approaches to collect live traffic information. However it is cost prohibitive to implement these systems on all roads. In addition centralized systems cannot scale to accommodate all possible vehicles in a major city.

The goal of the project is to create a system that could be implemented with currently available technology. Specifically the system would use the Global Positioning System (GPS) to determine current location and speed. Speed can be calculated from two consecutive GPS readings. The system would use a wireless networking communication medium such as 801.11 A, B or G. Currently many cars come equipped with GPS systems and 802.11 hardware is quite cheap at this point. A commercial version of this system could be sold for around $500. A price many people would pay to avoid traffic.

As vehicles travel through congested roads the traffic device tracks the speed of the vehicle at every location. GPS systems for monitoring automobile speed have already been deployed in a variety of situations [13]. Each device builds a local traffic map from the traffic that the vehicle experiences. As vehicles come close to each other they exchange their speed map with each other. Through these interactions each peer in the system will be able to build a map of expected speed on every road, even those they have not visited.

This paper studies this problem in a simulation. In the simulation cars drive random paths through a grid. The accuracy of the system is measured as difference between the speed map collected by the nodes in the system and the true speed map.

This paper introduces two distributed clustering algorithms. These algorithms are designed to function well in a network that is more often disconnected, than it is connected. This algorithm computes clusters using a epidemic diffusion model. Every time two cars interact with each other they try to share traffic information.

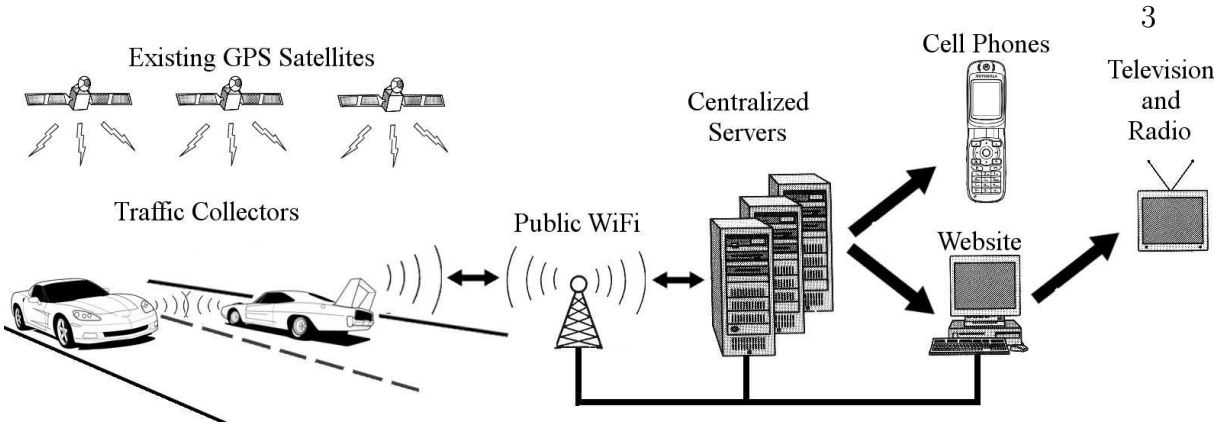


FIG. 1.1. System Overview

The system is tested in a Java simulation. A set of random traffic congestion areas are created. As the cars traverse the map they learn the location and intensity of traffic of the road they are traveling over. That information is exchanged to build an accurate global map.

1.1 System Goals

The primary objective of this system is to use vehicle based GPS devices to monitor road conditions. Other researchers, including some from the Federal Highway Administration have considered using GPS or mobile phones for collecting congestion information [20]. There are a variety of ways to collect that information and use it. This system is designed to use a combination of wireless base stations and a ad-hoc network of mobile devices. The congestion information collected off of the StreetSmart system is valuable as live traffic information and for historical purposes. Drivers want live information to make decisions about an optimal travel route. Government agencies and consumers want historical traffic information to make decisions about long term projects, such as new road construction.

1.2 Outline

Chapter 2 outlines the design goals of the system. Chapter 3 of this paper discusses related work and background information. Chapter 4 contains a description of the algorithms and data structures used. Chapter 5 describes the simulation used to test the algorithm. Chapter 6 contains the results of experiments conducted. In chapter 7 several attacks and defenses are discussed. Chapter 8 contains the conclusion and discussion of future work.

Chapter 2

SYSTEM DESIGN

StreetSmart is designed to be a useful system to drivers. The problems and issues covered address the needs of drivers. This approach leads to a design that is different from similar systems. The solutions presented here are result of careful design choices. This system does not exchange information on every section of road instead it only exchanges summary information on areas of unexpected traffic.

2.1 Need To Say

This paper relies on a principle that we coined as the "Need to Say" principle. This principle is related to the security "Need to Know" principle. The "Need to Say" principle seeks to reduce communication load by only transmitting useful information. Other researchers in the VANET field have focused on data fusion as a way to reduce communication load. We have not ignored that tool, but rather augmented it. A similar approach has been used for perform association rule mining (ARM) [33]. Wolff et al show that it is possible to find association rules in a distributed system with-out communicating all possible frequent item set counts. This principle is an adaptation of Shanon's Principle. We have adapted Shanon's principle to an area where the most likely data items are not exchanged.

2.1.1 Posted Speed Limit

The first application of this principle to this project is observing that one need not communicate any information if the vehicle is traveling at, or above the posted speed limit. Drivers do not need be notified if the road is clear. This is the motivation for representing the traffic as a collection of clusters of slow traffic. It is assumed that the number of clusters of congestion is significantly smaller than the total number of road segments. This simple insight compresses the set of all road segments down to just the deviations from the speed limit.

2.1.2 Expected Speed

It is possible to extend the idea of expected speed beyond the posted speed limit. Traffic congestion has very predictable trends which can be exploited. For example, major commuter routes will be slow during rush hour. If this information is available to all of the nodes of the network then each node only needs to communicate when the recorded speed is outside the variance of expected speed. This would reduce the communication significantly without any difference in information available to the end user.

2.2 System Adoption

This system does not expect that all drivers adopt it. In fact it is designed to work with only a small fraction of total drivers participating in the network. While it might be desirable that all cars use VANETs it is highly unlikely that will happen in the near future. It is far more plausible that drivers from higher economic classes and professional drivers will be the first adopters. The system was tested with a relatively small number of drivers, 50 to 500, in a fairly large space, over 64 square kilometers. This is a marked difference between most other proposed VANETs. Most other work

assume near total adoption rates.

While only a small number of drivers is needed to monitor the road, the higher percentage of participating drivers the better the results. For this system there is a tipping point of adoption. Below the tipping point the results will be too poor to be useful. Above the tipping point the results will accurately reflect traffic conditions. In order to achieve the necessary adoption rates it is imperative to engage corporations with a large number of professional drivers. The adoption of a single company such as FedEx, with a large number of vehicles on the road constantly, would push the adoption above the tipping point.

2.3 Serendipitous Exchange

In the StreetSmart Traffic system each vehicle collects, stores and exchanges all traffic information that is made available to it. The nodes in the network do not differentiate between information that might be useful to the driver and information that will be of little use to the driver. By cooperating nodes will build a more accurate global model of the road network. While information held by one vehicle of may be of no use to that car, it may serendipitously exchange that information to another car that values that information. SEREFE is an example of a system that uses serendipitous information exchange. The SEREFE file system was built to facilitate serendipitous file exchange [4].

A motivation for this method of information exchange is shown in figure 2.1. Car A is stuck in traffic and sends a message to car B. That information is of little value to car B since the congestion is on the other side of the road. However a little down the road B relays the information to car C. With that new information car C leaves the road for a better route.

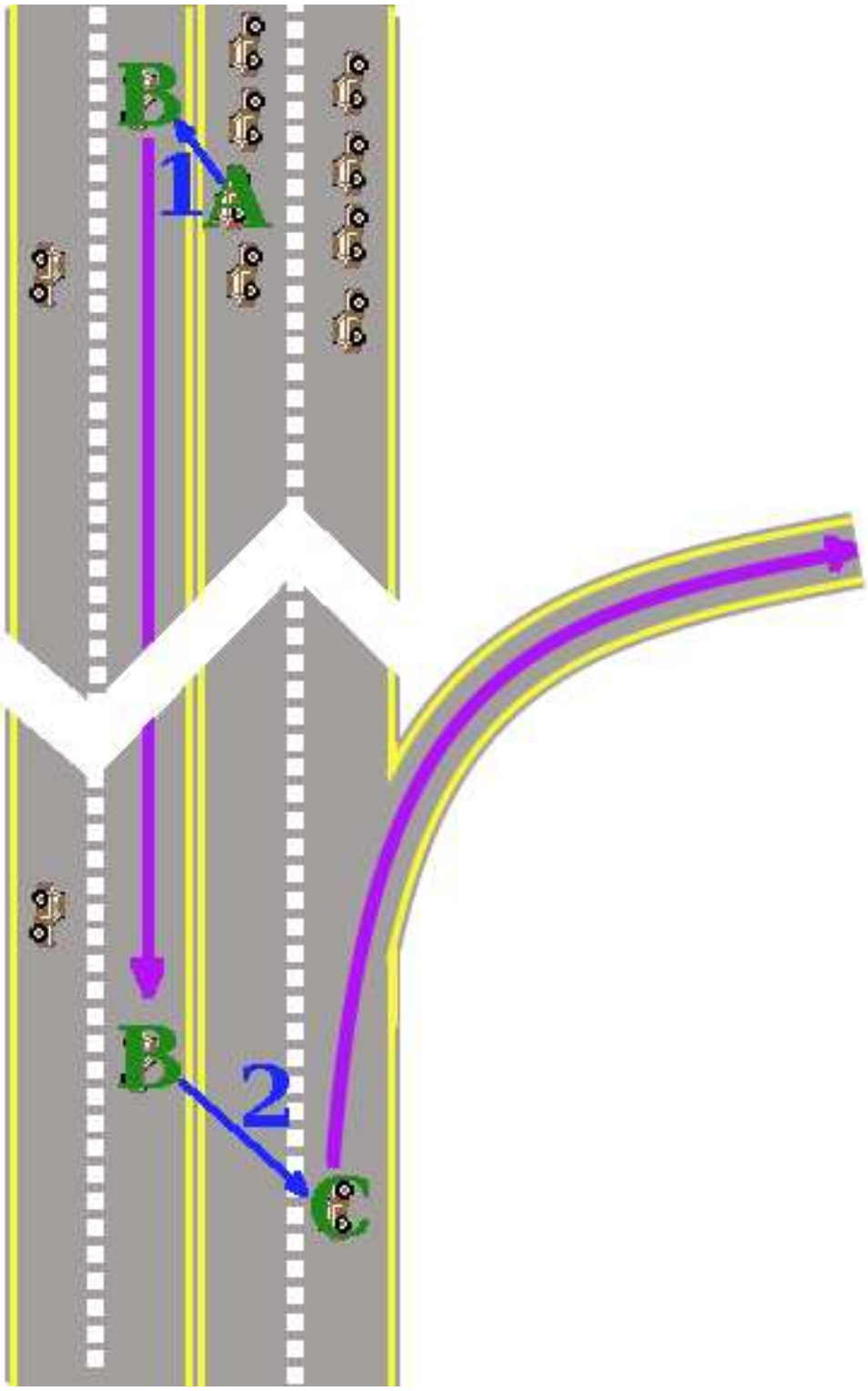


FIG. 2.1. Congestion Information Exchange

2.4 Assumptions

The experiments performed on the StreetSmart Traffic system operate under several significant assumptions. The simulation uses a simplistic model of vehicle traffic and wireless communication. Our simulation provides a sufficient level of detail to experiment with our application. Many of the assumptions, such as a random way point model, are shared by similar work. We have outlined several types of attacks and potential defenses. None of the attacks or defenses have been implemented or tested. One set of defenses try to protect against the introduction of malicious data, however a group of colluding nodes could overcome the defenses. We have outlined a framework to try to protect participating users privacy. The framework will not provide absolute privacy, however it should provide a level of privacy we believe will be acceptable to most users.

Chapter 3

RELATED WORK

StreetSmart builds upon the work of many other projects. This chapter explores relevant work. In each section we show the current state of the art, how we utilize it and improve upon the state of the art. Where relevant we demonstrate differences between our approach and previous approaches to similar problems.

3.1 VANETs

Modern vehicles contain many processing elements. There are individual devices with sensors, processing power, communication and storage. Most of the resources are in dedicated devices such as engine control, mobile phones or mp3 players. Networking together these resources across many cars it is possible to create Vehicle Ad-Hoc Networks(VANETs) [15][3]. Many major car manufactures and leading research institutions are investigating VANETs. VANETs are a small part of Intelligent Transportation Systems(ITS). ITS spans issues from road sensors to networked DUAL-MODE cars [31]. Much of the work in the VANET community focuses on simulation [11][17], multi-hop routing [39][14][9] and entertainment systems such as multi-player games [12] or streaming music [16].

A couple of researchers have studied the problem of using VANETs to discover and disseminate congestion information. Ways of organizing such networks were dis-

cussed in [10]. The authors of [27] discussed ways of extending an existing centralized GPS traffic monitor to use VANETs. TrafficView [28] addresses the issue of estimating road congestion using a network of vehicle based GPS systems. In essence networking car based GPS devices creates a sensor network. Instead of measuring temperature or vibrations, this network is measuring the congestion of the roads. However this sensor network has a few key distinctions from typical sensor networks. These are the key differences:

1. **Power consumption is not an issue.** The power used the computer systems is insignificant to the vehicle.

2. **Networks are very dynamic.** Vehicle traffic can lead to a fast changing sparsely connected network.

3. **Node size and cost are not severely limited.** Vehicle born GPS systems are larger and have more resources than typical sensor network nodes. By comparison to standard sensor network nodes in this network are relatively rich in computational resources.

Previous VANET congestion systems had very limited scope. The TrafficView [28] project focused on a very specific problem, what is congestion on the road directly ahead. It only addresses a 2D road in a single direction. The TrafficView project was able to demonstrate that it is possible to monitor vehicle congestion using a real VANET. A group at Technische Universitt Hamburg-Harburg have developed SOTIS - Self-Organizing Traffic Information System. [40] SOTIS only addresses the problem of detecting traffic in a 2 lane road.

3.2 Clustering

Clustering is an important data mining tool. Clustering is the process of finding data points that are similar to each other by some measure. Clustering is used widely for a variety of purposes from grouping gene sequences with similar function to grouping related news stories [38][19].

StreetSmart clusters together related recordings of unusual speed. StreetSmart uses clustering as a data aggregation technique. Our system tries to build a speed map identical to the actual conditions on the road. However if the expected speed is understood by all participating nodes, then only deviations from the expected value needs to be expressed. Therefore in StreetSmart the cars only express information about areas of unexpected speed. If the speed is not transmitted it is assumed to be close to the expected speed. It is assumed that in the deployed system the GPS device would have a map of the roads along with the posted speed limits. All GPS driving aids have this information today.

3.2.1 Distributed Clustering

Several algorithms have been presented to find clusters in a distributed fashion. For our purposes we wish only to consider the homogenous situation. We will assume that every car collects similar data types. Several papers present ways to perform a bit of data analysis at each node then send the results of that analysis to a central node. A few examples of this ensemble based approach can be found at [21] [25].

The MetaClusters algorithm applies this sort of technique however each node tries to do the analysis of the collected statistics, instead of just the central node. The problem of clustering clusters was addressed in [8].

There are several techniques for calculating K-Means in a P2P fashion. However all of them have relied on a fully connected network. In the algorithm P2P-K-

Means [5] every node must interact with other nodes every iteration. L2 Norm based K-Means [42] improves on this some by only asking each node to talk to its immediate neighbors on most iterations. However if there is an alert; the algorithm requires a global converge cast.

There are several algorithms for distributed clustering that rely on centralized ensembles. These algorithms calculate statistics at each node then send those statistics to a central site. Then the central site calculates the global clusters from the statistics from the other nodes. A major contribution of this paper is to take this ensemble approach to a peer-to-peer environment.

3.3 Epidemic Communication

StreetSmart Traffic deploys a form of communication that is alternately referred to as rumor [32], gossip [23] or epidemic [6] based. This class of communication protocols is based on the practice of saving messages from previous nodes and relaying those messages to other nodes. While these communication protocols do not provide any guarantee on the timeliness of the data, they are often the best possible protocols in disconnected settings. Performing data mining using gossip protocols was used in the Astrolabe system [7].

Much of the work in area of gossip file systems focuses on resolving updates to files. In situations where multiple users can modify the same file this is a non-trivial problem [41]. We do not have this problem because the only node that can update a set of centroids is the node that collected that data. Therefor we can use a data structure similar to a vector clocks [34] to store centroids and resolve updates.

It is possible to prove the accuracy of data mining tasks performed on loosely connected networks [24]. However all of the proofs so far have relied upon conservation of mass. For example in a means calculating scheme so long as no value is lost any

two nodes can average their values and the network will converge towards the correct value. In such a scheme the sum of all values is the same at the beginning of the algorithm as the end of the algorithm. The sum of the values is conserved throughout the algorithm, but at the end the values should be uniformly distributed. However in our experiments there is no conservation of mass. StreetSmart Traffic clusters the stream of road congestion information. In data stream application information is added and removed the entire time. When the nodes start out they have no knowledge of traffic, they discover the traffic as they travel through the map. So it is not feasible to distribute the mass throughout the network.

Chapter 4

ALGORITHMS

This paper presents a distributed clustering algorithms that does not require constant connectivity. The goal of these algorithms is to find clusters on a mobile ad hoc network where disconnected components are more common than connected components. There are many obstacles to creating a fully connected traffic monitoring system. Vehicle move quickly, which limits connection time. Vehicles moving in opposite directions have a very short window of time to exchange information. Initially any commercial system would take a bit of time to deploy. So in many situations there may be a large number of cars but only a few who are participating in the P2P traffic monitoring. For all of these reasons we cannot rely on a fully connected network.

The motivation of the algorithm is for each node to keep a compressed version of every other nodes traffic information. This information is expressed as summary statistics of the clusters. Nodes exchange summary statistics using epidemic communication. Every participating node calculate higher level clusters from the summary statics gathered locally and from the network.

4.1 Data Representation

The naive choice of data representation for geostatics data is latitude and longitude coordinates. Early experiments used this data model. That model worked well with the Meta KMeans algorithm. The distance could always be calculated using the L2Norm of the latitude and longitude. However that model presented significant problems. If one simply uses latitude and longitude coordinates it is possible that clusters of data appear to be inside city blocks or on the wrong side of the road. To overcome this deficiency congestion data is represented by a road identifier and an offset on that road, similar to a mile marker. Different sides of a road have different identifers. The similarity of two data points can be expressed by this function:

$$\textsc{difference}(d1, d2) = \begin{cases} \textsc{l2Norm}((d1.distance, d1.speed), \\ \qquad (d2.distance, d2.speed)) & d1.roadId = d2.roadId \\ 0 & Otherwise \end{cases}$$

4.2 Notation

4.2.1 General Notation

For notation n is the total number of nodes in the system. k is the number of clusters calculated. id is the identification number for the node in question.

4.2.2 Local Data Notation

As vehicles travel each vehicle collects traffic information. This information is derived from that vehicles interaction with the outside world. Each node collects a set of traffic history values denoted as η_i. Each η_i entry has the following attributes, speed, time and either RoadId and distance or X and Y.

Each node in the network clusters the η_i values to create local clusters. φ_i is the i^{th} cluster of the current node which is a point in on a map. ω_i is number of data points that contributed to φ_i, for the i^{th} cluster of the current node.

4.2.3 Exchange Notation

Each node will maintain an array of centroid information about other nodes. $\alpha_{i,j}$ is the j^{th} cluster of the i^{th} node as recorded by the current node. $\beta_{i,j}$ is the number of data points in the j^{th} cluster of the i^{th} node as recorded by the current node. μ_i is the time that node i calculated the clusters as recorded in $\alpha_{i,j}$ at the current node.

4.2.4 MetaCluster Notation

Each node will calculate MetaClusters from the centroids gathered from other nodes. Φ_i is the i^{th} MetaCluster at the current node. Ω_i is number of data points that contributed to Φ_i, for the i^{th} cluster.

4.2.5 Cluster Attributes

Each cluster either a φ_i, $\alpha_{i,j}$ or Φ_i have the following attributes:

Mean X X Variance Mean Y Y Variance	or	Mean Distance Distance Variance Road Id
Mean Speed Speed Variance Mean Age Mean Variance		

Table 4.1. Cluster Attributes

4.3 Information Decay

Traffic information is transient, congestion changes over time. In this system the contribution of each low level cluster is weighted by a decay function. This way the influence of each cluster decreases over time. After the weight is bellow a threshold level that low level cluster is discarded. The decay function is defined below:

$$decay(age) = \frac{1}{1 + e^{decayRate(age - waitTime)}} \tag{4.1}$$

The decay function is a sigmoid curve that varies from 1 to 0. The waitTime parameter controls how long after a piece of information is recorded before the influence of that information is reduced. The decayRate parameter controls the rate at which the influence decays.

4.4 Traffic Monitoring

Every iteration each node samples the traffic map and calculate the local centroids. Local clustering is only executed if the view of the traffic stream has changed significantly, by the addition of new information, or removal of outdated information. If the local clusters changed then the exchange data for this particular node must be updated. Each node executes the following code in every iteration:

```
calculateLocalCluster()
1  ▷ Sample data.
2  if sample contributed new data
3     then
4          ▷ calculate new ω and φ
5          α_{id,i} ← ω for all i
6          β_{id,i} ← φ for all i
7          μ_{id} ← now()
```

During each iteration the nodes attempt to send their exchange information to other nodes. Each node sends their view of the world to other nodes. Upon receiving a message each node checks to see if the data that it has received is newer than the information already stored at that node. If the message contains new information the new information is stored in the local $\alpha_{i,j}$, $\beta_{i,j}$ and μ_i variables. This is an epidemic form of communication.

```
reciveMessage()
1  for i ← 0 to length[message.μ]
2       do
3          if local.μ_i < message.μ_i
4             then local.α_{i,j} ← message.α_{i,j} for all j
5                  local.β_{i,j} ← message.β_{i,j} for all j
6                  local.μ_i ← message.μ_i
```

After sampling the data, and exchanging messages each node calculates new MetaClusters. Each node merges the snap shot of other nodes centroid data with its local centroids. The collection of $\alpha_{i,j}$ values are assigned to the closest Φ_i. The new Φ_i values are calculated based on a average of the $\alpha_{i,j}$ values weighted by the $\beta_{i,j}$

values of all the data points that are assigned to that Φ_i. We experimented with two different methods of calculating the MetaClusters.

4.4.1 K Means

The first method of calculating MetaClusters we experiment with was an adaptation of K-Means to an peer-to-peer environment. Each cluster is built as a weighted combination of the source data.

CALCULATEMETAKMEANS()

1 **repeat**
2 **for** each $\alpha_{i,j}$
3 **do**
4 $\triangleright$ assign $\alpha_{i,j}$ to the closest Φ_k.
5 **for** $k \leftarrow 0$ **to** *length*$[\Phi]$
6 **do**
7 $\triangleright$ Let ζ be the set of tuples $\{i, j\}$ where $\alpha_{i,j}$ is assigned to Φ_k.
8 $\Omega_k \leftarrow \sum_{i,j} \beta_{i,j}\text{DECAY}(\mu_i)$ where $\{i, j\}\epsilon\zeta$
9 $\Phi_k \leftarrow \dfrac{\sum_{i,j} \alpha_{i,j}\beta_{i,j}\text{DECAY}(\mu_i) \text{ where } \{i, j\}\epsilon\zeta}{\Omega_k}$
10 **until** Φ_i does not change for all i

Optimization Techniques There are several problems with the standard K-Means algorithm. To achieve acceptable results it was necessary to implement optimizations that avoid those problems. In this traffic monitoring simulation each node starts with no knowledge of the traffic, then learns as it goes along. As such there are two main problems. First all centroids start empty so empty clusters must be han-

dled. Also the K-Means gets stuck in local minima that are discovered early instead of finding the global minima.

To address the first problem after executing the local K-Means each node checks for empty centroids. If there are empty centroids the algorithm will move one empty centroid to the data element farthest from the established centroids.

To address the second problem periodically the nodes will generate several new sets of random centroids, then choose the best set of centroids. The best set of centroids is the set with the lowest mean squared error.

4.4.2 Hierarchical Agglomerative

To avoid some of the difficulties of K-Means we created a distributed hierarchical agglomerative clustering system. This system does not need to know the number of clusters a-priori. It is possible to use the custom similarity measure defined in 4.1. This system is not affected by initial conditions. Since this system is built on hierarchies it will be simple to extend the scheme to hierarchies of nodes in the network.

Hierarchical clustering is the process of combining the most similar elements. At the start of the process all of the elements are raw data items, however once items are combined, those clusters are also considered elements for combining. Typical hierarchical clustering continues until all nodes have been collected into one cluster. Instead of the traditional approach to traditional approach we choose to keep some sets of data points separate. Any data points that shared no similarity cannot be combined. This simple rule eliminates the problem of combining information from different roads.

The second rule is to not cluster together data points that are significantly different. This is implemented in different ways at the local level and at the network

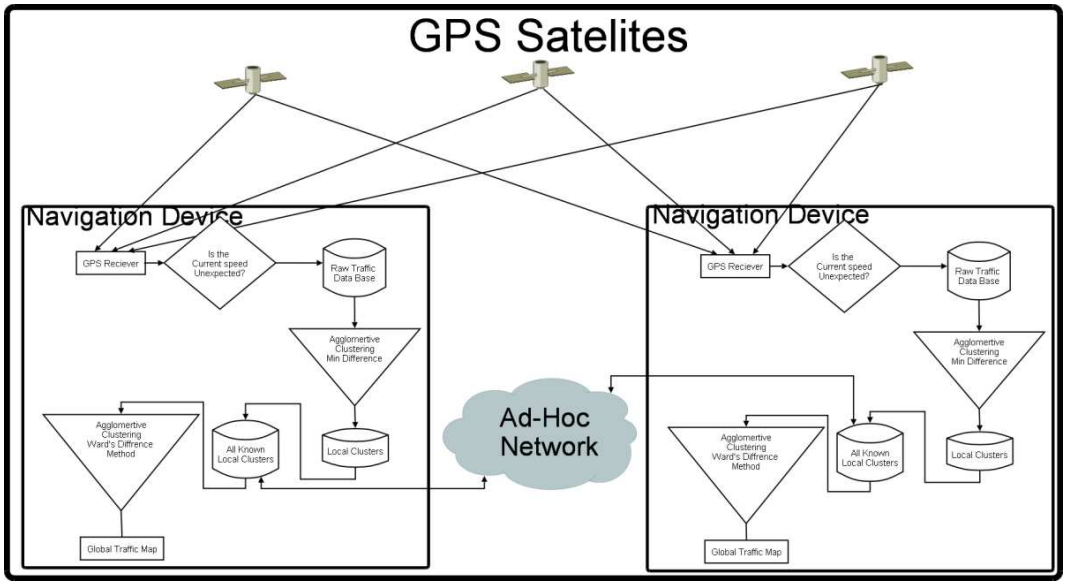


FIG. 4.1. Hierarchical Agglomerative Clustering Overview

level. This prevents significantly distinct areas of congestion being combined in error.

To adapt agglomerative clustering to a distributed setting we used a combination of two proximity measures. When deciding whether to combine two clusters one must decide their proximity. When clustering local traffic information we used the MIN proximity measure. This measure defines the proximity of two clusters and the proximity of their two closest elements. If the absolute difference between the two closest data points is greater than an threshold the clusters will not be combined.

It is unreasonable to send all traffic information from one node to another. For this reason we used Ward's method of proximity to measure proximity of distributed clusters. Under Ward's method the proximity of two clusters is defined to be the proximity of the centroids of the two clusters. Therefor instead of combining raw data elements, only summary statics are needed. If the centroids fall within a set number of standard deviations of each other then the centroids will be combined. If they are outside the factor of standard deviations then the centroids will not be

combined.

4.5 Analysis

Most of this paper focuses on simulated results on the StreetSmart system. This section contains theoretical review of this algorithm.

4.5.1 Communication Cost

An optimal local algorithm will scale infinitely with the number of nodes. This algorithm is not a totally local algorithm. However the size of the messages can be fixed using a heuristic. A heuristic is used to send the most important messages. Naively the size of each message is O(nk) where n is the number of nodes.

The heuristic fixes the size of the messages to a constant c. Such a scheme would limit the size of messages to O(c) = O(1) or infinitely scalable. The key to an effective implementation of this is choosing the most important c centroids and membership counts to send. Ideally one would like to send only the centroids that would contribute the most information to the solution. Some of the aspects to be considered would be the age, the quality information, the quantity and the unexpectedness of the information. One would want to send the most recent centroids. The centroids with the highest membership counts should be sent. An optimal solution would probably be a heuristic based on a weighted combination of these factors. For our simulation we chose to consider only the age and membership counts. Each node chooses to send only the nodes with the highest heuristic value that can fit within a fixed message size. The heuristic is $h(\mu_i, \alpha_{i,j}, \beta_{i,j}) = \sum_j \beta_{i,j}\alpha_{i,j}.age$.

Another way to approach this problem is to combine the clusters then send a summary. An example of a hierarchical approach such as this is presented in [22]. However if no formal tree is imposed there is a danger of over reporting some

MapId	Integer	4
Offset Mean and Variance	Double	16
Speed Mean and Variance	Double	16
Count	Integer	4
Age	Long	8
Total		**48**

Table 4.2. Cluster Size

items. One of the goals of this system was to accommodate hierarchical clustering. The agglomerative clustering data model can easily be extended to accommodate hierarchical clustering.

Message size can be broken down by the constitute parts. Table 4.5.1 details the size of a single cluster in the road ID data model. Each of the values in the table is the number of bytes used to represent that field.

As an example, assume that the expected number of traffic clusters any one car encounters is 10. Then the expected size of one local traffic map is only 480 bytes. If there are 1000 participating nodes all of the local traffic maps would have a combined size of 469 KB.

In this simple example the size of the largest messages is 469 KB. These messages are not tiny, however they are very reasonable size for transmission over 802.11 networks. We will also show that typically much fewer clusters are needed for good results.

Chapter 5

SIMULATION

StreetSmart Traffic was evaluated in simulation. While the simulation is fairly basic it models the environment to a sufficient level to evaluate the system. It would be desirable to use a more sophisticated simulation, however no one suitable simulation was found. Different simulators model different aspects of automobile traffic and communication. Some simulators such as STRAW[11] have combined vehicle simulation and MANET simulation to create a VANET simulator. However even this does not accurately model traffic congestion. At a future point it would desirable to combine a system like STRAW with a congestion simulator such as DYNAMIT[1]. Both communication simulation and traffic congestion are very computationally expensive. Any system that combined those two would be very slow. With the additional load of the StreetSmart Traffic distributed data mining the resulting system would take an extremely long time to run.

5.1 Communication

The StreetSmart simulator attempts to model basic digital broadcasting. In the simulator only every node tries to broadcast its message every second. If more than one node tries to acquire the medium within broadcast distance of other nodes; one will succeed the rest will fail. The successful node will broadcast its message, and not

try to acquire the medium in the next second.

The broadcast distance and allowable message size are designed to approximate the performance of 802.11 G. For most simulations the broadcast distance is set to 500 meters. For most simulations the message size is fixed 1000 centroids.

5.2 Vehicles

The simulation is a Manhattan grid of roads. The roads are designed to approximate highways. The speed limit is set to be 30 meters per second, or 108 km per hour. The vehicles perform a goal oriented random walk through the map. Each vehicle is given a random destination. The vehicles will drive to that destination. Upon reaching their destination, they are given a new destination. In this way the vehicles explore the environment, in a more realistic manner than a purely random walk.

As vehicles travel through the simulation their speed is determined by a set of artificial traffic disruptions. These disruptions are randomly placed through the map. The disruptions are defined by kernel functions. We experimented with Gaussian and quadratic kernels.

The both kernels use the distance function $d_i(r,t)$. Where i is the disruption in question, r is the road distance in question and t is the current time. Each kernel has a center in time and space. The intensity of the kernel is greatest at that time and place. For each kernel the center on the road d_i. The center in time is t_i. Let w_i denote the width of the kernel. The value k is a factor that controls the width of the kernel. The distance function is defined as:

$$d_i(r,t) = \frac{\sqrt{\frac{(d-d_i)^2}{w_i}+(t-t_i)^2}}{k}$$

The Gaussian kernel is defined by the formula:

$$gaussian_i(r,t) = e^{-d_i(r,t)^2}$$

The quadratic kernel is defined by the formula:

$quadratic_i(r,t) = max(1 - d_i(r,t)^2, 0)$

We found that the quadratic kernel seemed to model traffic congestion more accurately than the gaussian kernel. All of the results presented in the results section use the quadratic kernel.

For any point on the map the total delay can be calculated by taking the sum of all of the delays for that road. Let δ be the set of kernel delays that match the road in question. Let $kernel_i(r,t)$ express one of the two kernel functions defined above. Then the total delay will be:

$\sum_i^{\delta} kernel_i(r,t)$

Throughout the simulation the traffic changes. After a set interval the oldest disruption is replaced with a new random disruption. We set the lifetime of a disruption to be two hours. Figure 5.1 demonstrates how one kernel disruption changes in time and space. The deep blue color indicates an intense slow down.

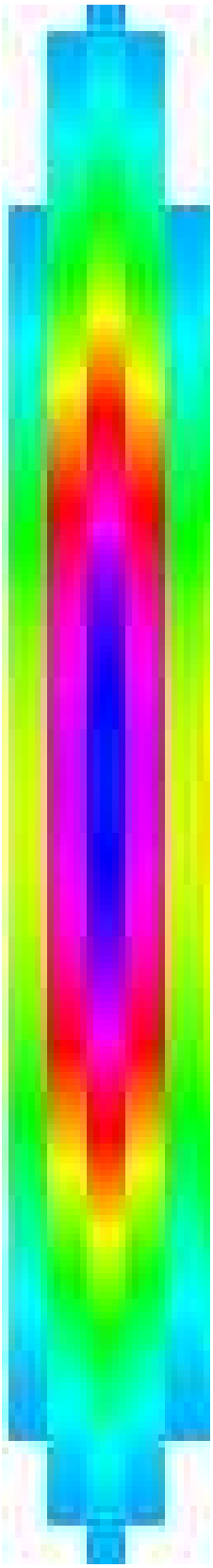


FIG. 5.1. Kernel Disruption in time and space

Chapter 6

RESULTS

We defined the accuracy of the StreetSmart Traffic system as the difference at every point along every road between the true traffic and estimated traffic. At uniform intervals the speed is sampled from the true speed map and the estimated speed map, τ and ϵ respectively. Let $\delta = \tau - \epsilon$. If the expected speed is greater than the true speed, $\delta < 0$, then there is an under estimate of the traffic congestion. If the expected speed is less than the true speed, $\delta > 0$, then there is an over estimate of the traffic congestion. The figure 6.1 illustrates this idea. In this figure there are two kernels that estimate one true kernel.

We have reported the absolute over and under estimations as well as the mean squared error. The mean squared error emphasizes gross errors. The accuracy measures reported in this section are the averages of the above mentioned measures across all nodes in the network. The results are also the averaged over several simulations. In most charts the mean result is labeled. On each side of the mean is the difference by one standard deviation.

The figures 6.2(a) and 6.2(b) use a black body color gradient to display the speed on roads in the simulation. This color gradient illustrated in Figure 6.2(c) was chosen because it has been shown that people accurately perceive changes in the gradient. The intensity of color increases monotonically with the increase in speed disruption.

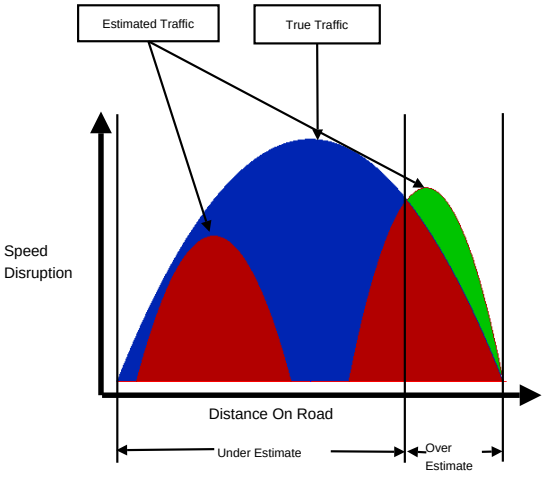
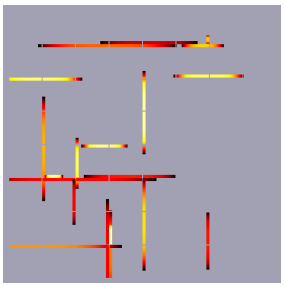


FIG. 6.1. Error Calculation

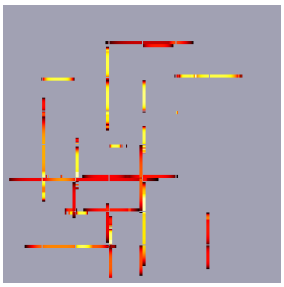
Figure 6.2(d) shows the difference between 6.2(a) and 6.2(b). The difference diagram uses a color gradient illustrated in Figure 6.2(e). These diagrams are the result of a simulation of 500 cars in an 80km x 80km manhattan grid. Figure 6.2(f) shows the mean squared error for the same simulation. Figure 6.2(g) shows the MSE error of the average participating node normalized by the error of knowing nothing, or naive error.

6.1 Data Model

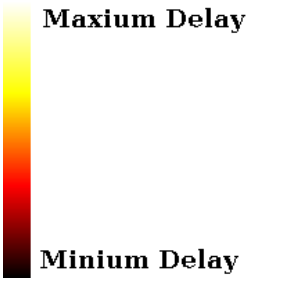
The initial data model used euclidian distance and latitude and longitude way points. This model has significant deficiencies. If two clusters are combined the discovered cluster could be in the middle of a city block, or on the wrong side of the road. When the data is represented by an road id and an offset that sort of confusion is impossible. The logical road location measure also reduces the cost of the cluster calculation by eliminating the possibility of clustering some nodes.



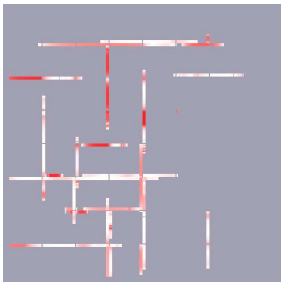
(a) The speed imposed on the vehicles by the simulation.



(b) The speed estimated by the basestation.

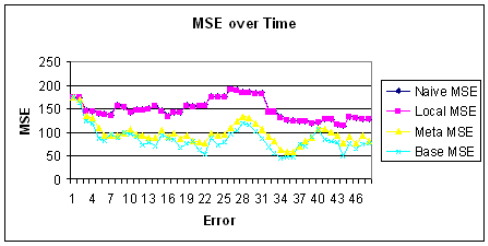


(c) The color gradient used to illustrate the true and estimated speed.

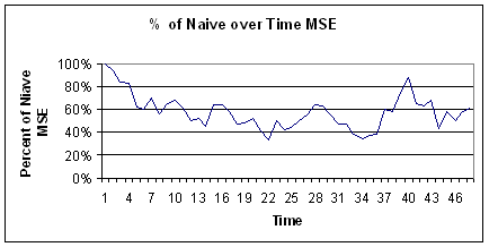


(d) The difference between the speed imposed on the vehicles by the simulation and the estimated speed.

(e) The color gradient used to illustrate the difference in speed.



(f) Means squared error.



(g) Means squared error of the system as a percentage of the naive MSE.

FIG. 6.2. Various results and visualizations of a typical simulation of 500 vehicles.

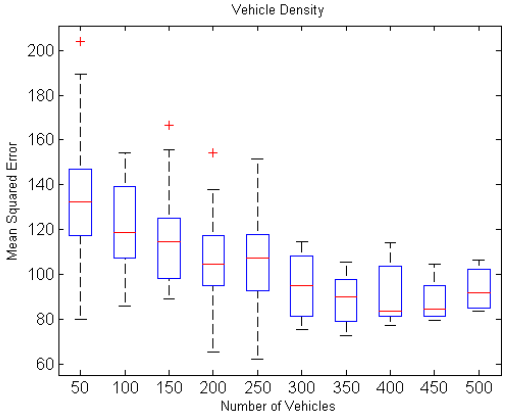
6.2 Cluster Algorithm

K-Means performed reasonably well in the StreetSmart Traffic system. The main problem with K-Means is how to choose the value of K. K-Means requires the value of K to set apriori. This is possible in simulation but not realistic in real life. K-Means did perform well with updates because new updates often fall close to previously found clusters.

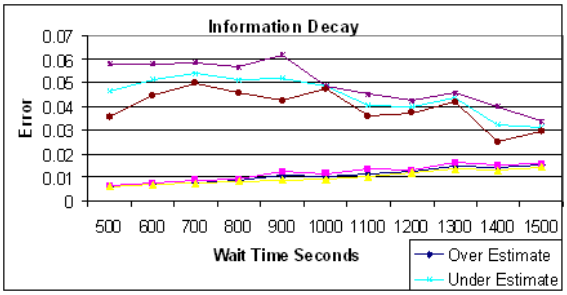
The agglomerative hierarchical cluster avoided the problem of specifying the K value. This algorithm did not report erroneous empty clusters and are not sensitive to starting conditions. This algorithm is more expensive to calculate than K-Means. Every time there is an update the entire hierarchy is recalculated. This process can be optimized with proper data structures. The cost of hierarchical agglomerative clustering is well worth the improved results.

6.3 Density of Vehicles

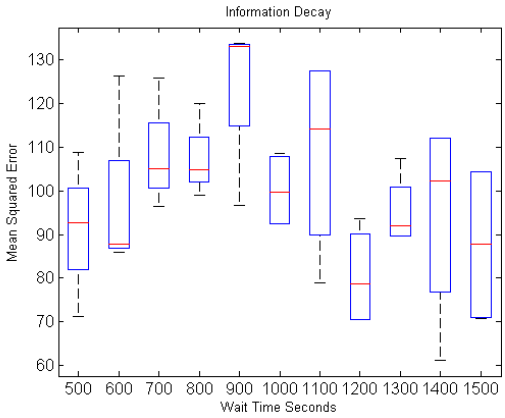
The number of connections between nodes has a strong effect on the accuracy of the StreetSmart system. Several experiments were performed to test this relationship. The first set of these experiments were conducted with different numbers of participating vehicles from 50 to 500, the results are presented in figure **??**. As more vehicles participate in the network the false negative error at each node decreases. This is because more vehicles are sampling the network, and that information is being spread farther and quicker. The under estimate error increases as more vehicles participate in the network. The over estimate increases because the congestion information is spread to more vehicles and will remain in the network at more nodes after the data is invalid. Figure 6.3(a) shows that the total error decreases with the number of participating vehicles.



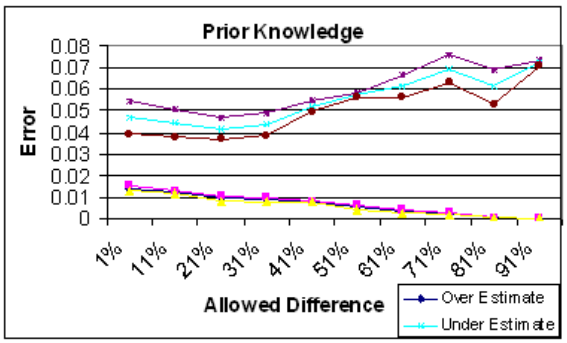
(a) MSE as a function of the number of vehicles



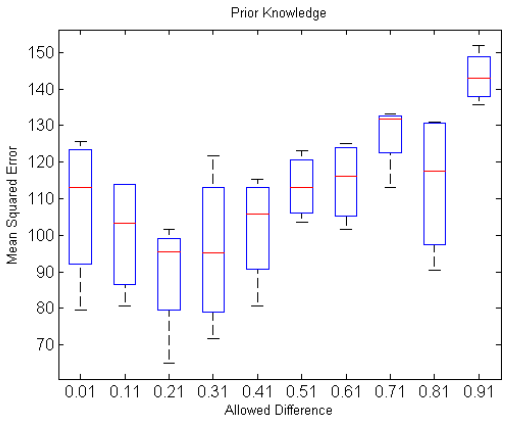
(b) Accuracy as a function of information decay parameters



(c) MSE as a function of information decay parameters



(d) Accuracy as a function of the use of known priors.



(e) The affect on the MSE of the use of known priors.

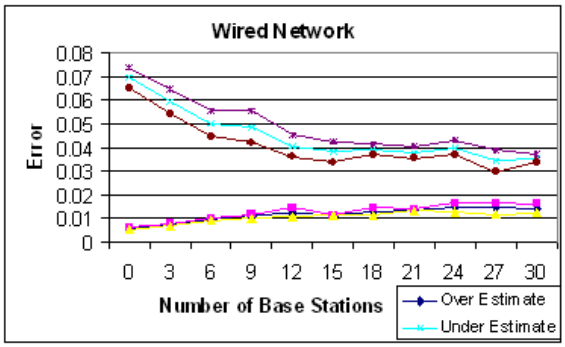
FIG. 6.3. Results from experiments with StreetSmart in simulation.

6.4 Information Decay

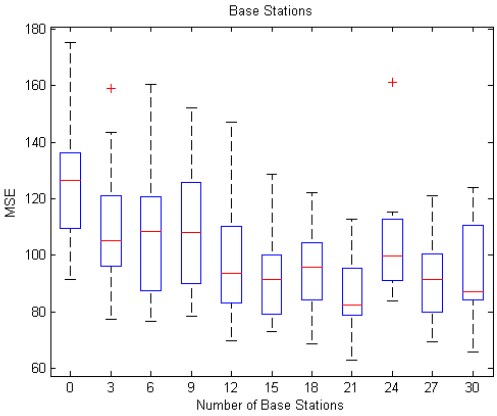
A series of experiments were performed to find optimal values for the decayRate and waitTime in the information decay function. In the simulation the areas of congestion are generated as quadratic polynomial kernels in time and space. The areas of congestion start off with a covering a small amount of a road. As time progress the intensity and size of the congestion increases to a peak, then decreases back to the starting state, before disappearing altogether. The behavior of these kernels are controlled by a couple of random variables with predetermined mean and variance. The optimal values for decayRate and waitTime are directly related to the behavior of the kernels. In practice the variables governing real traffic congestion are not known. These experiments will need to be reevaluated on a real network, or a more realistic model of automobile traffic such as [1]. Figures 6.3(b) and 6.3(c) shows the results of using different wait times. The overall error does not seem to change significantly with different wait times. Future versions will use a more sophisticated model of traffic decay. In such a model traffic information will be retained until a vehicle reports a change in the traffic congestion.

6.5 Influence of Priors

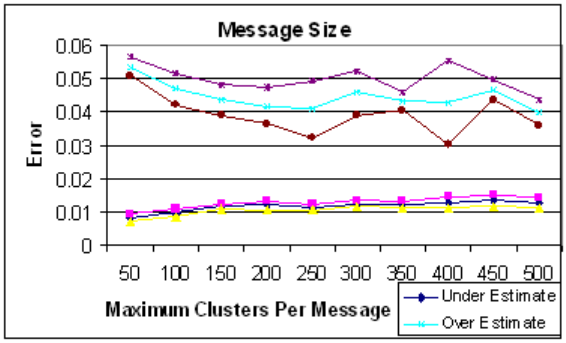
We performed a series of experiments to see the effect prior knowledge has on the accuracy of the systems. In these experiments traffic was only recorded if the speed was different than a fixed percent less than the expected speed. The results of the experiment are presented in figure 6.3(d) and 6.3(e). These experiments show that the accuracy decreases if too much of the speed deviations are ignored. It is clear to see that not all speed information needs to be monitored. It is safe to ignore low intensity traffic, the system will still accurately identify the worst areas of traffic.



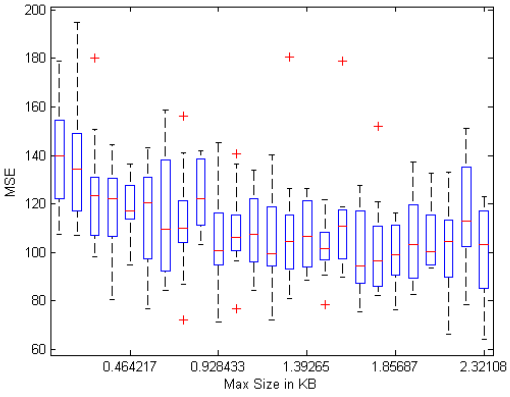
(a) Accuracy as a function of the number of wired sinks



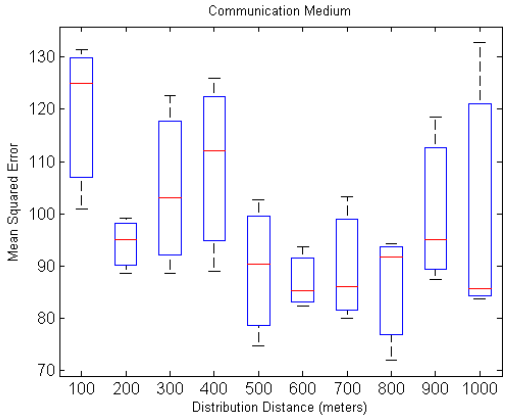
(b) The affect on the MSE of the number of wired sinks



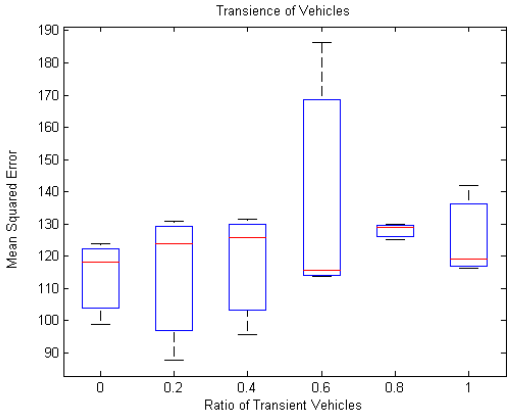
(c) The affect on the system accuracy of the maximum number of clusters allowed in each message.



(d) The affect on the MSE of the maximum number of clusters allowed in each message.



(e) Accuracy as a function of the transmission radius.



(f) Accuracy as a function of the type of vehicles

FIG. 6.4. Results from experiments with StreetSmart in simulation.

6.6 Wired Sinks

Wired network based sinks can increase the performance of the overall network. The wired sinks can take advantage of the reliable nature of wire based networks. The sinks have the advantage that they have no movement on their own. The relative speed of the sink to a vehicle is just the vehicles speed.

If the communication medium is a standard such as 802.11 B or G then any public base station could be a sink. In such a system a person who stopped at a Starbucks in the morning could get the morning traffic map from the Starbucks wireless base station.

These experiments show that the reliable fast nature of wired network improve the performance of the overall system. The results of these experiments are presented in figure 6.4(a) and 6.4(b). As the number of base stations increase the performance of the system increases. The wired sinks behave as connectors in the network. Malcom Gladwell [18] illustrated that a few well connected individuals can greatly increase the spread of an epidemic. The base stations emulate the people connectors that Gladwell identified; both have links to many geographically distinct communities. The addition of a single wired sink can have much more of an effect on the network than the addition of one new vehicle. Notice the relatively small number of open base stations, 0-30. In a typical city the number of networks that would be available would likely number in the thousands. Experiments with War-Driving often turn up thousands of 802.11 networks in metropolitan areas [29].

6.7 Transmission Thresholding

To constrain the size of messages, only the newest and most significant clusters are exchanged. We experimented with several different maximum message sizes, the results of these experiments are shown in figures 6.4(c) and 6.4(d). To our surprise

the performance did not degrade dramatically if only small number of clusters were exchanged. It appears that our simple heuristic is effective at choosing the most relevant clusters to exchange.

6.8 Transmission Radius

Several tests were performed with different transmission radii. The goal was to determine what size is needed to ensure that a high quality map can be built, the results of these experiments are show in figure 6.4(e). The transmission radius will affect the number of interactions that a car can have. Our results do not show a strong influence on the system error from the transmission radius. This is probably an artifact of the simplistic communication simulation. A more robust simulation would capture the true overhead of a wireless network.

For two vehicles to communicate they must remain within transmission distance of each other long enough for both sides to send their broadcasts. For two vehicles traveling in the same direction the distance need not be very large. They will have a small relative speed, and hence a large window of time to transmit information. For these types of interactions the distance may only need to be a couple of times the size of a car. Let s_1 and s_2 be the speed for the two cars. Let be t the time needed for two cars to exchange a set of messages. Let r be the transmission radius of the two cars. The value for t for this situation is constrained to be $t \leq \frac{r}{|s_1 - s_2|}$.

If two vehicles traveling in opposite directions wish to communicate they only have a small window of transmission. The two vehicles will have a large relative speed and small window of time to transmit their information. However that information is often the most important information. Cars traveling in the same direction will have information about the same portion of the map. Cars traveling from either direction can contribute information about opposite sides of the map. More importantly drivers

typically dont care what the traffic is like behind them; they want to know what is ahead of them. The value for t for this situation is constrained to be $t \leq \frac{r}{s_1+s_2}$.

If a vehicle and a base station wish to communicate they have twice as long as the vehicles traveling in opposite directions. The relative speed for these two nodes is just the speed of the vehicle, s. The value for t for this situation is constrained to be $t \leq \frac{r}{s}$.

The typical Bluetooth transmission distance is 10 to 42 meters. Two cars traveling in opposite directions will only be within transmissions distance for just 0.18 second. That is all the time the vehicles would have to perform all communication, including discovery. This is not feasible.

In an outdoor setting commercial class 802.11b can have transmission distances as great as 500 meters [30]. At such distance the transmission rate falls. Cars would have 9 seconds to communicate at that distance. This application may need a better transmission medium because it performs rather poorly at these small distances.

It is possible to boost the transmission distance of 802.11 to great distances using direction antennas. While it would not be desirable to discard omnidirectional antennas, it might be useful to use bi-directional antennas. Since roads tend to be fairly straight and cars tend to point along the axis of the road an antenna focused along the axis of the car would be focused on other cars on that same road. This technique could significantly improve the performance of the system. It is important to not loose the omnidirectional antenna in favor of the bi-directional antenna. Two cars traveling side by side would be communicating perpendicularly to the road and the car axis. However side by side vehicles would be close for relatively long periods of time and would not need extended coverage.

6.9 Vehicle Type

There are two types of vehicles in this simulation. One set of vehicle behave as a taxi or bus would; they never stop driving. These vehicles dont forget the traffic information that they have learned so far. It was thought that this would help the network monitor congestion much more effectively. The other set of vehicles model the behavior of private vehicles. Typical private drivers turn off their car when they reach their destination. For example commuting to and from work. To simulate this behavior when a vehicle reaches its destination it discards all known congestion information, and chooses a new id. A new id is needed so that the data collected by the old car can still echo around the network.

A series of experiments were performed to determine the influence of the ratio of commercial to consumer cars on the performance of the entire network, the results are presented in figure 6.4(f). The x-axis the ratio of total vehicles that forget all information when they reach their destination. Several different ratios are tested with higher performance coming from the tests with slightly higher numbers of commercial vehicles. The type of vehicle does not have a significant impact on the system.

Chapter 7

ATTACKS

In peer to peer systems it is possible that some entities will try to exploit the open nature of the system. For this system there are two main types of attacks. The first is to introduce inaccurate data. The second is to try to discover where a person has traveled. We have outlined several approaches to protect the system. It is not our goal to make it impossible for for a malicious node to disrupt the system or learn private information. Rather we have outlined several techniques to make it difficult for an average node to disrupt the system or comprise privacy. At the current point none of these defenses have been implemented.

7.1 Malicious Data

A malicious node in the network could introduce inaccurate data. For example a node could report traffic that does not exist. There are several ways of looking at this attack. The first is motivation. Who would gain from disrupting the system? Few people would actually gain from introducing malicious data. Automobile traffic is almost universally disliked. However that is not sufficient to dismiss the attack. Many hackers attack for the joy of proving it is possible. One simple way to overcome this attack is sufficient penetration of honest nodes. If the ratio of malicious nodes is small their influence will not be statistically significant.

The most elegant way to defeat this attack is through data validation. TrafficView took this approach. One can keep suspicious data around but only use it if it is a likely data point or it is corroborated by another node. This would involve inspecting every new data point to determine if it is an outlier to the current model [28]. Any data point that is an outlier to all other data would not used in cluster calculation or relayed to other nodes in the network. A group of colluding nodes could still introduce malicious data. A dedicated group of attackers could mount a distributed denial of service against StreetSmart Traffic.

7.2 Privacy

Driver's value their privacy very highly. American's travel mostly by car; revealing their driving patterns will reveal intimate details of their life. The slow adoption of E-Z Pass and similar devices, can be attributed to the lack of privacy in those systems. To ensure that this system is adopted we must ensure that the user's privacy is properly secured.

The first step to hiding a drivers tracks is that this system only reports summary statistics about areas where unexpected congestion was encountered. This abstraction will hide fine grained driving patterns.

7.2.1 Direct Look Up

The first attack on the system can be characterized as a direct look up. If each car uses a known identifier such as license plate number or VIN as the key into the vector clock it is easy to find a user's history. A simple fix to this problem is to use a private identifier for every car. Where attackers cannot directly map the ID to a driver.

7.2.2 Small Set

The second attack can be characterized as a small set attack. If a car broadcasts only one set of traffic data it is possible that the car did in fact visit those roads. However since nodes rebroadcast other nodes data an attacker cannot be certain that the vehicle did visit roads that it broadcasts. The small set attack can be protected against by if the data is K-Anonymous [37]. It is not feasible for this system to provide a result that guarantees K-Anonymity. Our approach is very similar to that taken in the K-Similar system [36]. The clustering used by StreetSmart Traffic will hide the exact details of a drivers behavior. However we would not alter data to provide K diverse points. Altering data could have disastrous effects on the accuracy of the system. As a result of the gossip system a node will broadcast information about congestion that it has encountered or that another vehicle encountered and was relayed to it. It is possible that a node broadcasts a set of $\alpha_{i,j}$, $\beta_{i,j}$, and μ_i, but that car did not collect any of the data. Any potential attacker would only know that one of the sets of clusters **might** be from the broadcasting car.

A way to ensure better privacy is for nodes to only listen until they have collected at least K-diverse data points. As is common with K-Anonymity systems, this creates a boot strapping problem. If every node waits until it gets broadcasts from others, the system will not start. Every node will be waiting for broadcasts from other nodes. While it is possible to provide our approximation of K-Anonymity it is not desirable that all nodes use this privacy model. For the network to perform well it is desirable that some of the nodes assume no privacy. These nodes will start the system. Professional vehicles such as delivery trucks are ideal candidates for privacy free nodes. While vehicles are waiting to collect K diverse data points they can still rebroadcast clusters collected from other vehicles. However due to the non-stationary nature of the data this may not provide much privacy from a determined attacker.

7.2.3 Non-stationary Data Points

The final attack is also a small set attack. This attack takes advantage of the non-stationary nature of the data. As a vehicle moves through traffic its estimate of that cluster of traffic will change with newly acquired data. For this attack the attacker needs constant visibility to the targets broadcasts. There are two ways for an attacker attempt to achieve constant visibility. The first method is if the attacker follows the target vehicle in a second vehicle. The second attack method is when the attacker maintains a wireless mesh network. The attacker would be able to record most incoming messages that the target car receives and most broadcasts. However even with such a network there is no guarantee that the attacker would have complete visibility to the communication medium. The attacker could monitor all of the targets broadcasts to see which data points change with time. If the attacker then removes all data points that were contributed by incoming messages the attacker can isolate the target car's identity.

Let β_0 be the set of all sets of centroids broadcast sent by the target vehicle at time 0. Let τ be the set of all broadcast that the attacker received from nodes other than the target. Let β_a be the broadcast sent by the target at the time of the attack. The set ι of centroids that the vehicle has in its internal database can be found by letting $\iota = \beta_a - \tau$. The sets β_0 and ι should be identical except for one set of centroids. The one set of centroids that could have changed would be the vehicles private data. In this way an attacker could find the targets private data.

Currently an identity is associated with a set of clusters. To protect against this attack the system needs to associate an identity with each cluster. This could be done by concatenating the cars identity with a counter and taking a one way hash of that string, then using that hash as the identity of the centroid in the gossip data structure. If each cluster has its own identity then the attacker can only learn one

thing, the target car's estimate of the road that it is on. This is of little value. In the first attack scenario the attacker is already physically following the target car. Therefore the attacker already knows what road the target is on. In the second attack scenario the attacker could already get a good idea of where the target is traveling simply by monitoring which wireless access point the target is accessing. The only new information available to the attacker is the exact roads the target is traveling. It is our opinion that this level of disclosure is acceptable to many drivers. The high value of real time traffic information justifies a small release in private information.

Chapter 8

CONCLUSIONS

The StreetSmart Traffic system has been shown in simulation to effectively find and disseminate automobile traffic congestion using ad hoc wireless networking. The system could be built on currently available technology, providing consumers with a useful tool in the near future. We introduce several approaches to finding clusters in a disconnected peer-to-peer network. The algorithms were been shown to perform well in challenging environments with non-stationary data and a sparsely connected dynamic network, converging to a good answer in the constrained setting of an automobile network.

This paper focused primarily on how to perform well with disconnected graphs. Further work should try to create a system that works well in a disconnected setting but also takes advantage of connected graphs. If a traffic system such as this was widely deployed it is conceivable that in a traffic jam a strongly connected graph would form. An ideal algorithm would take advantage of such a situation. In these situations a hierarchical network topology could provide significant advantages over the flat topology used in this paper.

This system has been designed to provide high value information to drivers in a timely fashion. While the promise of networking together vehicles to broadcast music or play games is intriguing, it has limited value to consumers. Instead of making the

time spent in cars more pleasurable, this system will allow people to spend as little time as possible in cars. If people can avoid traffic they can arrive at their destination quicker and then use traditional networks to enjoy music or video games.

The goal of this paper was to show that it is possible to build a peer-to-peer traffic monitoring system with standard components. These experiments demonstrated a possible solution to this problem. These experiments show the relationship between the accuracy of the system and a variety of factors. As more vehicles take part, more data is collected, and there is greater opportunities for connections. The physical constraints of the networking hardware limits the total number of connections, and the speed of data diffusion. In the very near future we will be able to build systems similar to StreetSmart at a price that consumers will be happy to pay.

REFERENCES

[1] http://mit.edu/its/dynamit.html.

[2] Traffic.com.

[3] Calling all cars. *Daimler Chrysler Hightech Report*, 2001.

[4] J. Ahn and J. S. Pierce. Serefe: Serendipitous file exchange between users and devices. pages 39–46. Mobile HCI, 2005.

[5] S. Bandyopadhyay, C. Gianella, U. Maulik, H. Kargupta, K. Liu, and S. Datta. Clustering distributed data streams in peer-to-peer environments. *Information Science Journal*, 2004. In Press.

[6] K. P. Birman. The surprising power of epidemic communication. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 97–102, 2003.

[7] Ken Birman, Robbert van Renesse, and Werner Vogels. Navigating in the storm: Using astrolabe for distributed self-configuration, monitoring and adaptation. 5th Annual International Active Middleware Workshop (AMS 2003), June 2003.

[8] C. Boulis and M. Ostendorf. Combining multiple clustering systems. In *8th European conference on Principles and Practice of Knowledge Discovery in Databases(PKDD), LNAI 3202*, pages 63–74, 2004.

[9] Z. Chen, H. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways. MobiHoc, 2001.

[10] I. Chisalita and N. Shahmehri. A peer-to-peer approach to vehicular communication for the support of traffic safety applications. pages 336–341. 5th IEEE

International Conference on Intelligent Transportation Systems, Singapore, Sept 2002.

[11] David Choffnes and Fabin E. Bustamante. Straw - an integrated mobility and traffic model for vanets. Number 10. International Command and Control Research and Technology Symposium (CCRTS), June 2005.

[12] M. Delio. Wireless caravan: Geeks on parade. *Wired Magazine*, 2003.

[13] M. Dello. Rent-a-car motto: Speed bills. *Wired Magazine*, July 2001.

[14] M. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode. Vitp: An information transfer protocol for vehicular computing. 2nd ACM International Workshop on Vehicular Ad-Hoc Networks (VANET), September 2005.

[15] Holger Fler, Marc Torrent-Moreno, Hannes Hartenstein, Matthias Transier, Roland Krger, and Wolfgang Effelsberg. Studying vehicle movements on highways and their impact on ad-hoc connectivity. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 2006.

[16] J. Garretson, W. Hess, J. Kanarek, M. Pignol, and M. Shai, 2005. http://roadcasting.org.

[17] S. Ghandeharizadeh and B. Krishnamachari. C2p2: A peer-to-peer network for on-demand automobile information services. First International Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (GLOBE'04), August 2004.

[18] Malcolm Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. Back Bay Books, 2002.

[19] Google. Google news.

[20] Mark Hallenbeck. Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation.

[21] E. Januzaj, H.-P. Kriegel, and M. Pfeifle. Dbdc: Density based distributed clustering. pages 88–105. EDBT in Lecture Notes on Computer Science, 2004.

[22] E. Johnson and H. Kargupta. Collective, Hierarchical Clustering From Distributed, Heterogeneous Data. In M. Zaki and C. Ho, editors, *Large-Scale Parallel KDD Systems. Lecture Notes in Computer Science*, volume 1759, pages 221–244. Springer-Verlag, 1999.

[23] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. Number 44. Annual IEEE Symposium on Foundations of Computer Science, 2003.

[24] W. Kowalczyk and N. Vlassis. Newscast em. *Advances in Neural Information Processing Systems*, pages 713–720, 2005.

[25] A. Lazarevic, D. Pokrajac, and Z. Obradovic. Distributed clustering and local regression for knowledge discovery in multiple spatial databases. Number 8, pages 88–105. European Symposium on Artificial Neural Networks, 2000.

[26] Tim Lomax and David Schrank. Urban mobility study. Technical report, Texas Transportation Institute, 2005.

[27] James E. Marca, Craig R. Rindt, and Michael G. McNally. Towards distributed data collection and peer-to-peer data sharing. Technical Report UCI-ITS-AS-WP-02-4, August 2002.

[28] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode. Trafficview: Traffic data dissemination using car-to-car communication. *ACM Sigmobile Mobile Comput-*

ing and Communications Review, Special Issue on Mobile Data Management, 8(3):6–19, July 2004.

[29] Wigle Net. wigle.net.

[30] Ramano P. The range vs. rate dilemma of wlans. *Wireless Net DesignLine*, January 2004.

[31] M. Parent and P. Daviet. Automated urban vehicles: Towards a dual mode prt (personnal rapid transit). *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, (4):3129–3134, 1996.

[32] P. Reiher, G. Popek, M. Gunter, and J. Salomone. Peer-to-peer reconciliation based replication for mobile computers. European Conference on Object Oriented Programming, Second Workshop on Mobility and Replication, June 1996.

[33] Assaf Schuster and Ran Wolff. Communicationefcient distributed mining of association rules. *Data Mining and Knowledge Discovery*, May 2004.

[34] Mukesh Singhal and Ajay Kshemkalyani. An efficient implementation of vector clocks. *Inf. Process. Lett.*, 43(1):47–52, 1992.

[35] Fredrik Svahn. In-car navigation usage: An end-user survey on existing systems. Technical report, Victoria Institute.

[36] L. Sweeney. *Computational Disclosure Control: Theory and Practice*. PhD thesis, MIT, 2001.

[37] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, pages 557–570, 2002.

[38] P. Tan, M. Steinbach, and V. Kumar. *Introduction to DataMining*. Pearson Addision Wesley, 2005.

[39] Jrg Widmer, Martin Mauve, Hannes Hartenstein, and Holger Fler. *The Handbook of Ad Hoc Wireless Networks*, chapter Position-Based Routing in Ad-Hoc Wireless Networks. CRC Press, 2002.

[40] Lars Wischhof, Andr Ebner, and Hermann Rohling. Self-organizing traffic information system based on car-to-car communication: Prototype implementation. International Workshop on Intelligent Transportation (WIT), MARCH 2004.

[41] I. Wokoma, I. Liabotis, O. Prnjat, L. Sacks, and I. Marshall. A weakly coupled adaptive gossip protocol for application level active networks. In *Policies for Distributed Systems and Networks*, pages 244–247, 2002.

[42] R. Wolff, K. Bhaduri, and H. Kargupta. Local l2 thresholding based data mining in peer-to-peer systems. Technical Report TR-CS-05-11, UMBC, 2005.

[43] F. Zhan. Three fastest shortest path algorithms on real road networks: Data structures and procedures. *Journal of Geographic Information and Decision Analysis*, 1(1):69–82, 1998.