Multi-modal design of an Intelligent Transportation System

by

Manish Shivshankar Chaturvedi

(201021007)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree

of

Doctor of Philosophy

in

Information and Communication Technology

to

Dhirubhai Ambani Institute of Information and Communication Technology

June 2016

Declaration

This is to certify that

1. the thesis comprises my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,

2. due acknowledgment has been made in the text to all other material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled "Multi-modal design of an Intelligent Transportation System" has been carried out by Manish Shivshankar Chaturvedi (201021007) for the degree of Doctor of Philosophy in Information and Communication Technology at this Institute under my supervision.

Thesis Supervisor Prof. Sanjay Srivastava

Abstract

An Intelligent Transportation System (ITS) plays a major role in generating fine grained vehicular traffic information for city wide or larger region. The real time traffic information is used to optimize traffic movement in a road network. However, due to the high cost of deployment and maintenance, limited ITS infrastructure is available in developing countries like India, and it is difficult to generate real time traffic information at large scale. Hence, there is a need for cost effective ITS solution.

The cellular network is widely deployed in India covering a major part of the road network. However, cellular network based positioning data has large location error (250-500 meters) making it unsuitable for the edge level travel time or speed estimation. Due to the increasing penetration of GPS enabled vehicles and smart phone users, the GPS probe data is considered an attractive source for real time travel speed estimation. However, the low penetration and only specific kinds of vehicles (cars and buses) having GPS make it inappropriate for developing countries like India. On Indian arterial roads, two wheelers form approximately 75% of the overall vehicle population.

In this dissertation, we study the problem of generating real time traffic information in a cost effective manner. There have been a variety of proposals in the literature that use one or more alternate sources of traffic information and are evaluated for various traffic conditions. Most of these solutions either use cellular network data alone and report a high error in the generated traffic information, or use data from other sources with or without fusion. This dissertation puts forward a novel mechanism of amalgamating widely available cellular network data, GPS probe data, and the data from limited ITS infrastructure for generating accurate traffic information.

Consequently, we design a map matching algorithm that processes erroneous vehicle location data collected using cellular network to generate vehicle trajectories. The vehicle trajectories are used to compute edge level vehicle flow, space occupancy, and congestion level data. The simulation results show the feasibility of accurate estimation of the traffic parameters in real time. We observe the need of additional accurate data sources for edge level speed estimation in the whole road network. Therefore, two models for selecting edges for ITS infrastructure deployment are proposed: the COngestion COverage MOdel (COCOMO) and the Edge COverage MOdel (ECOMO). The COCOMO selects edges

for ITS infrastructure deployment such that all the congestion levels (A to F) are covered by infrastructure edges; and the ECOMO finds clusters of similar edges based on their congestion profile and suggests infrastructure deployment on a few edges in each cluster. The edges with ITS infrastructure are used to learn the occupancy-speed relationship which is then spatially extrapolated to infrastructureless edges using GPS probe data to enable travel speed estimation on all the edges in the road network. The simulation results show that accurate edge level speed estimation is feasible using the proposed models. The infrastructure requirement of the COCOMO is constant and is independent of the number of edges in a road network. To the best of our knowledge, COCOMO is the only model with such a unique characteristic. The infrastructure requirement of ECOMO depends upon the diversity in congestion profile of edges. The models permit incremental infrastructure deployment and aim to maximize coverage using the available infrastructure units. This characteristic makes the proposed ITS suitable for deployment in developing countries like India. The robustness of the models is evaluated by disabling infrastructure on certain edges after deployment and its effect on error in speed estimation and coverage is observed. The simulation results show that the system can generate accurate speed estimations for all the edges or congestion levels covered with available infrastructure.

We design a MapReduce based distributed computing framework for the proposed ITS. The analysis of computation, communication, and storage requirement of the framework shows feasibility of large scale deployment of the ITS using available communication and storage technology.

To evaluate the utility of generated traffic information, an Advanced Traveler Information System (ATIS) is developed that uses the real time traffic information for trip planning and adaptive vehicle routing. The effect of application penetration on traffic condition (average vehicle trip duration and congestion distribution) in a road network is evaluated using simulations. The simulation results show that even with a small fraction of informed commuters, the traffic condition improves significantly. Also the performance of application with utilizing speed estimation from the proposed models is comparable to the performance with full deployment of ITS infrastructure.

To my parents,

who sacrificed a lot to educate me

Acknowledgements

I am deeply indebted to my thesis supervisor Prof. Sanjay Srivastava for his constant guidance, support and motivation. He devoted a significant amount of his valuable time to plan and discuss my Ph.D. work. Without his insightful inputs, it would have been very difficult for me to do quality work. I can not imagine someone being more selfless, having more of his students' interests in mind, or giving them more freedom. He has been a great source of inspiration for me since I took his first course during my graduate study at DA-IICT.

I extend my gratitude to Prof. Srikrishnan Divakaran and Prof Manjunath Joshi for their valuable suggestions during research progress seminars and during synopsis presentation, respectively. I am also thankful to DA-IICT for providing me the opportunity to pursue Ph.D.

I am thankful to my peers at DA-IICT Sunil Jardosh and Zunnun Narmawala whom I used to approach to discuss anything related to my Ph.D. work.

Last but not least, my sincere thanks to my wife Renu Sharma for her patience and constant support throughout this work.

Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
ANN	Artificial Neural Network
APTS	Advanced Public Transportation System
ATIS	Advanced Traveler Information System
ATMS	Advanced Traffic Management System
AVCS	Advanced Vehicle Control System
CCTV	Closed Circuit Television
COCOMO	COngestion COverage MOdel
CVO	Commercial Vehicle Operation
DFS	Distributed File System
DSRC	Dedicated Short Range Communication
ECOMO	Edge COverage MOdel
E-OTD	Enhanced Observed Time Difference
GA	Genetic Algorithm
GIS	Geographic Information System
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HDFS	Hadoop Distributed File System
HMM	Hidden Markov Model
ITS	Intelligent Transportation System
kNN	k Nearest Neighbor
LD	Loop Detector
MSC	Mobile Switching Center

PSO	Particle Swarm Optimization
RSSI	Received Signal Strength Indicator
SiMTraM	Simulation of Mixed Traffic Mobility
SUMO	Simulator for Urban MObility
SVM	Support Vector Machine
TRAI	Telecom Regulatory Authority of India
V/C Ratio	Volume to Capacity Ratio

Chapter 1

Introduction

1.1 Background

Road transport is the primary mode of transport in India which plays an important role in conveyance of goods and passengers and linking the centers of production, consumption and distribution. As per 2007-08 reports, the road network carry more than 56% of total freight traffic in the country [1].

The sustained economic growth, increasing disposable income, and rising urbanization has led to rising demand for road transport and personalized mode of transport (cars and two-wheelers), in particular [2]. The total number of registered motor vehicles has increased from 55 million in 2001 to 141.8 million in 2011 at the growth rate of 9.9%. On the other hand, the road network in the country has developed at the rate of 3.4% during the same period. That is the vehicle population has grown three times faster than the road network during the period. Eventhough the road network density in India (1.42 km/square km) compares favorably with many countries [3], the road network growth could not cope up with the explosive growth in vehicle population.

The share of two wheelers in overall vehicle population has increased from 8.8% in year 1951 to about 72% in the year 2011. On the other hand, the fraction of mass transit buses has reduced from 11% in year 1951 to 1.1% in the year 2011. The limited availability of mass transit and significant increase in affordable two wheelers along with limited road infrastructure are the major reasons of increasing traffic congestion in India.

Congestion wastes massive amount of time, fuel and money. As per urban mobility

report, 2011 [4], congestion is a significant problem in America's 439 urban areas. In 2011, 1.9 billion gallons of fuel was wasted (equivalent to about 2 months of flow in the Alaska Pipeline) and 4.8 billion hours of extra time was spent in vehicles (equivalent to the time Americans spend relaxing and thinking in 10 weeks) due to congestion in these areas. This resulted in $101 billion of delay and fuel cost (the negative effect of uncertain or longer delivery times, missed meetings, business relocation and other congestion related effects are not included). The cost to the average commuter was $713 in 2010 compared to an inflation-adjusted $301 in 1982.

An Intelligent Transportation System (ITS) uses electronics, communication, and information technology to improve efficiency and safety of the surface transportation [5]. The broad application areas of ITS are as listed below [6]:

- Advanced Traffic Management System (ATMS): The ATMS uses real time traffic information to predict traffic congestion or detect incident in a road network to improve efficiency of traffic movement. The traffic information may be used to control cycle time of adaptive traffic lights, enforce diversions or suggest alternate routes (to avoid incident region).

- Advanced Traveler Information System (ATIS): The ATIS aims to provide real time traffic information (location of incident, optimal route for a trip, road conditions, lane restrictions, etc.) to travelers in real time to enable choice of travel mode, planning of a trip or to make rerouting decisions during a trip.

- Advanced Vehicle Control System (AVCS): The AVCS is in-vehicle technology that aims to enhance the driver's control of a vehicle to make travel safer and efficient. It includes collision or lane departure warning systems, automatic braking system, etc.. The autonomous vehicles (e.g. driverless cars) represent the latest advancement in the category.

- Commercial Vehicle Operation (CVO): The private operators of trucks, vans and taxis track their vehicles in real time to improve productivity of their fleets and efficiency of their operations.

- The Advanced Public Transportation System (APTS) in urban and rural areas uses ITS to enhance the accessibility of information to users of public transportation and

to improve scheduling and utilization of the public transportation vehicles.

The thesis focuses on real time traffic information generation for the ATMS and ATIS. The traffic parameters of interest include vehicle flow, space occupancy, congestion, and speed. Vehicle flow is defined as the number of vehicles passing through an edge per unit time. It is a measure of service rate of an edge in a road network. An edge in a road network is a directed street connecting a pair of junctions. The space occupancy of an edge at a given time is defined as the ratio of number of vehicles present on the edge to the jam count of the edge. The jam count of an edge is defined as the maximum number of vehicles that can be accommodated on the edge. It is proportional to edge-length and number of lanes on the edge. In rest of the thesis the terms space occupancy and occupancy are used synonymously. Congestion is a measure of the comfort level of drivers while traveling in a road network. It is defined as the way in which vehicles interact to impede one another's movement. These interactions and their influence on the individual journey increase as the demand for road space reaches capacity. The Highway Capacity Manual (HCM 2000) designates six levels of service (LOS) for a road network: "A" represents the best operating conditions (from the drivers' perspective) and "F" represents the worst (jam condition or unstable traffic regime). Speed or Space Mean Speed is defined as the mean travel speed of vehicles while traversing an edge of known length.

The ITS infrastructure consists of traffic sensors (e.g. loop detectors, traffic cameras, etc.), infrastructure for communicating raw data and aggregated traffic information, computation infrastructure for processing raw data from individual traffic sensors to generate aggregated traffic information, and traffic control (e.g. adaptive traffic light control, dynamic message signs, etc.). The ITS Report of U.S. Department of Transportation [7] describes the deployment and maintenance cost of ITS infrastructure for various projects. According to the report, the cost of Closed Circuit Television (CCTV) is $50,000 per camera site. The estimated annual cost of Integrated Corridor Management system on the I-880 Corridor, San Francisco, California is $7.5 million considering the deployment and maintenance cost of hardware and software.

The high cost of traffic sensor deployment and maintenance leads to the exploration of alternative solutions. The cellular network is a widely deployed communication infras-

tructure worldwide. According to the Telecom Regulatory Authority of India (TRAI) press release 65/2015 [8], there are over 996 million cellular connections in India and the cellular teledensity is greater than 78%. In urban areas, the cellular teledensity is more than 145%. Hence, it is reasonable to assume that every vehicle in a road network is equipped with a cell phone. The location of all the cellular users are tracked by the infrastructure for efficient call forwarding. The location error of 100-500 meters is reported in the literature [9] [10] [11] [12]. Several studies have been carried out to assess feasibility of using noisy location data of vehicles collected from cellular network for edge level speed or travel time estimation [13] [14] [15]. The results show that estimates are highly erroneous (mean speed error of 15% or more).

Due to the increasing penetration of GPS enabled vehicles and smart phone users, the GPS probe data is considered an attractive source for real time travel speed estimation. The Mobile Century field experiment [16] demonstrated the use of GPS probe vehicles for generating traffic information. The results showed that 2-3% penetration of GPS enabled cell phones in the driver population is necessary to generate velocity estimates of traffic flow. In a similar but large scale study in Singapore [17], GPS enabled taxi probe data was used to estimate traffic volume information. However, the GPS probe data is not directly useful in Indian traffic scenario due to two major reasons: first, GPS probe penetration is low in India and second, GPS is mostly found on cars and public transport buses, whose movement characteristics are very different than that of two wheelers which form approximately 75% of the overall vehicle population on Indian arterial roads [18][19].

The researchers have also explored possibility of fusing data from multiple sources to generate accurate edge level traffic information. Park et al. [20] developed an expanded neural network model to estimate edge level travel speed using dual loop detector data and DSRC (Dedicated Short Range Communication) probe vehicle data. In a similar study [21], the Artificial Neural Network (ANN) technique was used to build a travel time estimation model with input traffic data coming from GPS-equipped intercity buses, vehicle detectors along the roadway, and the incident database. Anand et al. [22] used extended Kalman filter for fusing the vehicle flow data extracted from video and the travel time data from GPS equipped vehicles to estimate vehicle traffic density. In all the mentioned studies, the accurate data from multiple sources are fused either to increase coverage or to derive some traffic parameter for which the measure is not directly available

from traffic sensors.

We observe that no study in the literature claims to generate accurate edge level travel time or speed information using cellular network data alone. Most of the efforts are focused on effective processing of cellular signaling data to reduce location error with little emphasis on designing algorithms for generating edge level traffic information using erroneous location estimates. To the best of our knowledge, no study in the literature has attempted fusion of cellular network data with other sources to improve accuracy of the generated traffic information.

1.2 Problem Definition

1.2.1 Problem Statement

Design an Intelligent Transportation System (ITS) using cellular network and other available data sources with minimal additional infrastructure (traffic sensors).

1.2.2 Objectives

- To generate the accurate estimation of edge level traffic information in real time and examine its utility.
- To design the ITS infrastructure deployment models and evaluate them for infrastructure requirement, feasibility of incremental deployment and fault tolerance.
- To design and validate a distributed processing and communication framework for the proposed ITS to assess feasibility of large scale deployment.

1.2.3 Assumptions

- All the edges in the road network are covered by cellular infrastructure.
- The classification of cellular network users (vehicles or non-vehicles) is already done. Feasibility of the same is reported in [14] and [23].
- The cellular network provides location estimates periodically for all the vehicles using active signaling. The analysis of signaling overhead shows that active tracking

of all the vehicles at the interval of 30 seconds consumes approximately 5% of location capacity of a cellular carrier [15].

- The GPS probes report their precise positions periodically.
- The ITS infrastructure deployed on an edge provides accurate speed information for the edge periodically.
- Road network carries heterogeneous traffic similar to the Indian arterial roads with a large fraction of two wheelers (75%) [18][19].

1.2.4 Outcome

- Design of the proposed multi-modal ITS permitting real time accurate estimation of edge level traffic information.
- Distributed processing and communication framework for the proposed ITS.

1.3 System Model

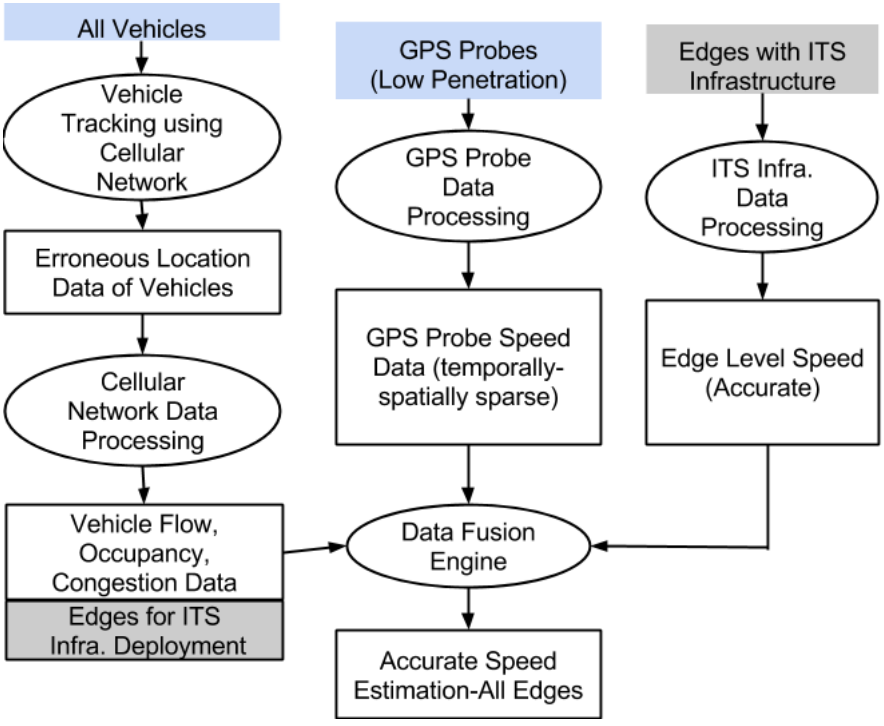


Figure 1.1: System Model

Figure 1.1 shows the system model. It is assumed that the cellular network generates location estimates of all the vehicles periodically using active signaling. It is a reasonable assumption considering cellular teledensity in India. If a small fraction of vehicles is not equipped with a cell phone, the traffic parameter estimation is not affected.

To examine location error in cellular network data, we carried out a localization experiment on a road stretch of 18 km in Ahmedabad city, India using fingerprinting approach. The experiment reports mean location error of less than 50 meters in dense population regions and less than 200 meters in relatively sparse population regions. In the dense population regions, 6-7 neighbor cells in addition to the main cell are visible with good signal strength (greater than -80 dbm). In the sparse population regions, 2-3 neighbor cells are visible with signal strength of less than -90 dbm.

To generate the vehicle-edge mapping using erroneous location estimates, we designed a map matching algorithm that processes a series of location points having maximum location error of 250-500 meters. The generated vehicle trajectories are used to estimate edge level vehicle flow, space occupancy and congestion level. Simulation results show that the mean estimation error of less than 10% is achievable in most of the cases.

We observe that speed estimation using cellular network data alone is erroneous and there is a need of additional accurate data sources to improve accuracy. To enable the accurate edge level speed estimation in whole road network, two models of ITS infrastructure deployment are proposed: COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO). Both the models use congestion profile of edges to select edges for ITS infrastructure deployment. The COCOMO aims to cover all six congestion levels using infrastructure edges. The maximum number of infrastructure units required to K-cover all the congestion levels are $6 \times K$. The ECOMO forms clusters of similar edges using the congestion profile of edges and deploys infrastructure on a few edges in each cluster to ensure K-coverage of all the edges in a road network.

The infrastructure requirement of COCOMO is independent of the road network size, whereas that in ECOMO is a function of road network size (as the road network size increases, the diversity in congestion profiles and hence the infrastructure requirement increases). The system does not make any assumption about the type of ITS infrastructure (any sensor providing speed estimation can be used) making it flexible to support variety of them (e.g. loop detector, video camera, etc.).

Due to low penetration of GPS probes and deployment on specific kinds of vehicles (cars and buses), the real time data of GPS probes is not used in the model. Instead, the average GPS probe speed data is computed and stored for every congestion level for every edge in a road network. It is used to spatially extrapolate the speed estimation from infrastructure edges to the infrastructureless edges.

On the edges with ITS infrastructure, the accurate speed information is available periodically, in addition to the traffic parameters (vehicle flow, space occupancy, and congestion) estimated using cellular network data. The occupancy-speed relationship is learned on infrastructure edges using polynomial regression. The aim is to spatially extrapolate occupancy-speed relationship learned on the infrastructure edges to the other edges in road network. However, it is observed that the vehicle speed on an edge is affected by many parameters (in addition to space occupancy). For example vehicle movement speed and edge length are highly correlated. For similar space occupancy values, the vehicle movement speed on a shorter-length edge is lower than that on a longer edge. To incorporate or overcome the effect of these parameters on speed estimation, we use historical data of GPS probes to spatially extrapolate the occupancy-speed relationship from infrastructure edges to infrastructureless edges. The simulation results show that accurate speed estimation with ninety percentile error of 10-22% and 10-13% is achievable in COCOMO and ECOMO, respectively.

The performance of the infrastructure deployment models is evaluated for limited infrastructure deployment (to check feasibility of incremental deployment) and for fault tolerance when a certain fraction of infrastructure is not available. The simulation results show that the models permit the graceful degradation of service when sufficient infrastructure is not deployed or is not available due to failure.

To deal with the unavailability of GPS probe data, the COCOMO and ECOMO were modified to incorporate the effect of static parameters of edges, namely, edge length, number of lanes, degree of the edge and presence of traffic lights at the junction. The simulation results show that the modified models have higher infrastructure requirement and generate less accurate speed estimations (ninety percentile error of 20-25%).

We represent the proposed ITS as a MapReduce based distributed computing framework. The MapReduce framework proposed by Google permits data processing on a large cluster of commodity machines and is proven to work well under heavy computation load

condition involving large amount of data [24]. The analysis of computation, communication and storage requirement of the proposed system shows feasibility of large scale deployment using the currently available communication and storage technology.

The edge level speed estimations generated by the system has a certain amount of error. Also, the estimations may not be available on some edges or at some times due to unavailability of infrastructure. We developed an Advanced Traveler Information System (ATIS) that uses the real time traffic information generated by the system to suggest alternate routes to commuters during their trip. Simulation results show that the system is effective in reducing average trip duration and congestion in the road network. Additionally, the performance of ATIS with using speed information from the proposed models is comparable to the performance with full deployment of the ITS infrastructure.

1.4 Organization of the Thesis

Chapter 2 surveys proposals for Intelligent Transportation System (ITS) design. The mechanisms for generating real time traffic information using cellular network, GPS probes, dedicated sensors, and multiple sources are discussed and analyzed. The ITS infrastructure requirement of various proposals are examined. The chapter concludes with remarks on limitation of state of the art and contributions of the thesis. Chapter 3 elaborates design of the proposed ITS. The details of proposed ITS infrastructure deployment models, namely, COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO) are discussed. The mechanism for estimating real time traffic information by fusing data from cellular network, GPS probes and dedicated ITS infrastructure is elaborated. Chapter 4 studies performance of the proposed ITS using simulations. The ITS infrastructure requirement and accuracy of traffic information is examined for various scenarios. Chapter 5 discusses a MapReduce based distributed computing framework for the propose ITS. The communication and storage requirement of the proposed ITS is analyzed to evaluate feasibility of large scale deployment. The design of an Advanced Traveler Information System (ATIS), that consumes traffic information generated by the proposed ITS, is discussed in chapter 6. The simulation based performance analysis of the ATIS is elaborated. Finally, the conclusion and future work are written in chapter 7.

Chapter 2

Related Work

The effective functioning of an Advanced Traveler Information System (ATIS) and Advanced Traffic Management System (ATMS) requires real time traffic information for the whole road network. The research on traffic information generation is broadly classified into three categories:

- Traffic information generation using cellular network: The aim of these studies is to use widely deployed cellular infrastructure for generating meaningful traffic information. The problems addressed by these studies include localization and vehicle tracking, and generating fine-grained or coarse-grained traffic information. The fine-grained traffic information includes edge level travel time and speed estimation; the coarse-grained information includes identifying hot spots in a road network, and observing traffic movement pattern (origin-destination) at different times of day during week-days and week-ends.

- Traffic information generation using dedicated sensors: These studies use a variety of sensors for generating fine grained traffic information. The sensors include loop detectors, traffic cameras, Bluetooth or RF sensors, acoustic sensors, GPS or Dedicated Short Range Communication (DSRC) probe vehicles, etc.. The performance is evaluated in terms of accuracy of the generated traffic information and cost effectiveness.

- Traffic information generation using multiple sources: These studies combine traffic data from various sensors to improve coverage, derive some traffic parameter for which the measurement is not available, or make the solution more cost effective.

2.1 Traffic Information Generation using Cellular Network

2.1.1 Localization

Cellular based localization techniques are broadly classified in two categories [23]: active techniques generate additional signaling traffic to estimate users' location, but permits location tracking of any user terminal independent of its state; passive techniques silently collect signaling information from one or more points (A interface for location area updates or A_{bis} interface for handover data) in the cellular network with no impact on offered network load, but the amount and type of retrieved information depend on the placement of the monitoring points and on the state of the user terminal (more activity on user terminal permits more information to be collected). Cayford et al. [15] carry out detailed study of amount of signaling required for generating a reasonably accurate traffic information using active techniques.

Trevisani et al. [10] and Raja et al. [11] study performance of various location technologies used in GSM (Global System for Mobile Communication) cellular network and report that Enhanced Observed Time Difference (E-OTD) has location error of 50-150 meters independent of cell size. Chen et al. [9] compare performance of centroid based, finger printing based and Gaussian process based localization algorithm, in the Downtown area (66 cells per square kilometers) and residential area (26 cells per square kilometers) with respect to location error, amount of training required, speed of execution, and storage requirement. They carried out performance analysis using top seven in-range cells from single or all providers, and all in-range cells from all providers. They reported the location errors as mentioned in Table 2.1. Varshavsky et al. [12] used fingerprinting and centroid algorithm for indoor and outdoor localization. They report the median error of 75 meters and 213 meters with fingerprinting and centroid algorithm respectively. Mohan et al. [25] observed very dense cell tower deployment in Bangalore city of India (average inter tower spacing of about 100 meters) and reported the median error of 117m in location estimate.

Table 2.1: Performance of Positioning Algorithms: Median Location Error (meters)

Algorithm	Downtown (66 cells/sqkm)			Residential(26 cells/sqkm)		
	Single provider(7)	Cross provider(7)	Cross provider(all)	Single provider(7)	Cross provider(7)	Cross provider(all)
Centroid	232	166	170	760	456	574
Finger-printing	94	153	245	277	313	297
Gaussian process	126	87	65	196	147	134

2.1.2 Traffic Information Generation

Valerio et al. [23] [26] analyze mobile hand off related cellular network signaling data to generate road traffic information. They make the following observations: (1) cellular signaling pattern is different on week days and weekends; (2) cellular signaling pattern is different at different times of a day; (3) train users generate different cellular signaling pattern than car users; (4) in an event of incidence, the signaling notch (high decrease) followed by a peak occurs. This clearly indicates that the cellular user classification using hand off or other cellular signaling data is possible.

Bar-Gera [13] processed cellular data of handover events collected on 14km of Ayalon freeway in Israel with 10 interchanges in both directions during January-March 2005 to estimate travel times. The cellular based system received observations for about 1-3% of the total traffic during day time (1000-2000 hrs) and generated 63% valid travel time estimates for 27 road segments. Cellular data was more noisy (14%) than loop detector data (5%). The noise was measured as the average absolute relative difference between travel time estimates for consecutive five minutes intervals. However, the algorithms used for map matching and travel time estimation are not described in the work.

Calabrese et al. [14] used cellular signaling information available at A_{bis} interface (handover) and A-interface (location area updates) in the city of Rome. A_{bis} signaling data was processed in real time to predict user terminals' position and speed to produce the traffic map. Received signal power (RXLEV and RXQUAL) and Time Advance (TA) value were used to estimate the location of active terminals. The location error of 159 meters in urban area, 295 meters in suburban areas, and up to 1457m in the extra urban area were reported. The error in considering moving user as still and vice-versa was only 3.2%. The travel time estimation error when compared to the readings taken using GPS

and odometer was 14.88% on bypass roads, 10.08% on primary urban streets, and 17.66% on secondary urban streets. The A interface signaling was used to generate coercive grain location information about active or idle users using location area updates.

Traffic Online, Vodafone [27], analyzed signaling information on A interface and A_{bis} interface to generate traffic information. They claim to generate high quality traffic information without mentioning methods or algorithms used in signal processing.

Liou et al. compute cell residence time and edge level speed using sparse cellular handover data [28]. They propose the LinChangHuangfu (LCH) scheme and evaluate its performance for speed estimation on National Highway 3 in Taiwan. To improve the accuracy of speed estimation, the road segment filtering (using location area information) and historical vehicle traces are used. The cellular data of a fifteen minutes period is aggregated for speed estimation. After bias removal, the mean discrepancy between the cellular based speed estimation and loop detector data is 7.51%.

Demissie et al. [29] use cellular handover data to generate non-realtime traffic state information. The study shows that there is correlation between the traffic volume on a road segment and count of handover in the corresponding cell (correlation coefficient of 0.76 is reported). The solution uses Multinomial logit and Artificial Neural Network (ANN) to relate the sparse data of handover to the traffic state on arterial roads. The performance of the proposed methodology is evaluated using five case study areas of Lisbon city, Portugal. The accuracy of 78.1% is achieved in traffic state estimation.

Caceres et al. [30] observe that if the cell boundary or location area boundary is precisely known, the number of vehicles crossing that boundary can be counted just by counting the number of hand overs (considering the percentage of users making a call at a given time). This can be used as an induction loop detector, counting number of vehicles crossing it, provided the cell is sectored and cell boundary or sector boundary maps to a unique road segment. However the assumption made in the paper, i.e. cell boundary or location area boundary is precisely known and it maps to a unique road segment, is very unrealistic.

Cayford et al. at Institute of Transportation Studies, Berkeley conducted a study to evaluate effect of location accuracy, frequency of location measurements, and number of locations monitored, on traffic information generation using cellular network [15]. With location error of 100 meters, a vehicle could be mapped to a correct road for 98.4% of

all surface streets and 98.9% of all free ways. With the update frequency of 30 seconds (preferred by all the cellular carriers) and location accuracy of 50 meters, 98.8% of the road segments could be identified. As the number of locations increases from 1 per second per square mile to 10 per second per square mile, the percentage of roads covered increases very rapidly. Above 20 locations per second per square mile, the increase in the percentage of roads covered declines, and there is little benefit of using more than 40 locations per second per square mile. With network based location technology, the measurements of 85% of the roads can be generated in every five minutes interval using approximately 5% location capacity of a cellular carrier. Operating continuously, the traffic information approaches 97.7% of the road segments, the maximum possible with a location technology accurate up to 100m using 30 seconds update frequency. The similar results are observed in handset based location technology. The authors did not consider location errors larger than 100 meters. The algorithms used for map matching and traffic estimation are not mentioned in the paper.

NCHRP 70-01 Report, 2007 [31] contains a survey of various projects/studies carried out on using cellular based data for generating traffic information.

Table 2.2 summarizes the works on the traffic information generation using cellular network.

2.2 Traffic Information Generation using GPS Probes

The Mobile Century field experiment [16] demonstrated the use of GPS probe vehicles for generating traffic information on a free way. The data collection was done using 100 GPS probe vehicles over a ten miles road stretch on the I-880 highway, near Union city, California, for eight hours. The data analysis showed that a 2-3% penetration of GPS probes is required to get atleast one speed reading for every crossing point on the road stretch during an aggregation period of five minutes. It was reported that the speed data collected from GPS probes and loop detectors were not in agreement at few locations due to bias in data from either source. Zhao et al. [32] analyze the effect of number of GPS probe samples on edge level speed estimation. The authors use Curve-Fitting Estimation Model (CFEM) to estimate traffic information using GPS probe data. The experiment results show that 90% accuracy in speed estimation requires at least 15 GPS points for

Table 2.2: Traffic Estimation using Cellular Network

Research work	Traffic parameters of interest	Data collection	Methodology	Accuracy
Bar-Gera, 2007 [13]	travel time	2000 hrs data on 14km of Ayalon freeway in Israel	A and A_{bis} interface data processing	63% travel times valid (15% noise)
Valerio et al., 2009 [23] [26]	user classification (road-non road users) and incident detection	beltway A23 South-East and the highway A2 South from Vienna to Italy	A and A_{bis} interface data processing	unique signaling pattern is observed for road or non-road users, and incident
Calabrese et al., 2010 [14]	user classification and travel time estimation	100 km^2 area of Rome	A and A_{bis} interface data processing	3.2% error in user classification and 14-18% in travel time estimation
Liou et al., 2013 [28]	speed	at specific location on National Highway 3 in Taiwan	A_{bis} interface data processing	mean error of 7.5%
Demissie et al., 2013 [29]	density	five areas of Lisbon City, Portugal	A_{bis} interface data processing	accuracy of 78.1%

each edge per aggregation period. Minh et al. [35] use Artificial Neural Network (ANN) and Genetic Algorithm (GA) to overcome low penetration of GPS probes and estimate vehicle speed and density information. The authors show that at least 25% GPS probe penetration is required for convergence of the proposed model. The traffic parameters are estimated with accuracy of more than 90%.

Balan et al. [33] use historical data of GPS taxi probes to estimate travel time and trip-fare in real time. The authors use data of 250 million taxi trips collected using 15000 taxis for 21 months in Singapore city. The k-nearest neighbor (kNN) technique along with domain knowledge of peak hour fare and congestion pricing as enforced by the authorities is used for travel time and fare estimation in real time. The system supports thousands to millions of queries per second with fare error of less than 1 Singapore dollar and mean trip duration error of less than 3 minutes.

Zhang et al. propose Path2Go [34], a multimodal traveler information system using GPS data. The system processes transit and road network information, and real time GPS data from mobile users, buses and trains to identify mode of user (e.g. walking, traveling by bus or train, driving, etc.) and enables context aware service. The system

Table 2.3: Traffic Estimation using GPS Probes

Research work	Traffic parameters of interest	Data collection	Methodology	Accuracy
Mobile Century field experiment, 2010 [16]	speed	data of 100 GPS probes collected on 10 mile stretch of I-880 near Union City, California for 8 hours	GPS probe data are processed for speed estimation; for user privacy, virtual trip lines are used	speed data of GPS probe and LD does not match at different locations and times due to bias in LD data or GPS probe data
Zhao et al., 2011 [32]	speed	one day data of more than 16000 GPS equipped taxis collected in central region of Shanghai, China	Curve-fitting estimation model	90% accuracy when number of GPS data points are more than 15
Balan et al., 2011 [33]	trip duration	data of 250 million taxi trips collected using 15000 taxis for 21 months in Singapore	k-nearest neighbor and domain knowledge	trip duration error of less than 3 minutes
Path2Go, 2011 [34]	travel mode	data of 1000 users collected over four months in San Francisco Bay Area	analysis of road network data and GPS traces of users and public transport	92% accuracy
Minh et al., 2012 [35]	speed and density	data of three locations on National route no.16, Japan	Artificial Neural Network and Genetic algorithm	more than 90% accuracy in speed estimation

exploits user itinerary and mode information to conserve battery power by turning off the GPS and reducing communication to the server. The experiment results show that the system detects travel mode with 92% accuracy. Nitsche et al. [36] attempts to classify travel mode among the eight classes (e.g. walk, bicycle, motorcycle, car, train, etc.) using sensors commonly available in smartphones. The proposed solution uses cellular signaling and accelerometer data to enable location tracking in case of GPS outage. For travel mode classification, a probabilistic classifier working on a randomly chosen subset of features is combined with discrete Hidden Markov Model (HMM). The proposed approach is evaluated using 355 hours of probe data collected over two months in Vienna, Austria. Experiment results show that more than 80% of travel modes are identified correctly.

Table 2.3 summarizes the works on the traffic information generation using GPS probe data.

2.3 Traffic Information Generation using Dedicated Sensors

Sen et al. propose "Kyun Queue" [37] using 802.15.4 based RF transceivers for real time traffic state classification (congested or not congested) and vehicle queue monitoring at a junction. The system consists of a series of RF transmitter-receiver pairs deployed across a road. The transmitter continuously transmits beacon packets and the receiver measures packet reception ratio and signal strength. The authors observe and exploit the strong correlation between the link metrics and the occupancy level of road between a pair of sensors to compute the traffic parameters. The system can be used on multi-lane roads carrying lane sharing heterogeneous traffic observed in developing countries like India. The experimental data of 16 hours collected on two roads of Mumbai, India show 90% accuracy of traffic state classification. Kassem et al. [38] [39] use 802.11 based RF transceivers for human-vehicle classification, vehicle counting and speed estimation in a single-lane set up. The median speed error of 15% is reported based on 45 minutes of data collected on a 7.5m long single lane road stretch.

Sen et al. [40] [41] used a pair of acoustic sensors deployed on road side to observe a Doppler shift in the measured frequencies of vehicular honks. The system uses honk detection and matching to estimate vehicle speed and in turn classify the road traffic condition as congested or free flowing. The relative speed error of less than 10% is reported based on 18 hours of data collected on two road stretches of Mumbai, India.

Ali et al. design a multiple-inductive-loop sensor which is suitable for lane sharing heterogeneous traffic [42]. The loops are connected serially and every loop has unique resonance frequency and inductance. The change in inductance and resonance frequency is used to detect and classify vehicles moving over the loops. With automatic or manual periodic calibration, the loops permit accurate vehicle classification (accuracy more than 95%) and speed estimations (error of less than 1km/hr).

Taghvaeeyan and Rajamani design a portable magnetic sensor system for vehicle counting, classification, and speed estimation in a lane adjacent to the sensor [43]. The variations in magnetic field due to vehicle movement are processed using a signal processing algorithm to achieve accurate estimates of traffic parameters (vehicle counting error of less than 1% and speed estimation error of less than 2.5%). Robustness of the system

is shown by counting number of right turns at an intersection with 95% accuracy.

Levenberg uses a pair of inertial sensors (accelerometers) deployed in a road pavement for vehicle speed estimation [44]. The vehicle speed estimation is done by finding the cross correlation between accelerometer signals. The estimation error of less than 2km/hr in sixty percent cases and less than 5 km/hr in ninety percent cases is observed. Stocker et al. design a software system architecture that uses vibration data collected from accelerometer sensors, digital signal processing, machine learning, and knowledge representation and reasoning for vehicle detection (vehicle or no vehicle) and classification (heavy or light vehicle) [45]. The Semantic Sensor Network Ontology (SSNO) is used to describe sensor data and the Situation Theory Ontology (STO) is used to represent real world situation or knowledge generated after processing sensor data. The system uses accelerometers deployed on road side to measure road pavement vibrations generated by vehicle movement. The precision and recall of vehicle detection are 95% and that of vehicle classification is more than 80%.

Zoto et al. [46] used two spatially separated Bluetooth sensors to estimate the average speeds of vehicles traveling in different types of lanes, e.g. express lanes and local lanes, without knowing the lanes individual vehicles were traveling in.

Table 2.4 summarizes the works on the traffic information generation using dedicated sensors.

2.4 Traffic Information Generation using Multiple Sources

Park and Lee [20] process dual loop detector data and Dedicated Short Range Communication (DSRC) probe data using Bayesian estimator and an expanded neural network for speed estimation (three levels of speed) on arterial roads in the city of Jeonju in Korea. Traffic data of two days (total eight hours) are used to validate the models. The link speed error of less than 10 km/hr is observed. The high error in speed estimation is due to the small sample of data and discrete levels of speed estimation (only three). In a similar study, Wie and Lee [21] use Artificial Neural Network (ANN) to forecast travel time of intercity buses. The authors use data from GPS-equipped intercity buses, vehicle

Table 2.4: Traffic Estimation using Sensors

Research work	Traffic sensor(s) used	Traffic parameters of interest	Data collection	Methodology	Accuracy
Sen et al.,2010 [40] [41]	acoustic sensors	traffic state and speed	18 hours of data collected on two road stretches of Mumbai, India	Doppler shift analysis of vehicular honk frequencies	less than 10% speed error
Sen et al.,2012 [37]	802.15.4 based RF transceivers	traffic state and vehicle queue	16 hours of data collected on two roads of Mumbai, India	analysis of packet reception ratio and signal strength	90% accuracy in traffic state classification
Kassem et al., 2012 [38] [39]	802.11 based RF transceiver	vehicle count and speed	45 minutes of data collected on 7.5 m long single lane road stretch	SVM, statistical and curve fitting	speed error of 15%
Ali et al.,2013 [42]	multiple Induction Loop	vehicle classification and speed	heterogeneous traffic data of less than 500 vehicles	analysis of variations in inductance and resonance frequency	vehicle classification error of 5% and speed error of less than 1km/hr
Rajamani et al.,2014 [43]	magnetic sensor	vehicle count, classification, and speed	data of 188 vehicles	analyze variations in magnetic field due to vehicle movement	vehicle counting error of less than 1% and speed estimation error of less than 2.5%
Levenberg, 2014[44]	accelerometers	speed	data of 19 vehicles collected at three places of deployment	analysis of cross-correlation between accelerometer signals	ninety percentile error of 5km/hr

detectors, and incident database, collected for nine days on 89 km of freeway stretch in Taiwan. The travel time error of less than 20% is observed in all the cases. Anand et al. [22] use extended Kalman filter for fusing vehicle flow data manually extracted from a video and travel time data from GPS equipped vehicles to estimate heterogeneous vehicle traffic density. The data collection was done for two days (total 5 hours) on IT corridor in Chennai, India. The mean estimation error of less than 10% is observed. In a city scale study in Singapore [17], the data from GPS enabled taxi probes and loop detectors are used to estimate all-vehicle traffic volume information. The one month data of 16000 taxis and 1000 loop detectors is used in the study. The authors relate taxi count data generated by GPS probes and all-vehicle count data from loop detector using logistic regression and linear regression, and report an error of less than 10%. To spatially ex-

Table 2.5: Traffic Estimation using Multiple Sensor Data Fusion

Research work	Traffic sensor(s) used	Traffic parameters of interest	Data collection	Methodology	Accuracy
Park & Lee, 2004[20]	loop detectors(LD) and DSRC probes	speed	8 hours data collected on arterial roads of Jeonju, Korea	Bayesian estimator and an expanded neural network	error of less than 10 km/hr
Wie & Lee, 2007[21]	LD, GPS on buses, incident data	travel time	data of 9 days collected on 89 km of freeway in Taiwan	Artificial Neural Network	less than 20% error
VTrack, 2009[47]	GPS and WiFi	travel time and traffic hot-spot detection	800 hours of driving traces gathered from 25 cars having GPS and WiFi	Hidden Markov Model (HMM) and Viterbi decoding based map matching	25% error in travel time, 80% hotspots detected accurately
CTrack, 2011[48]	cellular data, compass, accelerometer	vehicle trajectory	urban driving traces of 126 hours	two pass HMM	more than 75% of median accuracy
Anand et al., 2011[22]	video and GPS	heterogeneous vehicle traffic density	5 hours data collected on IT corridor, Chennai, India	extended Kalman filter	less than 10% error
Aslam et al., 2012[17]	LD and GPS probes	vehicle count	one month data of 16000 taxis and 1000 loop detectors in Singapore	logistic regression, linear regression	less than 30% error
Bhaskar et al., 2014[49]	LD and Bluetooth sensor	speed and density	3 hours simulation based on real traffic data	cumulative plots	more than 90% accuracy
Nitech et al., 2014[36]	cellular data, accelerometer, GPS	travel mode	data of 355 hours collected in Vienna, Austria	probabilistic classifier and discrete HMM	more than 80% accuracy

trapolate the relationship to edges not having a loop detector, the authors identify similar edges based on static parameters (Euclidean distance, orientation, number of lanes) and the difference between taxi count. For edges without loop detector, the all vehicle count is estimated from taxi count data using the regression model learned on similar edges having a loop detector. The estimation error of less than 30% is reported.

In VTrack [47], the travel time estimation and traffic hotspot detection using GPS and WiFi based vehicle trajectory is attempted. The system tolerates GPS outage by processing inaccurate position samples of WiFi (upto 70 meters location error) using Hidden Markov Model (HMM) and Viterbi decoding based map matching technique. The

authors observe that accurate suggestions for shortest route are feasible even with erroneous (25% median error) edge level travel time estimations. In CTrack [48], Thiagarajan et al. avoid use of battery power consuming sensors like WiFi and GPS, and generate vehicle movement traces using the cellular network based localization. The authors use two-pass HMM to process cellular based erroneous location data (175 meters of mean error). Some common systematic errors in localization are corrected using accelerometer and magnetic compass data. The real driving traces of 126 hours (1074 miles) in an urban environment are used to evaluate performance of the system. The experiment results show that CTrack retrieves over 75% of the users' drive accurately in the median.

Bhaskar et al. propose a data fusion model to estimate speed and density on traffic light controlled urban roads using loop detector (point data) and a pair of Bluetooth sensors (zone data) [49]. The authors observe that Bluetooth sensors provide good estimates of vehicle travel time, but the data is sparse in time and space, whereas the travel time and density estimations using vehicle count data from loop detectors are erroneous. The proposed model fuses data from these two complementary sources to estimate travel time and vehicle density accurately (more than 90% accuracy). The model is validated using real traffic data and simulations. Bachmann et al. [50] use loop detector data and vehicle probe data (collected from Bluetooth sensors, DSRC probes, or others) to evaluate six data fusion techniques such as Bar-Shalom/Campo, Kalman filters, OWA, Artificial Neural Network, etc. for vehicle speed estimation. The authors simulate a 5 km road stretch of Highway-400 in Toronto, Canada using Paramics microscopic simulator. The simulation results show that all the fusion techniques generate statistically better speed estimates than individual sources. A small fraction of probe vehicles significantly improves speed estimation (as much as 40%) when loop detector data is highly erroneous, specifically under congested traffic conditions. As the penetration of probe vehicles increases, the advantage of fusion reduces as the probe data itself accurately represents the average traffic speed.

Table 2.5 summarizes the works on the traffic information generation using multiple data sources.

2.5 Analysis of ITS Infrastructure Requirement

The Sensor Location Problem aims to determine the optimal number and placement of traffic sensors to ensure coverage of the whole road network for traffic information estimation. The solutions are attempted in the literature using three approaches [51]: first, the mathematical optimization approach with the objective of maximizing flow coverage or minimizing sensor requirement, assuming constraints such as flow conservation; second, the algebraic approach where a system of linear equations based on link-path incidence or link-node incidence are solved using one step or iterative method; and third, the graph theory based approach in which the spatial relationship between links and nodes in a network is exploited to identify a set of critical edges for sensor deployment. L. Bianco et al. identify the minimum number of nodes in a road network for sensor deployment to enable the flow volume estimation for all the edges in a road network [52]. The authors assume that the flow split ratio are available for all the edges incident to a node from historical data and exploit the conservation of flow principle for estimating flow on all the edges in a road network. The authors formulate flow estimation problem as a system of linear equations and deploy sensors on a certain number of nodes to reduce the number of unknowns in the equations to enable solution. When a sufficient number of sensors are not available, the system is under specified and the system of linear equations may have an infinite number of solutions [53]. In that case, the problem becomes bi-level: the upper level decides the sensor location and the lower level computes best possible flow estimation for locations of interest. Fie et al. in [54] study the problem in the similar context and suggest that the sensors should be deployed independent of one another on the most unstable links carrying a large volume of vehicles and having high variation (e.g. links upstream of the recurring bottleneck). To identify the edges for sensor deployment, the authors introduce a notion of eigen-volumes and eigen-links to measure network observability and uncertainty. The authors in [55] extend the study to maximize information gain and O-D coverage under stochastic traffic conditions (e.g. in presence of incident and vehicle re-routing). The authors use Greedy Randomized Adaptive Search Procedure, which is an adaptation of the deterministic procedure proposed in their earlier study ([54]). The performance of the current sensor location assignment is evaluated by generating a series of random incidents, and relocation is done in an iterative manner.

The results show a 13-17% reduction in demand uncertainty. Hu et al. [56] identify minimum number of edges for infrastructure deployment using linear algebra based method. The authors use link-path incidence matrix to identify a subset of edges that form the basis of the vector space. The solution does not require the flow split ratio or other traffic information. The authors determine infrastructure requirement for various types of networks and analyze effect of network size, network connectivity, number of O-D pairs and number of paths on infrastructure requirement. Experiment results show that the infrastructure requirement varies from 60-80% of number of edges in the road network. The authors also study the sensor location problem using heterogeneous sensors - active sensors (license plate recognizer) and passive sensors (vehicle detectors) [51]. The authors identify sensor deployment locations using road network graph analysis, and full path coverage and flow conservation constraints. The resultant deployment permits O-D demand estimation with 5-25% error for different road networks.

Park et al. propose a mechanism to deploy additional portable sensors at selective locations in a road network to reduce the travel time error [57]. The authors assume that the fixed sensors like loop detectors are already deployed in the road network and enable estimation of travel time error. Based on this information, the portable sensors are relocated in a network while considering the relocation expenses and reduction in error. The simulation results show an error reduction of more than 100% in certain cases.

Zhu et al. propose a mobile sensor (probe) based traffic monitoring system using Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [58]. A measure of traffic information acquisition benefits is developed and used to compare the surveillance performance of the proposed system with static sensor deployment. The simulation results show that the mobile sensors perform better than fixed deployment under all traffic conditions specifically with fewer number of sensors. The advantage of mobile sensors over fixed sensors decreases with increase in congestion and number of sensors. Under moderate and heavy congestion, and with more than 15 and 19 sensors, respectively, the objective function value is lower for mobile sensors than fixed sensors in a road network with 9 nodes and 28 links.

Table 2.6 summarizes works on ITS infrastructure deployment for traffic information generation.

Table 2.6: Summary of Literature Survey - Infrastructure Deployment

Research work	Traffic sensors	Traffic parameters of interest	Methodology	Infra. requirement	Accuracy
Hu et al.,2009 [56]	–	flow	link-path incidence matrix and linear algebra based analysis of network topology assuming conservation of flow	60-80% of edges	mathematically accurate flow estimation
Hu et al.,2014 [51]	loop detector and license plate recognizer	origin-destination flow	analysis of road network, full path coverage and flow conservation constraints	65-70% of number of edges (12 sensors- 4 active and 8 passive - in Fishbone network having 18 links; 120 sensors - 48 active and 72 passive - in NCKU network, Taiwan city having 154 links)	6.08 % error in Fishbone network and 21.46% error in NCKU network
Zhu et al., 2014[58]	mobile sensor	traffic information benefit	Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).	35 mobile sensors provide 62% (1.87/3) of traffic information acquisition benefit in SiouxFall network having 76 links	

2.6 Discussion

The following observations are made from the literature survey:

- Due to the large location error in cellular data (50-250 meters on average) [10] [11] [9] [12] [25], the traffic information estimation using cellular network is erroneous (the mean error is 14-22%) [14] [29].

- GPS probes provide accurate estimation of traffic information (more than 90% accuracy) but require sufficient penetration (25% according to [35] and 15 GPS data points per aggregation period according to [32]). Spatial extrapolation of GPS probe data to all vehicle data is erroneous (30% error reported in [17]).

- For heterogeneous traffic monitoring, the use of RF transceivers [37], acoustic sensors [40] [37], multiple-inductive-loop sensor [42], magnetic sensor [43], and accelerometer [45] is explored in literature. While dedicated sensors provide accurate traffic information, their network-wide deployment is costly.

- Multi-sensor data fusion is carried out using loop detectors, probe data (GPS, DSRC), WiFi, and Bluetooth sensor data. All these sensors provide accurate traffic information and data fusion is used to improve coverage of the generated traffic information, derive some unmeasurable traffic parameter of interest, or to make the solution cost effective. CTrack [48] is the only study to the best of author's knowledge that combines cellular network data with other data sources (accelerometer and magnetic compass) to improve accuracy of vehicle trajectory estimation (75% median accuracy).

- The works on infrastructure deployment models assume conservation of flow or availability of turning ratio information for whole road network [52] [56] [51]. These assumptions are very unrealistic for road network of developing countries like India. Also, infrastructure deployment is required on 60-80% of edges in the road network [56] [51].

The present work differs from the literature as follows:

- The erroneous location data of vehicles collected from cellular network are processed using a novel map matching algorithm and temporal extrapolation to derive vehicle flow and congestion information in real time. The simulation results show the estimation error of less than 10% in arterial road network scenarios.

- To enable edge level speed estimation, a set of edges for ITS infrastructure deployment are identified using historical data of congestion. Two models for infrastructure deployment, namely, COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO) are proposed. The infrastructure requirement of COCOMO depends upon the number of congestion levels (six i.e. A to F) and is independent of the road network size. As ECOMO forms clusters of similar edges using congestion profile, the infrastructure requirement depends upon diversity in the congestion profile of edges. Simulation results show that full coverage of a road network can be assured with infrastructure deployment on less than 30% edges.

- As GPS probe speed data are sparse (due to low penetration) and biased (due to deployment on certain types of vehicles), they are not used in real time and are

collected for every congestion level. Historical data of GPS probes are used for spatial extrapolation of speed estimation from infrastructure edges to infrastructureless edges. Use of GPS probe data improves accuracy of speed estimation significantly. Simulation results show ninety percentile error of 10-22% and 10-13% for COCOMO and ECOMO, respectively.

The proposed ITS is also evaluated for feasibility of large scale deployment and the utility of generated traffic information. Consequently, a MapReduce based [24] distributed processing framework for the proposed ITS is designed. The computation, communication, and storage requirement of the framework are analyzed to establish feasibility of large scale deployment. An Advanced Traveler Information System (ATIS) is designed using real time traffic information generated by the proposed ITS. The ATIS suggests trip route and en-route changes to end-users based on real time traffic condition in a road network. The simulation results show that the average trip duration and congestion in the road network improves significantly.

Chapter 3

Proposed Approach

3.1 Localization Using Cellular Network Data

To assess location error in the cellular network, we use fingerprinting based localization. The reason for using this technique is that it does not require access to the cellular infrastructure elements and is viable using any GPS enabled smart phone. An Android application was developed to collect cellular fingerprint (main cell ID and RSSI value, neighbor cells and RSSI values) and corresponding GPS coordinates every second. Appendix-1 specifies the format and sample of data records. The data collection was done for 10 days while moving from Bapunagar area to Nirma university area in Ahmedabad city of India. The movement stretch is 18 kilometers long and traverses through typical urban roads of the city. The dataset is published on web [59] and is available for download.

Algorithm 3.1: Localization Using Fingerprinting Approach

Input: a database F containing (C, L) pairs, where C and L specify a cellular fingerprint and corresponding GPS coordinates, respectively; an input cellular fingerprint, C_i for which the location is to be computed

Output: location estimate L_i for C_i

1: Extract $F_J = \{(C_j, L_j)\}$ such that the main cell ID of C_j equals the main cell ID of C_i
2: **for all** $(C_j, L_j) \in F_J$ **do**
3: Compute S_j, the Cosine similarity between C_j and C_i
4: **end for**
5: Let $F_K = \{(C_k, L_k)\}$ be the K records having the highest similarity score
6: $L_i = \sum_{k=1}^{K} W_k \times L_k$, where $W_k = \frac{S_k}{\sum_{p=1}^{K} S_p}$

For computing location error, the leave-one-out cross validation is used wherein the

nine days data are used as historical data to compute location estimates for the remaining day data. The localization process is elaborated in Algorithm 3.1. We use cosine similarity to find the similarity score between a pair of fingerprints. With a given cellular fingerprint C_i as an input, the set of fingerprints having the same main cell ID, F_J, is extracted from the historical data set, and cosine similarity between every pair $(C_i, C_j), \forall C_j \in F_J$ is calculated. The location estimate is computed as the weighted sum of locations associated with the K most similar $C_j \in F_J$. K=5 is used in the experiment. The weights are assigned based on the cosine similarity value. The experiment results show the mean location error of less than 50 meters in the regions with very dense cell tower deployment (6-7 neighbor cell towers in addition to the main cell are visible with good signal strength of approximately -80 dbm). In the regions with relatively sparse cell deployment (2-3 neighbor cells), the mean location error of less than 200 meters is observed.

3.2 Map Matching Algorithm

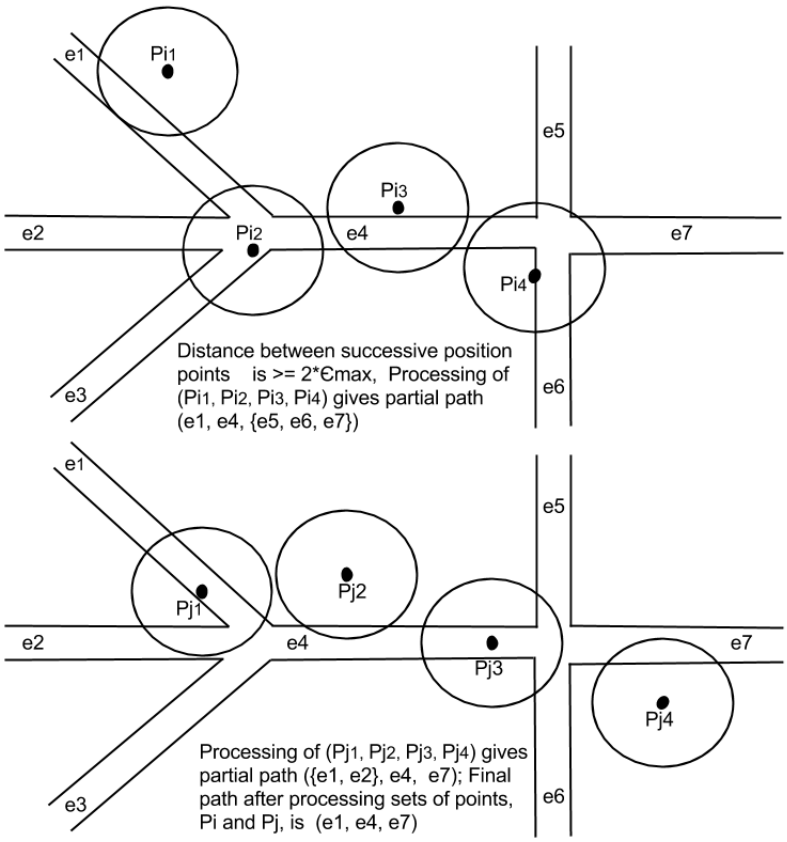


Figure 3.1: Functioning of the Map Matching Algorithm

Algorithm 3.2: Map Matching Algorithm

Input: road network topology, series of position estimates for a vehicle taken every 30 seconds, maximum position error ϵ_{max}

Output: estimated path of a vehicle

1: Let $P = (P_1, P_2, \ldots, P_n)$ be a series of position estimates of a vehicle recorded at time $T = (t_1, t_2, \ldots, t_n)$, respectively, where $t_{i+1} = t_i + 30$ seconds, $\forall i < n$.
2: **for all** $P_i \in P$ **do**
3: Let E_i be the set of edges overlapping with a circle centered at P_i and having radius ϵ_{max}. E_i contains all the edges the vehicle may be traversing at time t_i. Each $e \in E_i$ is of the form $[ID, position]$, where, ID is a unique identifier of the edge and $position = (x_1, y_1, x_2, y_2)$ represents end points of overlapping region of the edge.
4: **end for**
5: **for all** $P_i \in P$ **do**
6: $P_{i_1} = P_i$
7: Select position points P_{i_j}, $j = 2, \ldots, 4$ such that the distance between $P_{i_{j-1}}$ and P_{i_j} is minimum and is greater than $2 \times \epsilon_{max}$.
8: **for** $j = 1, \ldots, 3$ **do**
9: Get E_{i_j} and $E_{i_{j+1}}$ for P_{i_j} and $P_{i_{j+1}}$, respectively.
10: **for all** $(e_1, e_2) \in E_{i_j} \times E_{i_{j+1}}$ **do**
11: **if** $e_1.ID == e_2.ID$ **then**
12: Determine direction of vehicle movement using $e_1.postion$ and $e_2.position$ and compare with direction of the edge.
13: If the directions do not match, discard (e_1, e_2) from further consideration.
14: Else if minimum distance between the two segments can not be traversed by a vehicle at maximum speed (S_{max}), discard (e_1, e_2) from further consideration.
15: Else record (e_1, e_2) for further processing.
16: **else if** $e_1.ID \neq e_2.ID$ **then**
17: Get path from e_1 to e_2 using the route table generated using Dijkstra's shortest path algorithm.
18: If the path can not be traversed by a vehicle at maximum speed, discard (e_1, e_2) from further consideration.
19: Else, record $(e_1, path, e_2)$, for further processing.
20: **end if**
21: **end for**
22: **end for**
23: Generate partial path using $(e_i, [\text{optional path}], e_j)$ information recorded in above three iterations using concatenation operations.
24: **end for**
25: Generate full path of a vehicle trip using partial paths generated so far. This involves concatenation, set intersection and sequencing operations.

After examining location error in the cellular network, a map matching algorithm is developed that processes the location estimates of a vehicle generated using cellular network to compute the vehicle trajectory. The map matching process is described in Algorithm 3.2. Due to large location error, it is not possible to determine a correct edge or correct movement-direction of a vehicle using single location point. Hence, a series of location points are processed to get the direction of vehicle movement. Instead of mapping the vehicle to a particular edge, we map it to a set of edges and then try to filter out some of the improbable edges using antecedent or subsequent position estimates. This mechanism works well except for the location points towards the beginning or end of the trip. Due to the absence of appropriate antecedent or subsequent position estimates in these cases, it becomes difficult to determine exact edges. Specifically, it is difficult to predict whether a vehicle took a U-turn at the beginning or end of its journey. Such cases are not taken care of by the map matching algorithm and it is assumed that a vehicle does not take U-turn at the beginning or end of its trip. However, this assumption leads to the prediction of a truncated path when a vehicle actually takes U-turn at the beginning or end of its trip.

Figure 3.1 shows functioning of the map matching algorithm. A location point along with three subsequent location points, separated by at least $2 \times \epsilon_{max}$ from one another, is processed to estimate partial path traversed by a vehicle. This adds some amount of overlapped processing of location estimates and gives overlapping partial path. A unique partial path estimation is computed by considering only common edges which are part of all the predicted partial paths. This assures the correct edge selection independent of the amount of location error. The complete path of a vehicle trip is estimated by concatenating the overlapping partial paths.

Computation complexity of the algorithm is $\theta(|P| \times |E|^2)$, where P is a series of position estimates for a vehicle and E is the set of edges overlapping with a position estimate. $|P|$ and $|E|$ represents the number of elements in sets P and E, respectively. $|P|$ increases with the sampling frequency and $|E|$ increases with the location error.

3.3 Traffic Parameter Estimation

The map matching algorithm processes a series of position estimates for each vehicle v_j to generate a probable set of edges, E_{t_i,v_j}, the vehicle may be traversing at time step t_i. It is observed that a vehicle spends more time at the down stream end of an edge than at the up stream end of an edge. This is due to the fact that vehicles reduce their speed while approaching a junction, whereas they move with higher speed and acceleration at the upstream end of an edge. Moreover, at a traffic light controlled junction, vehicles may be forced to stop for one or more cycle times, further extending time spent at the downstream end of an edge. Hence, the exit time of a vehicle v_j from an edge e, $t^{exit}_{v_j,e}$, is approximated as the maximum time at which the vehicle was detected on the edge. This approximation work well in most of the cases, specifically for traffic light controlled edges. The vehicle v_j's duration for edge e, $D_{v_j,e}$, is adjusted as per the approximated exit time from the predecessor edge. Let $C_{A,e}$ be the count of vehicles that were detected on edge e during aggregation period A. Hence,

$$C_{A,e} = |V| \text{ where } V = \{v_j | D_{v_j,e} \text{ overlaps with } A\} \tag{3.1}$$

The vehicle flow, $f_{A,e}$, on edge e during aggregation period A is defined as the count of vehicles exiting from edge e during aggregation period A. Hence,

$$f_{A,e} = |V| \text{ where } V = \{v_j | t^{exit}_{v_j,e} \text{ falls in } A\} \tag{3.2}$$

The space occupancy, $\rho_{A,e}$, of edge e at the end of aggregation period A is defined as

$$\rho_{A,e} = \frac{C_{A,e} - f_{A,e}}{J_e} \tag{3.3}$$

where $(C_{A,e} - f_{A,e})$ is the number of vehicles present on edge e at the end of the aggregation period A, and J_e is the jam count on edge e. The jam count of an edge specifies the maximum number of vehicles that can be accommodated on the edge and is proportional to the length of the edge and the number of lanes on the edge. In rest of the thesis the terms space occupancy and occupancy are used synonymously.

To avoid the instantaneous error in the estimation of traffic parameters, we aggregate

the data for sufficiently large period (aggregation period of 10 minutes) and record the aggregated information every minute.

3.3.1 Removing Time Lag in Traffic Data

As the map matching algorithm needs to process a series of erroneous position points before associating a vehicle to an edge, the vehicle flow and space occupancy have a time lag of about one aggregation period, and can not be used in a real time traffic information system. To overcome the above limitation, temporal extrapolation of vehicle count and flow data is required.

Various statistical methods for time series forecast (exponential moving average, Box-Jenkins, regression, etc.) are compared in [60] and shown that the exponential moving average gives a reasonably good short range forecast with the least computation and storage requirement. Also, machine learning techniques such as regression, artificial neural network, genetic algorithm, etc., are used successfully in literature for temporal extrapolation [61]. The machine learning techniques have higher storage requirement in general as they use historical data for learning model parameters. The regression based models are the simplest among machine learning techniques and have the least computation requirement.

Considering the real time processing requirement of vehicle count and flow data, we choose the simplest and least computation intensive methods from each category, namely, (1) exponential moving average and (2) polynomial regression, for temporal extrapolation of traffic data.

In exponential moving average method, the temporally extrapolated flow, f_{t+h}^{ext}, is computed as,

$$\begin{aligned} f_{t+h}^{ext} &= \overline{f_t^{est*}} + h \times \overline{G_t}, \\ \overline{f_t^{est*}} &= \overline{f_t^{est}} + \left(\frac{1-\alpha}{\alpha}\right) \times \overline{G_t}, \\ \overline{f_t^{est}} &= \alpha \times f_t^{est} + (1-\alpha) \times \overline{f_{t-1}^{est}}, \\ \overline{G_t} &= \beta \times \left(\overline{f_t^{est}} - \overline{f_{t-1}^{est}}\right) + (1-\beta) \times \overline{G_{t-1}} \end{aligned} \tag{3.4}$$

where, f_t^{est} is estimated flow at time t, $h = 10$, $\alpha = \beta = 0.125$ (chosen as suggested in

[60] for the extrapolation period). The $\overline{f_t^{est}}$ is smoothed average of estimated flow at time t and typically lags the most recent changes in flow. Hence the adjustments are made to bring the average up to date and $\overline{f_t^{est*}}$ is computed using $\overline{G_t}$ as suggested in [60]. The $\overline{G_t}$ defines the average trend in flow variation at time t.

The regression based extrapolation method learns the relationship between the flow at time t and $t+h$ on an edge using historical data:

$$f_{t+h,e} = a_0 + (a_1 \times f_{t,e}) + \left(a_2 \times f_{t,e}^2\right) + \epsilon \tag{3.5}$$

where a_0, a_1, a_2 are the regression model parameters and ϵ is residual. The flow estimation at time t is provided as input to the regression model of an edge to compute the temporally extrapolated flow.

3.3.2 Congestion Level Estimation

Congestion level of an edge is estimated using volume to capacity ratio (V/C ratio) [62]. The V/C ratio on edge e during aggregation period A is defined as,

$$V/C_{A,e} = \frac{f_{A,e}}{f_e^{cap}} \tag{3.6}$$

Capacity flow on edge e, f_e^{cap}, is defined as the maximum average flow on edge e over fifteen minutes time span and is computed using historical vehicle flow data of the edge. The V/C ratio ranges as reported in [63] for lane sharing heterogeneous traffic are used to classify the traffic condition on an edge among six congestion levels, A-F (Table 3.1).

Table 3.1: V/C Ratio for Congestion Levels

Congestion Level	V/C ratio range
A	<0.125
B	0.125 - 0.276
C	0.276-0.479
D	0.479-0.715
E	0.715 - 1.00
F	>1.00

To avoid misclassification at the boundary values, range estimates (instead of point estimates) are used. That is, if $V/C_{A,e}$ is the estimated volume to capacity ratio on

edge e during aggregation period A, the range estimate for the same is computed as $[(1-\epsilon) \times V/C_{A,e}, (1+\epsilon) \times V/C_{A,e}]$. If the range estimate of V/C ratio is contained in the V/C ratio range of single congestion level (Table 3.1), unique congestion level is estimated; otherwise, two congestion levels are estimated. For example, if estimated V/C ratio is 0.28 and $\epsilon = 0.1$, the range estimate of V/C ratio $[0.252, 0.308]$ spans across V/C ratio ranges of congestion level B and C. So, congestion level BC is returned in this case.

The congestion status on every edge is recorded every minute. Using the historical data of congestion, the congestion profile of every edge is computed. The congestion profile of an edge contains information about the percentage of overall time the edge spends under given congestion level. As there are six congestion levels (A-F) reported in the literature, the congestion profile of an edge, ϱ^{e_i}, is represented as six-tuple:

$$\varrho^{e_i} = (\varrho_A, \varrho_B, \varrho_C, \varrho_D, \varrho_E, \varrho_F)^{e_i} \tag{3.7}$$

The congestion profile of edges is used for selecting edges for ITS infrastructure deployment and edge level speed estimation as described in section 3.5.

3.3.3 Error in Flow Estimation

Let $\tau^v = [t_1^v, t_2^v]$ be the ambiguous period during which a vehicle v is detected on multiple edges, i.e. edge e and the successor edge, by the map matching algorithm. The span of ambiguous period, $|\tau^v|$, is a function of location error, L^ϵ, and speed of the vehicle, S^v:

$$|\tau^v| = \frac{2 \times L^\epsilon}{S^v} \tag{3.8}$$

and, the average span of ambiguous period is,

$$|\overline{\tau}| = \frac{2 \times L^\epsilon}{\overline{S}} \tag{3.9}$$

where $\overline{S}$ is the average speed of vehicles. It is observed that vehicles reduce their speed while approaching a junction (down stream end of an edge), whereas they move with higher speed and acceleration at the upstream end of an edge. That is a vehicle spends the larger fraction of ambiguous period on edge e than on the successor edge. Hence, the

updated average span of ambiguous period, after discarding its first half, is

$$|\overline{\tau}| = \frac{L^{\epsilon}}{\overline{S}} \tag{3.10}$$

The ambiguous period of a vehicle v, $[t_1^v, t_2^v]$ overlaps with an aggregation period $A = [t_1^A, t_2^A)$, in one of the following three ways: (i) $t_1^v < t_1^A < t_2^v < t_2^A$ (ii) $t_1^A \leq t_1^v < t_2^v < t_2^A$, and (iii) $t_1^A < t_1^v < t_2^A < t_2^v$. In case (i) and case (iii), ambiguous period spans across the aggregation period boundary leading to error in flow estimation. In case (ii), the ambiguous period is contained in the aggregation period and does not cause any error in flow estimation. In the current proposal, the vehicle is assigned to edge e (the predecessor edge) in case of ambiguity, i.e. it is assumed that the vehicle exited edge e at time t_2^v. Hence, case (i) and case (iii) leads to over counting and under counting, respectively, in flow estimation of the aggregation period. The difference between over count and under count is the flow error during the aggregation period. t_1^v can take any value in $[t_1^A - |\tau^v|, t_2^A)$ for τ^v to overlap with A. Case (i) applies when $t_1^v \in [t_1^A - |\tau^v|, t_1^A)$. Probability for the same is,

$$P_1 = P\left(t_1^v \in [t_1^A - |\tau^v|, t_1^A)\right) = \frac{t_1^A - (t_1^A - |\tau^v|)}{t_2^A - (t_1^A - |\tau^v|)} = \frac{|\tau^v|}{t_2^A - t_1^A + |\tau^v|} \tag{3.11}$$

Case (ii) applies when $t_1^v \in [t_1^A, t_2^A - |\tau^v|)$. Probability for the same is,

$$P_2 = P\left(t_1^v \in [t_1^A, t_2^A - |\tau^v|)\right) \;=\; \frac{t_2^A - |\tau^v| - t_1^A}{t_2^A - t_1^A + |\tau^v|} \tag{3.12}$$

Case (iii) applies when $t_1^v \in [t_2^A - |\tau^v|, t_2^A)$. Probability for the same is,

$$P_3 = P\left(t_1^v \in [t_2^A - |\tau^v|, t_2^A)\right) = \frac{t_2^A - (t_2^A - |\tau^v|)}{t_2^A - (t_1^A - |\tau^v|)} = \frac{|\tau^v|}{t_2^A - t_1^A + |\tau^v|} \equiv P_1 \tag{3.13}$$

Let actual flow at the beginning and end of the aggregation period A be $f_{A,e}^1$ and $f_{A,e}^3$, respectively. Then, the flow error on edge e for aggregation period A is,

$$\begin{aligned} f_{A,e}^{\epsilon} &= f_{A,e}^1 \times P_1 - f_{A,e}^3 \times P_3 \\ &= P_1 \times \left(f_{A,e}^1 - f_{A,e}^3\right) \end{aligned} \tag{3.14}$$

The equation (3.11) and (3.13) show that flow error decreases with increase in aggregation period and increases with increase in location error. The equation (3.14) shows, if the flow variations do not occur during an aggregation period (specifically at the beginning and end of the aggregation period), flow error is negligible.

Effect of Flow Estimation Error on Temporal Extrapolation of Flow

Let f_t be actual flow, f_t^{est} be the estimated flow, f_t^{ϵ} be the error in flow estimation, and f_t^{ext} be the temporally extrapolated flow, at time t. $f_t^{est} = f_t + f_t^{\epsilon}$. According to equation 3.4,

$$\begin{aligned} \overline{f_t^{est}} &= \alpha \times f_t^{est} + (1-\alpha) \times \overline{f_{t-1}^{est}} \\ &= \sum_i \alpha \times (1-\alpha)^i \times f_{t-i} + \sum_i \alpha \times (1-\alpha)^i \times f_{t-i}^{\epsilon} \\ &= \overline{f_t} + \overline{f_t^{\epsilon}} \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} \overline{G_t^{est}} &= \beta \times \left(\overline{f_t^{est}} - \overline{f_{t-1}^{est}}\right) + (1-\beta) \times \overline{G_{t-1}^{est}} \\ &= \sum_i \beta \times (1-\beta)^i \times (f_{t-i} - f_{t-i-1}) + \sum_i \beta \times (1-\beta)^i \left(f_{t-i}^{\epsilon} - f_{t-i-1}^{\epsilon}\right) \\ &= \overline{G_t} + \overline{f_t^{\sigma_\epsilon}} \end{aligned} \tag{3.16}$$

where $\overline{f_t^{\sigma_\epsilon}} = \sum_i \beta \times (1-\beta)^i \left(f_{t-i}^{\epsilon} - f_{t-i-1}^{\epsilon}\right)$ represents the error in trend estimation due to flow estimation error. Substituting values from equation 3.15 and 3.16 in equation 3.4, we get

$$\begin{aligned} f_{t+h}^{ext} &= \overline{f_t^{est*}} + h \times \overline{G_t^{est}} \\ &= \overline{f_t^{est}} + \left(\frac{1-\alpha}{\alpha}\right) \times \overline{G_t^{est}} + h \times \overline{G_t^{est}} \\ &= \overline{f_t^{est}} + \left(\frac{1-\alpha}{\alpha} + h\right) \times \overline{G_t^{est}} \\ &= \overline{f_t} + \overline{f_t^{\epsilon}} + \left(\frac{1-\alpha}{\alpha} + h\right) \times \left(\overline{G_t} + \overline{f_t^{\sigma_\epsilon}}\right) \end{aligned} \tag{3.17}$$

Hence, the error in flow extrapolation due to error in flow estimation is

$$f^{\epsilon}_{t+h} = \overline{f^{\epsilon}_{t}} + \left(\frac{1-\alpha}{\alpha} + h\right) \times \overline{f^{\sigma_\epsilon}_{t}} \tag{3.18}$$

The minimum error in flow extrapolation occurs when $\overline{f^{\sigma_\epsilon}_{t}}$ (variation in flow estimation error with time) is minimal. In that case $\overline{f^{\sigma_\epsilon}_{t}} \approx 0$, and $f^{\epsilon}_{t+h} \approx \overline{f^{\epsilon}_{t}}$.

When variations in flow estimation error are high, the second term in equation 3.18 becomes significant and flow extrapolation error increases in a multiplicative manner.

Due to a small extrapolation period (h=10), the variations in flow estimation error are typically low leading to reasonably accurate flow extrapolation. For example, consider a traffic light controlled edge where the actual flow observes a zig-zag pattern and the estimated flow is smoothed out due to large location error. For instance, the temporal sequence of flow values on an edge is $f + \Delta f$, $f - \Delta f$, $f + \Delta f$, and so on, whereas the estimated flow value is roughly f in all the measurements. Hence, the f^{ϵ}_{t} sequence is Δf, $-\Delta f$, Δf, and so on, and the $\left(f^{\epsilon}_{t-i} - f^{\epsilon}_{t-i-1}\right)$ sequence is $2 \times \Delta f$, $-2 \times \Delta f$, $2 \times \Delta f$, and so on. Hence, we get

$$\overline{f^{\epsilon}_{t}} = \sum_{i} \alpha \times (1-\alpha)^{i} \times f^{\epsilon}_{t-i} = \frac{\alpha \times \Delta f}{2-\alpha}$$

and,

$$\overline{f^{\sigma_\epsilon}_{t}} = \sum_{i} \beta \times (1-\beta)^{i} \left(f^{\epsilon}_{t-i} - f^{\epsilon}_{t-i-1}\right) = \frac{2 \times \beta \times \Delta f}{2-\beta}$$

Substituting these values along with $\alpha = 0.125$, $\beta = 0.125$, and $h = 10$ in equation 3.18, we get $f^{\epsilon}_{t+h} \approx 2.33 \times \Delta f$.

3.4 GPS Probe Data Collection and Use

A small fraction of vehicles in India (specifically cars and public transport buses) are GPS-enabled. The periodic position updates reported by GPS probes are used to compute GPS probe speed on an edge during an aggregation period and is recorded every minute. However, the GPS probe speed data are not directly useful for real time edge level speed estimation due to the following reasons: first, due to low penetration of GPS probes, speed estimates are not available for all the aggregation periods and for all the edges;

and second, GPS is deployed on some fraction of cars and public transport buses which have very different speed characteristics compared to the two wheelers that form about 75% of the overall vehicle population on Indian arterial roads.

We use GPS probe data as follows: when the GPS probe data is available for an edge for a given aggregation period, it is recorded along with the congestion level data. The historical data of GPS probes collected in this manner is used to compute speed transition value for extrapolating vehicle speed estimation from the infrastructure edge to the infrastructureless edge and to compute anticipated speed error to be reported to the end user (discussed in section 3.5.3).

3.5 Edge Level Speed Estimation

As noted earlier, speed estimation using cellular network data alone is erroneous, hence additional accurate data sources are needed to improve accuracy. This section details the methodology adopted for selecting edges for ITS infrastructure deployment and edge level speed estimation. Two models for ITS infrastructure deployment are proposed: COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO). Both the models use congestion profile of edges to select optimal set of edges for ITS infrastructure deployment.

3.5.1 Infrastructure Deployment using COngestion COverage MOdel (COCOMO)

The COCOMO aims to cover all the congestion levels occurring in a road network using infrastructure edges. When a congestion level is observed on an infrastructureless edge, the infrastructure edge covering the congestion level is used for speed estimation on that edge.

The congestion profile of an edge is computed using the historical data of congestion levels observed on the edge. Let the congestion profile of an edge e_i be ϱ^{e_i}. An edge that spends sufficient time (typically >25%) under given congestion level is a candidate for having infrastructure to cover the congestion level. The intuition behind choosing this criterion for infrastructure edge selection is, an edge that has observed a congestion level

for a significant fraction of time in the past is likely to observe the same in the future (may be with slight change in the fraction) even after variations in traffic condition in the region over a long time scale. Infrastructure edges are selected by computing K-coverset for all the congestion levels using a greedy approximation algorithm [64]. The methodology is described in Algorithm 3.3.

Algorithm 3.3: Infrastructure Edge Selection for K-coverage of Congestion levels

Input: a set of candidate edges for having infrastructure, $E = \{e_1, e_2, \ldots, e_n\}$; K; and congestion profile of all the edges: $\varrho^{e_i}, \forall e_i \in E$

Output: the set of edges for infrastructure deployment: I

1: Determine the congestion levels covered by every edge $e_i \in E$ using its congestion profile ϱ^{e_i}.
2: Infrastructure Edge Set I={}
3: Count=$\{c_A, ..., c_F\}$, where c_j is number of times the congestion level j is covered by the infrastructure edges so far. Set $c_j = 0, \forall\ j = A, \ldots, F$;
4: R is set of congestion levels yet to be covered to fulfill K-coverage criteria; set R=$\{A, \ldots, F\}$
5: **repeat**
6: Select e_i from E such that it covers maximum number of j's in R.
7: I = I $\cup$ $\{e_i\}$; E = E $-$ $\{e_i\}$
8: For each congestion level j covered by e_i, $c_j = c_j + 1$
9: For each $c_j \geq K$, remove j from R.
10: **until** R={}

$K = 3$ is used in simulations to ensure availability of three infrastructure edges for a given congestion level. Here we need to mention that the infrastructure requirement is independent of the road network size and is determined by K and the number of congestion levels. The model permits incremental infrastructure deployment. When the number of ITS infrastructure units, N, is specified, a variation of the Algorithm 3.3 is used to select N edges for ITS infrastructure deployment (Algorithm 3.4): the algorithm tries to 1-cover all the congestion levels using minimal infrastructure units, and repeats the process until all N infrastructure units are deployed.

3.5.2 Infrastructure Deployment using Edge COverage MOdel (ECOMO)

The ECOMO aims to cover all the edges in a road network using infrastructure edges. An edge e_j is said to be covered by an infrastructure edge e_i if ϱ^{e_j} is similar to ϱ^{e_i}, where

Algorithm 3.4: Maximal Congestion level Coverage using Limited Infrastructure

Input: a set of candidate edges for having infrastructure, $E = \{e_1, e_2, \ldots, e_n\}$; count of infrastructure units, N; and congestion profile of all the edges: $\varrho^{e_i}, \forall e_i \in E$

Output: the set of N edges for infrastructure deployment: I;

1: Determine the congestion levels covered by every edge $e_i \in E$ using its congestion profile ϱ^{e_i}.
2: Infrastructure Edge Set I={}
3: R is set of congestion levels yet to be covered at current coverage level; set R=$\{A, \ldots, F\}$
4: **while** Number of edges in I <N **do**
5: Select e_i from E such that it covers maximum number of j's in R.
6: I = I $\cup$ $\{e_i\}$; E = E $-$ $\{e_i\}$;
7: For each congestion level j covered by e_i, remove it from R if it is present
8: If R={}, set R=$\{A, \ldots, F\}$
9: **end while**

ϱ^{e_i} and ϱ^{e_j} are congestion profile of e_i and e_j, respectively. The speed estimation on an infrastructureless edge is done using the infrastructure edge similar to it. The intuition behind using this criteria is that the edges with similar congestion profile observe similar traffic conditions permitting reasonably accurate spatial extrapolation of speed.

The congestion profile of an edge is a 6-tuple specifying percentage of time the edge spends under each congestion level (A to F) and is computed using the historical data of congestion levels observed on the edge. As the congestion profile data follows a multinomial distribution, the Chi-Square test is found suitable for determining similarity between a pair of edges. A comprehensive survey of various similarity or distance measures can be found in [65].

Let the congestion profile of edge e_p and edge e_q be ϱ^{e_p} and ϱ^{e_q}, respectively. The Chi-Square variance is computed as,

$$\chi^2 = \sum_{j=A}^{F} \frac{\left(\varrho_j^{e_p} - \varrho_j^{e_q}\right)^2}{\varrho_j^{e_p}} \tag{3.19}$$

Degree of freedom is 5. If $\chi^2 < \chi^2_{th}$, the edges are considered similar. $\chi^2_{th} = 20.0$ is used in simulations. For example, if $\varrho^{e_p} = (15.0, 30.0, 30.0, 13.0, 10.0, 2.0)$ and $\varrho^{e_q} = (10.0, 25.0, 38.0, 10.0, 12.0, 5.0)$, the $\chi^2 = 10.23$. Hence, edge e_q is considered similar to e_p. For every edge e_i, the set of similar edges is recorded. Infrastructure edges are selected by computing K-coverset for all the edges using a greedy approximation algorithm [64].

The methodology is elaborated in Algorithm 3.5. $K = 3$ is used in simulations to ensure availability of three infrastructure edges for every infrastructureless edge.

Algorithm 3.5: Infrastructure Edge Selection for K-coverage of edges

Input: a set of candidate edges for having infrastructure, $E = \{e_1, e_2, \ldots, e_n\}$; K; and congestion profile of all the edges: $\varrho^{e_i}, \forall e_i \in E$

Output: the set of edges for infrastructure deployment: I

1: For every edge $e_i \in E$, determine the set of edges similar to it using congestion profiles.
2: Infrastructure Edge Set I={}
3: Count=$\{c_{e_1}, ..., c_{e_n}\}$, where c_{e_j} is number of times the edge e_j is covered by the infrastructure edges so far. Set $c_{e_j} = 0$, for $j = 1, \ldots, n$;
4: R is set of edges yet to be covered to fulfill K-coverage criteria; set R=$\{e_1, \ldots, e_n\}$
5: **repeat**
6: Select e_i from E such that it covers maximum number of e_j's in R.
7: I = I $\cup$ $\{e_i\}$; E = E $-$ $\{e_i\}$; R = R $-$ $\{e_i\}$
8: For each edge e_j covered by e_i, $c_{e_j} = c_{e_j} + 1$
9: For each $c_{e_j} \geq K$, remove e_j from R.
10: **until** R={}

The infrastructure requirement of ECOMO depends upon the number of edges in a road network and the traffic profile of edges. When the number of ITS infrastructure units, N, is specified, a variation of the Algorithm 3.5 is used to select N edges for infrastructure deployment (Algorithm 3.6): the algorithm tries to 1-cover all the edges using minimal infrastructure units, and repeats the process until all N infrastructure units are deployed.

Algorithm 3.6: Maximal Edge Coverage using Limited Infrastructure

Input: a set of candidate edges for having infrastructure, $E = \{e_1, e_2, \ldots, e_n\}$; N; and congestion profile of all the edges: $\varrho^{e_i}, \forall e_i \in E$

Output: the set of N edges for infrastructure deployment: I

1: For every edge $e_i \in E$, determine the set of edges similar to it using congestion profiles.
2: Infrastructure Edge Set I={}
3: R is set of edges yet to be covered at current coverage level; set R=$\{e_1, \ldots, e_n\}$
4: **while** number of edges in I <N **do**
5: Select e_i from E such that it covers maximum number of e_j's in R.
6: I = I $\cup$ $\{e_i\}$; E = E $-$ $\{e_i\}$; ; R = R $-$ $\{e_i\}$
7: For each edge e_j covered by e_i, remove it from R if it is present
8: If R={}, set R=$\{e_1, \ldots, e_n\}$ - I
9: **end while**

3.5.3 Using Infrastructure Edges for Speed Estimation

It is assumed that the infrastructure edges collect correct speed information, in addition to the flow, space occupancy and congestion information estimated using the cellular network, every minute. Every infrastructure edge learns occupancy-speed relationship using historical data.

$$S = a_0 + (a_1 \times \rho) + \left(a_2 \times \rho^2\right) + \epsilon \tag{3.20}$$

where S and ρ are speed and space occupancy, respectively; a_0, a_1, and a_2 are the regression model parameters; and ϵ is residual.

In COCOMO, an infrastructure edge learns the occupancy-speed relationship for every congestion level that it covers. Every infrastructureless edge gets the congestion level and space occupancy estimation from cellular network every minute. The space occupancy estimation of current time step is provided as an input to the regression model of the infrastructure edges associated with the congestion level for speed estimation.

In ECOMO, an infrastructure edge learns a regression model relating occupancy-speed data spanning across all the congestion levels. Every infrastructureless edge gets the congestion level and space occupancy estimation from cellular network every minute. The space occupancy estimation of current time step is provided as an input to the regression model of infrastructure edges similar to the edge for speed estimation. The congestion level along with the historical data of GPS probes is used in speed transition function (discussed next).

It is observed in our simulation results that the space occupancy estimations are biased in certain cases (for a pair of adjacent edges with and without traffic lights, the traffic light controlled edge gets under-estimate of space occupancy whereas the edge without traffic lights gets over-estimate of space occupancy). This leads to a certain amount of error when spatial extrapolation of speed is done from the infrastructure edge to the infrastructureless edge. Also, the vehicle speed on an edge is affected by many other parameters, e.g. edge length, presence of traffic lights at junction, number of connected edges, etc.. For example, the edge length and edge speed are highly correlated (Pearson's correlation coefficient >0.9) for the edges with traffic lights, according to our simulation results.

Instead of determining all these parameters and incorporating their effect on estimated

speed, historical data of GPS probe speed are used to compute speed transition function for a pair of edges for a given congestion level. The following additive speed transition function is used in this work to extrapolate the speed estimation from infrastructure edge i to edge j for congestion level c:

$$\delta_{i,j}^{c} = \overline{S_j^{g,c}} - \overline{S_i^{g,c}} \tag{3.21}$$

where, $\overline{S_j^{g,c}}$ is average GPS probe speed on edge j for congestion level c. After applying speed transition, K speed estimations, $S_m, m = 1, \ldots, K$ are available for an edge (due to K-coverage). The average of the K speed estimations, $\overline{S}$, along with the anticipated error is reported to the end user. The anticipated error, γ, is computed as

$$\gamma = \frac{\sum_{m=1}^{K} |\overline{S} - S_m|}{\sum_{m=1}^{K} S_m} \tag{3.22}$$

where $|\overline{S} - S_m|$ specifies the absolute value of the term. When all K infrastructure edges are not available, historical data of GPS probes is used to compute the anticipated error. For every infrastructure edge i that is used to estimate speed for edge j for given congestion level c, the historical data of GPS probe speed on i is transformed to the speed data on j using $\delta_{i,j}^{c}$:

$$\widehat{S}_{j,p}^{g,c} = S_{i,p}^{g,c} + \delta_{i,j}^{c}, \text{for each gps data point } p \tag{3.23}$$

Then, the anticipated error is computed as,

$$\gamma = \frac{\sum_{(p,q)\in P\times Q} |\widehat{S}_{j,p}^{g,c} - S_{j,q}^{g,c}|}{|P| \times |Q| \times \overline{S_j^{g,c}}} \tag{3.24}$$

where P and Q are the set of GPS probe speed data points on edge i and j, respectively; $|P|$ and $|Q|$ specifies the number of elements in set P and Q, respectively. There is a slight misuse of notation in the equation as $|\widehat{S}_{j,p}^{g,c} - S_{j,q}^{g,c}|$ in numerator represents the absolute value of the term. Note that the calculation of anticipated error is done using historical data of GPS probes and can be done offline. The actual speed estimation error depends upon location error and sampling rate of the cellular network data, temporal extrapolation error of flow and space occupancy data, spatial extrapolation, and speed

transition error.

Effect of using GPS probe data on speed estimation error: As mentioned earlier, an additive speed transition function $\delta^c_{i,j}$ (3.21) is used to spatially extrapolate speed estimation from infrastructure edge e_i to infrastructureless edge e_j for congestion level c. To examine its effectiveness in reducing speed estimation error, we take an example where speed estimation is attempted using a biased space occupancy measurement. It is assumed for simplicity that e_i and e_j have same static and dynamic parameters except for the fact that e_j has a positive bias in space occupancy estimation i.e. $\rho + \Delta\rho$, $\Delta\rho{>}0$ is measured instead of ρ. It is further assumed that the occupancy-speed relationship observed on edge e_i is linear (Figure 3.2(a)). When $\rho + \Delta\rho$ is provided as an input to the occupancy-speed model of e_i, the speed estimation $S - \Delta S$ is returned, showing the under-estimation of speed. Figure 3.2(b) shows GPS probe speed on edge e_i and e_j for space occupancy value $\rho+\Delta\rho$. The value of speed transition function is $\delta_{i,j} = S^g_j - S^g_i > 0$. The final speed estimation on e_j is $S - \Delta S + \delta_{i,j}$, where $\delta_{i,j}$ compensates for the under-estimation of speed improving accuracy.

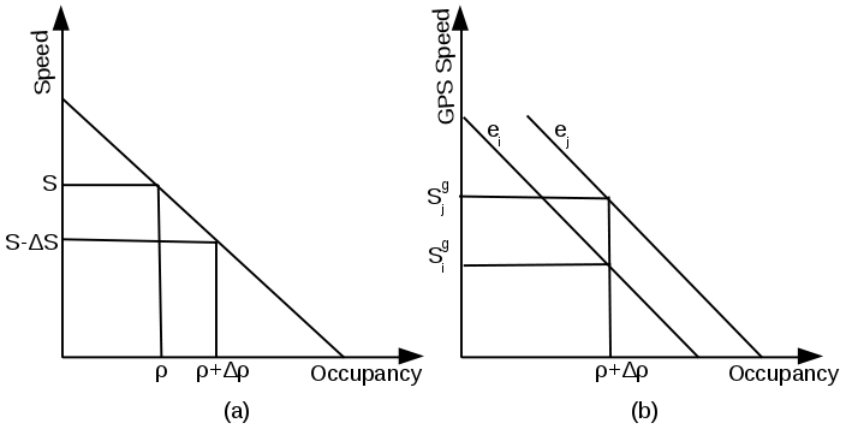


Figure 3.2: Occupancy-Speed Relationship

The above example illustrates the effectiveness of GPS probe speed data in reducing speed estimation error when the space occupancy estimate on the infrastructureless edge is biased. Here it needs to be clarified that the speed transition function is used to compensate for the cumulative speed error introduced by various known or unknown parameters (such as edge length, presence of traffic lights, etc.) including bias in space occupancy estimation. If the value of $\delta^c_{i,j}$ is closer to the difference between average speed of all the vehicles on e_i and e_j for congestion level c, the error in speed estimation is

minimized.

3.5.4 Error in Speed Estimation due to Space Occupancy Error

To simplify the error analysis, we assume a linear relationship between space occupancy and speed. A linear regression model learned on an infrastructure edge for occupancy-speed relationship is:

$$S = a_0 + a_1 \times \rho \tag{3.25}$$

where S and ρ are speed and space occupancy, respectively, and a_0 and a_1 are regression model parameters. On an infrastructure edge, correct speed, correct space occupancy, and estimated space occupancy (with measurement errors) are available. It is assumed that the error in space occupancy estimation is unbiased with mean $\overline{\epsilon_\rho} = 0$ and variance $\sigma^2_{\epsilon_\rho}$. A regression model can be learned using correct space occupancy and correct speed information or estimated space occupancy and correct speed information. On an infrastructureless edge, only estimated space occupancy is available for speed estimation.

- If correct space occupancy and correct speed information is used to learn occupancy-speed relationship, then the linear regression model is,

$$\begin{aligned} S &= a_0 + a_1 \times \rho \\ a_0 &= \overline{S} - a_1 \times \overline{\rho} \\ a_1 &= \frac{\sum_{i=1}^{n} (\rho_i - \overline{\rho}) \times \left(s_i - \overline{S}\right)}{\sum_{i=1}^{n} (\rho_i - \overline{\rho})^2} = \frac{cov(\rho, S)}{\sigma^2_\rho} \end{aligned} \tag{3.26}$$

  where $\overline{\rho}$ and σ^2_ρ are mean and variance in space occupancy, and $\overline{S}$ is mean speed. The space occupancy estimation on an infrastructureless edge may be biased (with respect to the space occupancy measurements used for learning the regression model on infrastructure edge). Let the input occupancy be $(\rho_j + \Delta\rho_j + \epsilon\rho_j)$, where $\Delta\rho_j$ and $\epsilon\rho_j$ are bias and error in the input occupancy, respectively. The speed estimation generated by the model is,

$$\begin{aligned} s_j &= a_0 + (\rho_j + \Delta\rho_j + \epsilon\rho_j) \times a_1 \\ &= \overline{S} - a_1 \times \overline{\rho} + (\rho_j + \Delta\rho_j + \epsilon\rho_j) \times a_1 \\ &= \overline{S} + (\rho_j - \overline{\rho}) \times a_1 + (\Delta\rho_j + \epsilon\rho_j) \times a_1 \end{aligned} \tag{3.27}$$

where $(\Delta\rho_j + \epsilon\rho_j) \times a_1$ is error in speed estimation due to occupancy error.

- If estimated space occupancy and correct speed information is used to learn occupancy-speed relationship, then the linear regression model is,

$$\begin{aligned} S &= a'_0 + a'_1 \times \rho \\ a'_0 &= \overline{S} - a'_1 \times \overline{\rho} \\ a'_1 &= a_1 \times \left(\frac{1}{1 + \frac{\sigma^2_{\epsilon_\rho}}{\sigma^2_\rho}} \right) = \frac{a_1}{1+q} \end{aligned} \tag{3.28}$$

The speed estimation generated by the model with input occupancy value $(\rho_j + \Delta\rho_j + \epsilon\rho_j)$ is

$$\begin{aligned} s'_j &= a'_0 + (\rho_j + \Delta\rho_j + \epsilon\rho_j) \times a'_1 \\ &= \overline{S} + (\rho_j - \overline{\rho}) \times a'_1 + (\Delta\rho_j + \epsilon\rho_j) \times a'_1 \\ &= \overline{S} + (\rho_j - \overline{\rho}) \times a'_1 + \frac{(\Delta\rho_j + \epsilon\rho_j) \times a_1}{(1+q)} \end{aligned} \tag{3.29}$$

where $\frac{(\Delta\rho_j + \epsilon\rho_j) \times a_1}{(1+q)}$ is error in speed estimation due to space occupancy error.

When speed estimation error of the two regression models is compared using simulations, it is observed that s'_j has less error than s_j in about 60% cases. As described earlier, a speed transition function computed using historical data of GPS probes is used to compensate for the error in speed estimation introduced by the space occupancy bias and other parameters. These mechanisms significantly improve accuracy of speed estimation.

3.5.5 Infrastructure Deployment and Speed Estimation when GPS Probe Data is Not Available

The vehicle speed on an edge is affected not only by space occupancy, but also by many other parameters, e.g. static parameters of an edge like number of lanes, edge length, presence of traffic lights, etc., and dynamic parameters like traffic condition on adjacent edges. The historical data of GPS probes are used to compute a speed transition function, that amalgamates effect of all these parameters, for extrapolating speed estimation from infrastructure edges to infrastructureless edges.

When GPS probe data is not available, spatial extrapolation of speed estimation needs to consider all the above parameters explicitly. The following four most influential static parameters of edges are identified that affect the vehicle movement speed on an edge: number of lanes, edge length, in-degree of the edge, and presence of exit traffic light on the edge. It is assumed that the information about these static parameters is available for all the edges in a road network.

The COCOMO is modified as follows to K-cover congestion levels occuring on all the edges. A congestion level occuring on an edge e_j is covered by an infrastructure edge e_i if (i) e_i covers the congestion level and (ii) e_i is similar to e_j as per the static parameters. The weighted cosine-similarity is used to compute the similarity score of a pair of edges. A 4-tuple of the static parameters of every edge is extracted from the digital map: (number of lanes, edge length, in-degree of the edge, presence of exit traffic lights on the edge). As the range of values for these parameters is different (e.g., number of lanes varies from 2 to 4, edge length ranges from 500 meters to 1800 meters, in-degree of edge takes value in the range of 1 to 5, and presence of traffic lights has value of 0 or 1), they are normalized using the maximum value of each parameter for the pair of edges. For example, if $A = (2.0, 1000.0, 3.0, 1.0)$ and $B = (3.0, 800.0, 2.0, 1.0)$, the corresponding normalized vectors are $\widehat{A} = (0.67, 1.0, 1.0, 1.0)$ and $\widehat{B} = (1.0, 0.8, 0.67, 1.0)$.

A different weight is associated with each parameter based on its influence on traffic movement. The weight 4-tuple used in our simulations is $W = (0.1, 0.1, 0.4, 0.4)$. These weights were tuned by carrying out a series of simulations. The in-degree of an edge and presence of traffic lights at the downstream end of an edge were found the most influencing parameters to the traffic movement on an edge, hence were assigned the highest weight of 0.4.

The weighted cosine-similarity between vectors $\widehat{A}$ and $\widehat{B}$, using weight vector W is defined as

$$SS(\widehat{A}, \widehat{B}, W) = \frac{\sum_i^n a_i \times b_i \times w_i}{\sqrt{\sum_i^n (a_i^2 \times w_i)} \sqrt{\sum_i^n (b_i^2 \times w_i}} \tag{3.30}$$

Here a_i, b_i, and w_i are elements of vectors $\widehat{A}$, $\widehat{B}$, and W, respectively, and $n = 4$. The Algorithm 3.7 specifies the procedure for K-covering congestion levels occuring on all the edges.

Algorithm 3.7: Infrastructure Edge Selection for K-coverage of Congestion levels considering Static Parameters of Edges

Input: a set of candidate edges for having infrastructure, $E = \{e_1, e_2, \ldots, e_n\}$; K; congestion profile of all the edges: $\varrho^{e_i}, \forall e_i \in E$; and static parameter information for all the edges

Output: the set of edges for infrastructure deployment: I

1: Determine the set of (edge, congestion level) pairs covered by every edge $e_i \in E$ using its congestion profiles ϱ^{e_i} and static parameter information for all the edges.
2: Infrastructure Edge Set I={}
3: Count=$\{c_j^l : j \in \{A, \ldots F\}, l \in E\}$, where c_j^l is number of times the (l, j) pair is covered by the infrastructure edges so far. Set $c_j^l = 0, \forall (l, j) \in E \times \{A, \ldots, F\}$;
4: R is set of (edge, congestion level) pairs yet to be covered to fulfill K-coverage criteria; set $R = E \times \{A, \ldots, F\}$
5: **repeat**
6: Select e_i from E such that it covers maximum number of (l, j) pairs in R.
7: I = I $\cup \{e_i\}$; $E = E - \{e_i\}$; $R = R - \{(e_i, j), \forall j \in \{A, \ldots, F\}\}$
8: For each (l, j) pair covered by e_i, $c_j^l = c_j^l + 1$
9: For each $c_j^l \geq K$, remove (l, j) from R.
10: **until** R={}

The speed estimation method is similar with the modified model except for the fact that the output of the regression model is considered as the final speed estimation (without applying speed transition function). Every infrastructureless edge gets congestion level and space occupancy estimation from cellular network every minute. The space occupancy estimation of current time step is provided as an input to the regression model of the infrastructure edges, that covers the congestion level and are similar to the infrastructureless edge, for speed estimation.

The ECOMO requires minor modifications to permit speed estimation without historical data of GPS probes. The Edges are selected for Infrastructure deployment with the following modification to step-1 of Algorithm 3.5: edge e_i covers edge e_j only if

- e_i and e_j are similar according to static parameters
- e_i and e_j are similar according to congestion profile

The speed estimation method is similar with the modified model except for the fact that the output of the regression model is considered as the final speed estimation (without applying speed transition function), avoiding the requirement of historical data of GPS probes.

It needs to be stated that the effect of dynamic parameters (e.g., traffic condition on adjacent edges) and bias in space occupancy estimation is not captured using the static parameters leading to increase in speed estimation error. Also, when sufficient number of covering infrastructure edges are not present, the anticipated error calculation is not possible without historical data of GPS probes.

Chapter 4

Simulation

4.1 Introduction

A simulation model enables abstraction of a real system and permits one to focus on interesting phenomena. At the same time, the results obtained using simulation study are eloquent and useful only if the simulation model closely represents the real world scenario. Many traffic simulators are available in the literature (e.g. SUMO, VISSIM, VanetMobiSim, PARAMICS, etc., to name a few) that permit microscopic traffic simulation [66].

Simulator for Urban MObility (SUMO) is an open source, highly portable, and microscopic road traffic simulation package designed to handle large simulation scenarios. SUMO includes many support applications that help in preparing detailed microscopic traffic simulation model. The road network parameters (e.g., number of lanes, length, permitted vehicle types, etc.), vehicle parameters (e.g., vehicle type, acceleration, length, maximum speed, etc.), vehicle route and flow parameters, right of way, traffic lights, etc., are highly configurable. Various car following and overtaking models are available as part of SUMO distribution. A fast openGL based graphical user interface, fast execution speed, interoperatibility with other applications at run time using Traffic Control Interface (TraCI), and edge level, vehicle level, and detector level outputs are the attractive features.

A major limitation of SUMO is that it enforces lane discipline (does not allow lane width sharing) on all vehicle movements. The road network of cities in developing coun-

tries carries heterogeneous traffic [67] and the road space is shared by vehicles of various kinds, such as two wheelers, auto rickshaws, cars, buses, trucks, to name a few. The presence of vehicles with narrow widths in the traffic stream greatly increases the capacity of roads. Narrow vehicles fill-in the lateral and longitudinal gaps between wide vehicles permitting more efficient utilization of road space than the homogeneous traffic.

To overcome the limitation of SUMO, SiMTraM (Simulation of Mixed Traffic Mobility) was developed by Transportation Research Group at IIT Bombay by adapting SUMO version 0.12 [68]. SiMTraM divides the lane width in a configurable number of strips and associates fixed number of strips with every vehicle type according to the vehicle width. This enables sharing of lane width by multiple small vehicles. A major known bug in SiMTraM is, it crashes while running in certain cases, specifically while simulating a large number of vehicles. Some of these bugs are inherited from SUMO and are already dealt with in recent versions of SUMO.

Due to limitations of existing simulators, we redeveloped SiMTraM using SUMO version 0.17, which was the latest version of SUMO at the time of development, and made it available as open source to the user community through web [69]. The SUMO-0.17 and upgraded SiMTraM were compared with respect to the edge level vehicle flow, speed and flow-speed relationship, for various traffic and road network scenarios [70]. The simulation results show that upgraded SiMTraM permits higher vehicle density with more realistic vehicle movement speed.

However, the upgraded version of SiMTraM crashes with simulation scenarios involving city wide large road networks and more than twenty thousand vehicles. Hence, it was decided to use SUMO with appropriate parameter tuning for the present study. It is our sincere belief that SiMTraM has potential to support large scale simulations of lane sharing heterogeneous traffic as it is adapted from SUMO which is known to support large scale simulations. The task of fixing bugs in SiMTraM is kept as future work.

As per Ahmedabad Mobility Report, 2012 [18][19], the vehicle distribution on Ahmedabad city roads consists of 76.23% two wheelers, 5.27% three wheelers, 13.36% four wheelers, and about 5% buses and trucks. A similar vehicle distribution is observed in most of the cities in India and other Asian countries. Table 4.1 shows vehicle distribution and other parameters used in all the simulations in this work. The vehicle speed, acceleration, and minimum gap parameters are tuned and finalized during simulation model validation

Table 4.1: Vehicle Distribution

Vehicle type	% of total vehicles	Maximum speed (m/s)	Acceleration (m/s^2)	Length (m)	Minimum gap(m)	Static PCU [71]
2-wheelers-class1	15	08.75 m/s (32 km/hr)	1.2	2.0	1.0	0.5
2-wheelers-class2	45	11.11 m/s (40 km/hr)				
2-wheelers-class3	15	15.00 m/s (54 km/hr)				
3-wheelers-class1	1	07.80 m/s (28 km/hr)	0.7	3.0	2.0	1.5
3-wheelers-class2	3	09.72 m/s (35 km/hr)				
3-wheelers-class3	1	11.11 m/s (40 km/hr)				
Car-class1	3	09.72 m/s (35 km/hr)	1.5	4.0	2.5	1.0
Car-class2	9	11.11 m/s (40 km/hr)		4.5		
Car-class3	3	15.27 m/s (55 km/hr)		5.0		
Bus-class1	3	08.35 m/s (30 km/hr)	0.8	12.0	3.5	3.0
Bus-class2	2	12.50 m/s (45 km/hr)				

(section 4.2).

The car following and overtaking model in a microscopic traffic simulator formally define interactions of a vehicle with other vehicles in a road network (how one vehicle follows another vehicle in an uninterrupted flow, and rules for lane changing and overtaking). The Krauss car following model [72] available in SUMO is used in all the simulations in this work. The model is computationally efficient with a small set of equations defining lane disciplined safe movement and overtaking conditions for a vehicle. The model permits custom specification of parameters such as acceleration, deceleration, sigma (driver imperfection), tau (driver reaction time), and minimum gap for every vehicle type. Olstam and Tapani [73] present a detailed comparison of car following models used in AIMSUN, MITSIM, Paramics and VISSIM microscopic simulators.

Table 4.2 shows the simulation scenarios used to examine the performance of the proposed models. The CG road scenario (S1), 132-feet ring road scenario (S2), and

Table 4.2: Simulation Scenarios

Scenario Name	Description	Location error	Vehicle Insertion Rate
CG road (S1)	a 3.7 km road stretch from Navrangpura to Shastri Marg on CG road in Ahmedabad, India with one traffic light controlled junction	250m, 500m	1800-4200 veh/hr for peak and off peak duration of 14 hrs (total 38400 vehicles)
132 feet ring road (S2)	a 4.8 km road stretch from Akhabar nagar to Andhajan Mandal on 132 feet ring road in Ahmedabad, India with one traffic light controlled junction	250m, 500m	1800-5800 veh/hr for peak and off peak duration of 14 hrs (total 47200 vehicles)
Naroda-Narol highway (S3)	a 17.5 km road stretch from Naroda to Narol on National Highway 8-c in Ahmedabad, India with no traffic lights	250m, 500m	Nine merging/diverging flows with 200-900 veh/hr per flow for 14 hrs (total 72590 vehicles)
Grid network (S4)	a $15km \times 15km$ road network with road segment length of 1km each, with traffic lights at each junction	250m, 500m	2100-4200 veh/hr in seven merging/diverging flows for peak and off peak duration of 14 hrs (total 40950 vehicles)
Random network (S5)	a $18km \times 16km$ road network with road segment length of 0.5-1.5km each, with traffic lights at each junction	250m, 500m	2400-4800 veh/hr in eight merging/diverging flows for peak and off peak duration of 14 hrs (total 46800 vehicles)

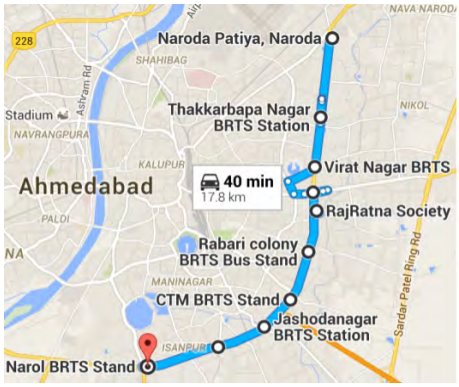
Naroda-Narol highway scenario (S3) are validated using real traffic data taken from [74] (for S1 and S2) and [18] (for S3). The details of simulation model validation are described in section 4.2. The two city wide large road network scenarios, namely Grid network (S4) and Random network (S5), generated using NETGEN utility of SUMO, are used for large scale evaluation of the proposed models. Figure 4.1 shows a snapshot of the road network scenarios. The road network snapshot for S1, S2 and S3 are extracted from Google maps and that for S4 and S5 are taken from SUMO. All data processing is done using Python version 2.7.3 scripts.

4.2 Simulation Model Validation

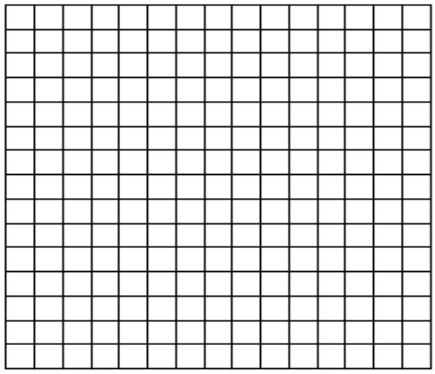
SUMO supports traffic heterogeneity by permitting different types of vehicles in a road network. However, it enforces strict lane discipline and does not allow multiple vehicles

(a) CG Road: S1 (3.7 km)

(b) 132 feet Ring Road: S2 (4.8 km)



(c) Naroda-Narol Road: S3 (17.5 km)



(d) Grid Network: S4 (15km×15km)

(e) Random Network: S5 (18km×16km)

Figure 4.1: Simulation Scenarios

to share the lane width. As already mentioned, in India and specifically in Ahmedabad city, more than 75% of vehicles moving in the road network are two-wheelers [18][19] and they share the lane width.

This section examines feasibility of using SUMO for representing Indian traffic scenario. We claim using the following few representative scenarios that, with proper parameter tuning, SUMO can approximate lane width sharing Indian traffic.

4.2.1 CG Road and 132-feet Ring Road Scenario

Varmora et al. [74] conducted traffic surveys on CG Road and 132 feet ring road in Ahmedabad, India using video camera. CG Road is a 2-lane wide arterial road which carries heavy traffic, specifically during peak hours. The 132-feet ring road is a 3-lane wide arterial road which was developed to offload the traffic from core arterial roads of the city. The data collection was done between Girish coldrink and Jade Blue on CG road and between Shastri nagar Cross roads and Jaymangal on 132 feet ring road (indicated with an arrow in Figure 4.1a and 4.1b, respectively). The authors manually processed the video data to generate traffic flow (Passenger Car Units (PCU)/Hr) and speed information. Table-4.3 shows the vehicle flow and speed data collected on these two streets. Aim is to generate the similar vehicle flow-speed relationship using SUMO simulator.

Table 4.3: Traffic Flow - Speed Data of CG Road and 132-feet Ring Road

CG Road (2-Lane)		132-feet Ring Road (3-lane)	
PCU/hr	Speed (km/hr)	PCU/hr	Speed (km/hr)
1000	36.0	1500-2000	42.5
1500-2200	32.5	2500	40.0
2500-2800	30.0	3000-4000	35.0

CG Road network configuration: A 3.7 km road stretch on CG road is simulated (Figure 4.1a). To generate the similar flow-speed relationship as mentioned in Table-4.3, we selected the road network parameters and traffic flow parameters as mentioned in Table-4.4. For simulating a 2-lane wide CG road scenario, we required three heterogeneous lanes (one slower lane) in simulation. The reason being, on Indian roads there are a large fraction of small vehicles and they share the road space more effectively. Also, lane width sharing increases entropy on the road, leading to lower movement speed. Table-4.5 shows actual and simulated flow-speed readings on the measurement edge (indicated with an arrow in Figure 4.1a). When the actual flow increases from 1000 PCU/hr to 2786 PCU/hr, the average vehicle speed decreases from 36 km/hr to 30 km/hr; whereas in SUMO, when the simulated flow increases from 950 PCU/hr to 2700 PCU/hr, the average vehicle speed decreases from 34.9 km/hr to 27.8 km/hr. These results show a good match between the actual and simulated measurements.

132 feet ring road network configuration: A 4.8 km road stretch on 132 feet

Table 4.4: CG Road Network Simulation Parameters

Parameter	Value	Remarks
Number of Lanes	3	Two major-lanes have speed limit of 12 m/s. The third slower lane has speed limit of 8 m/s. On slower lane only 2-wheelers and 3-wheelers are permitted.
Number of Junctions	5	One-traffic-light-controlled junction (GirishColdrink: cycle time=70 s, green time=31 s)
Vehicle Generation	1250-4200 veh/hr	Vehicles were generated for three hours as per vehicle distribution mentioned in Table-4.1

Table 4.5: CG Road Network Simulation Results

Actual		Simulated	
PCU/hr	Speed (km/hr)	PCU/hr	Speed (km/hr)
1000	36.0	950-1200	34.9 -34.3
1500-2200	32.5	1500-2250	33.3 - 30.8
2500	30.0	2400-2550	30.2-29.3
2786	30.0	2700	27.8

ring road is simulated (Figure 4.1b). To generate the same flow-speed relationship as mentioned in Table-4.3, we selected the road network parameters and traffic flow parameters as mentioned in Table-4.6. For simulating a 3-lane wide 132 feet ring road scenario, we required four lanes (one slightly slower lane) in simulation. Table-4.7 shows actual and simulated flow-speed readings on the measurement edge (indicated with an arrow in Figure 4.1b). When the actual flow increases from 1500 PCU/hr to 4000 PCU/hr, the average vehicle speed decreases from 42.5 km/hr to 35 km/hr; whereas in SUMO, when the simulated flow increases from 1350 PCU/hr to 3980 PCU/hr, the average vehicle speed decreases from 42.1 km/hr to 37.6 km/hr. These results show a good match between the actual and simulated measurements.

Table 4.6: 132-feet Ring Road Network Simulation Parameters

Parameter	Value	Remarks
Number of Lanes	4	Three major-lanes have speed limit of 15.7 m/s. The fourth lane has speed limit of 13.89 m/s. On this lane only 2-wheelers and 3-wheelers are permitted.
Number of Junctions	4	One traffic-light-controlled junction (Shastri junction: cycle time=72 s, green time=31 s)
Vehicle Generation	1000-5800 veh/hr	Vehicles were generated for three hours as per vehicle distribution mentioned in Table-4.1

Table 4.7: 132-feet Ring Road Network Simulation Results

Actual		Simulated	
PCU/hr	Speed (km/hr)	PCU/hr	Speed (km/hr)
1500-2200	42.5	1350-2250	42.1 - 40.7
2500	40	2550	40.4
3000-4000	35	2850-3980	39.9-37.6

4.2.2 Naroda-Narol Highway Scenario

The Naroda-Narol road is a 17.5 km road stretch on National Highway 8-c in Ahmedabad, India (Figure 4.1c). It also carries heavy local traffic during peak hours. There are 8 junctions on the road stretch and all the edges are three lane wide. The flow data at each junction on the road stretch is taken from [19]. The edge level speed information is not available for the road stretch. Instead, the Average trip speed information for the whole road stretch is obtained from [18]. It is 35.4 km/hr during off peak hours and 29 km/hr during peak hours. The aim is to generate the similar vehicle flow at each junction while ensuring similar average trip speed for the whole road stretch.

To approximate the actual flow at each edge on the road stretch and to match with the average trip speed, we selected the road network parameters and traffic flow parameters as mentioned in Table-4.8. For simulating 3-lane wide Naroda-Narol road scenario, we required four lanes (one slightly slower lane) in simulation. The set up lead to average trip speed of 38.8 km/hr in simulation. Table-4.9 shows edge wise actual and simulated flow. A very good match is observed between the actual and simulated vehicle flow on all nine edges (average error of 1.03%). The difference in actual average trip speed (35.4 km/hr) and simulated trip speed (38.8 km/hr) is 9.6%.

4.2.3 Discussion

It is observed in simulations that the strict lane discipline enforced by SUMO reduces the traffic carrying capacity of a road network. Hence, to match with the vehicle flow of lane width sharing Indian traffic, an additional lane is required in the SUMO road network. Due to the horizontal movement of vehicles in lane width sharing traffic, the average vehicle speed in the road network decreases. Hence, the appropriate lane speed adjustments are required in SUMO road network. As a part of this exercise, the vehicle

Table 4.8: Naroda-Narol Road Network Simulation Parameters

Parameter	Value	Remarks
Number of Lanes	4	Three major-lanes have speed limit of 15.70 m/s. The fourth lane has the speed limit of 13.89 m/s. On this lane only 2-wheelers and 3-wheelers are permitted.
Number of Junctions	8	Three traffic-light-controlled junctions (Virat Nagar: cycle time=87s, green time=46s; Soni Chal: cycle time=87s, green time=46s; Rabari colony: cycle time=116s, green time=75s;)
Vehicle Generation	total 15500 vehicles	as the vehicles were permitted to enter and exit the road stretch at any junction, nine merging-diverging flows with 200-900 veh/hr per flow were generated for three hours as per vehicle distribution mentioned in Table-4.1

Table 4.9: Naroda-Narol Road Network Simulation Results

Edge Id	Edge Length (km)	Actual PCU/hr	Simulated PCU/hr	Simulated Edge Speed (km/hr)
e1	2.9	1526	1525	41.4
e2	2.1	1526	1530	38.2
e3	2.0	2016	1914	37.3
e4	1.8	3049	3039	39.3
e5	1.5	3049	3043	32.1
e6	1.2	3852	3894	37.7
e7	2.0	3560	3507	38.8
e8	1.5	1987	1977	41.2
e9	2.5	1633	1632	41.5

parameters such as maximum speed and acceleration are tuned for use in all the simulation scenarios (Table 4.1).

An adequate match is observed between actual and simulated traffic parameters in all three scenarios. The simulation results show that, with appropriate parameter tuning, SUMO is able to approximate vehicle flow-speed relationship observed in lane width sharing Indian traffic. Hence, it is concluded that SUMO can be used to simulate lane width sharing Indian traffic scenarios.

4.3 Performance of the Map Matching Algorithm

Table 4.10 and 4.11 present result of path estimation and trip speed estimation for a sample of five vehicles for grid network (S4) and random network (S5) scenarios. The

Table 4.10: Performance of Map Matching Algorithm: Grid Network (S4)

Veh. Id	Path	Trip Speed (m/s)	250m Location error			500m Location error		
			L1*	L2*	L3*	L1*	L2*	L3*
v0	12 edges with no U-turn	11.76	12/12	12.01	02.13	10/12	11.01	06.38
v1	12 edges with no U-turn	11.76	12/12	12.01	02.13	10/12	11.01	06.38
v2	5 edges with no U-turn	10.42	05/05	11.01	05.66	04/05	10.16	02.50
v3	6 edges with U-turn at one end	12.50	05/06	11.01	11.92	03/06	09.90	20.80
v4	12 edges with U-turn at one end	10.81	11/12	10.09	06.66	11/12	10.09	06.66
*L1=accuracy of path estimation defined as ratio of correctly identified edges to total number of edges on a path, L2=estimated trip speed (m/s), L3= speed error (%)								

Table 4.11: Performance of Map Matching Algorithm: Random Network (S5)

Veh. Id	Path	Trip Speed (m/s)	250m Location error			500m Location error		
			L1*	L2*	L3*	L1*	L2*	L3*
v0	16 edges with no U-turn	10.80	16/16	11.01	01.94	15/16	10.54	02.41
v1	13 edges with no U-turn	11.03	13/13	11.26	02.09	11/13	10.04	08.98
v2	7 edges with U-turn at one end	10.26	06/07	09.62	06.23	03/07	06.69	34.80
v3	8 edges with U-turn at one end	11.49	07/08	10.61	07.65	06/08	09.22	19.76
v4	6 edges with U-turn at both ends	11.29	03/06	06.77	40.03	not found	–	–
*L1=accuracy of path estimation defined as ratio of correctly identified edges to total number of edges on a path, L2=estimated trip speed (m/s), L3= speed error (%)								

following observations are made from simulation results: intermediate edges of all the paths are determined accurately in both the scenarios (S4 and S5) with maximum location error of 250 meters and 500 meters. Due to conservative selection of edges, the algorithm has no false positive, i.e. the algorithm never claims an incorrect edge on a path. However, due to limited overlapping location points at the beginning and the end of a trip, the algorithm faces difficultly in selecting edges correctly. In that case, it discards one or more edges from the head and tail of the path and computes a truncated path. The

number of truncated edges increases with the increase in location error. When a vehicle takes U-turn at the beginning (or end) of the journey, the algorithm always discards the first (or last) edge from the estimated path. With the maximum location error of 250 meters, when a vehicle does not take U-turn, the path is estimated with 100% accuracy in both, S4 and S5, scenarios.

When path estimation is more than 85% accurate, the error in trip speed estimation is less than 12% in both the scenarios (S4 and S5).

4.4 Estimation of Traffic parameters using Cellular Network Data

For evaluating the performance of the proposed traffic parameters estimation model, the following methodology is used: every scenario (Table 4.2) is simulated three times by changing random seed. To use only steady state data of simulation runs, the data from the first and the last edge of all the routes, the initial 30 minutes data, and the data after 14 hrs of simulation are discarded. The aggregation period of 10 minutes (the raw data of 10 minutes is used to compute traffic parameters) and update period of 1 minute (traffic parameters are computed every minute) are used.

4.4.1 Vehicle Flow Estimation

Figure 4.2 shows mean error in non-real time flow estimation (before temporal extrapolation) and real time flow estimation (after temporal extrapolation using exponential moving average and regression, each) for all the simulation scenarios with maximum location error of 250m and 500m. Figure 4.3 shows percentile error in flow estimation in all the scenarios. The regression based temporal extrapolation is done using leave-one-out cross validation technique (historical data of two days is used for learning and one day data is used for testing). The variations in flow error are denoted by error bars in the graph (Figure 4.2). The height of an error bar equals the standard deviation in flow error.

The non-real time flow estimation is very accurate in all the scenarios with mean and ninety percentile error of less than 5%. We identify the following two reasons for this high accuracy: first, the map-matching algorithm is effective in generating accurate

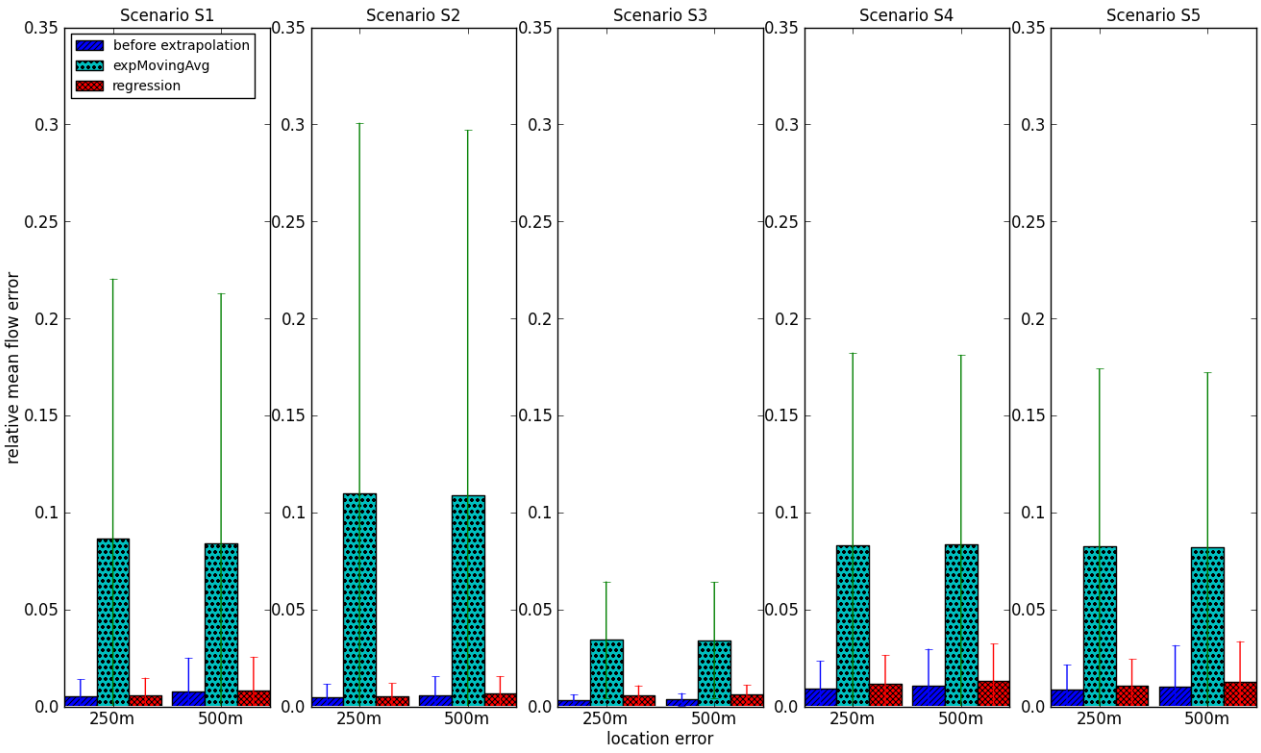
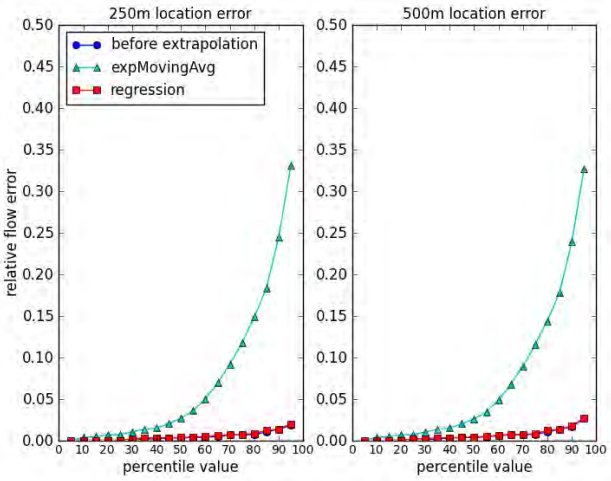


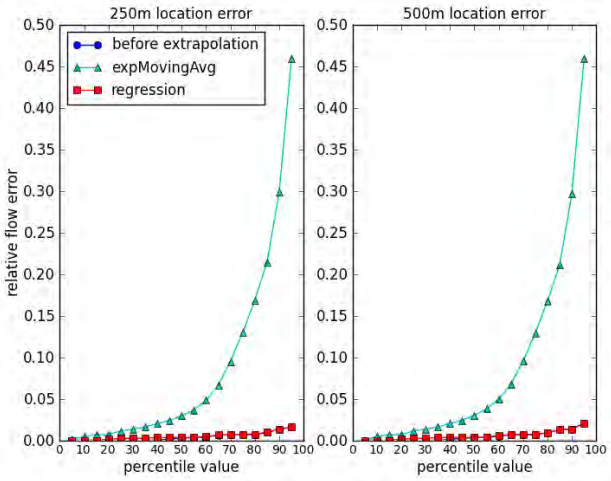
Figure 4.2: Mean Flow Error

vehicle trajectories; and second, reasonably large aggregation period (10 minutes) is used for computing vehicle flow data. Also, a little impact of location error is observed on flow error. This is due to two reasons: first, the ambiguous period (when a vehicle is detected on multiple edges) introduced due to location error is comparatively smaller than the aggregation period (a vehicle traveling at 30 km/hr speed may face the ambiguous period of one minutes with the maximum error of 500m, in worst case), and second, the ambiguous period must span across the aggregation period to contribute to the flow error. Section 3.3.3 contains detailed discussion on effect of location error and aggregation period on flow estimation error.

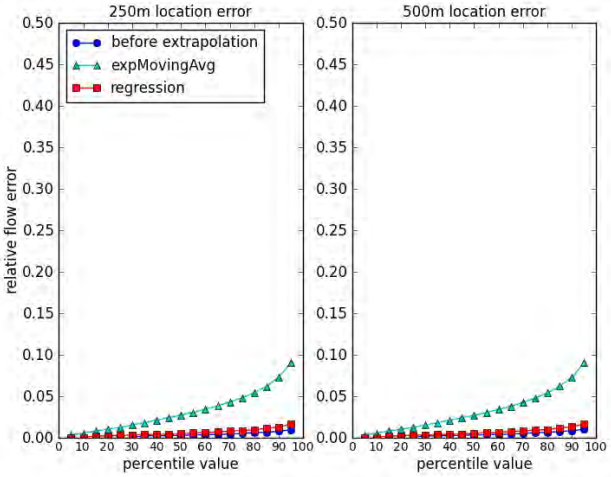
The regression based temporal extrapolation is equally accurate with the mean flow error of less than 5%. This alludes to the fact that regression based extrapolation does not contribute any error in real time flow estimation. The reason being, the traffic flow variations during peak hours and off-peak hours in the three days simulation data are similar, representing recurrent traffic conditions in a road network. The regression model learnt using the two days flow data fits well when tested using the third day data, leading to accurate extrapolation of vehicle flow.



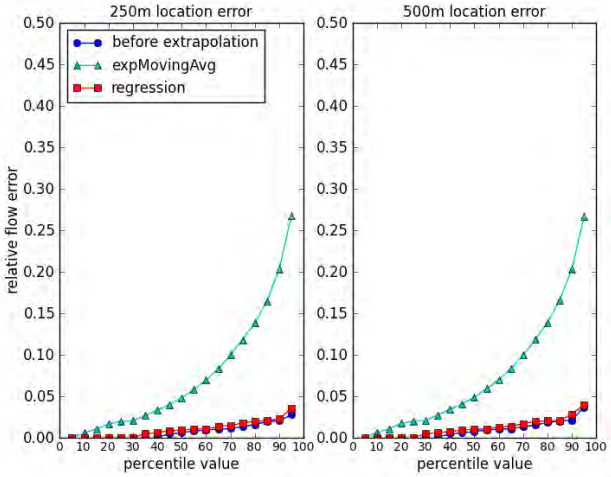
(a) Scenario S1



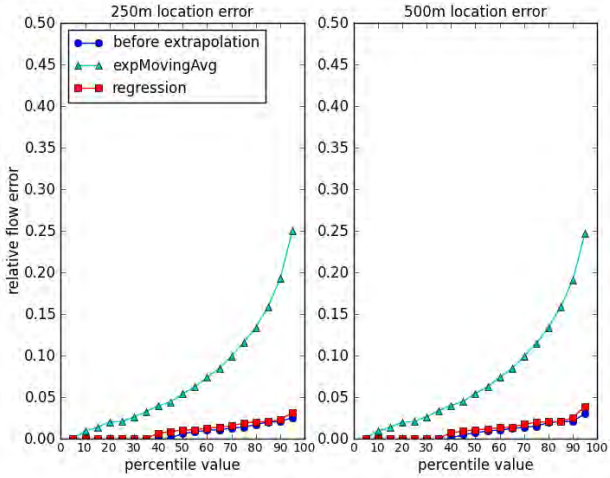
(b) Scenario S2



(c) Scenario S3



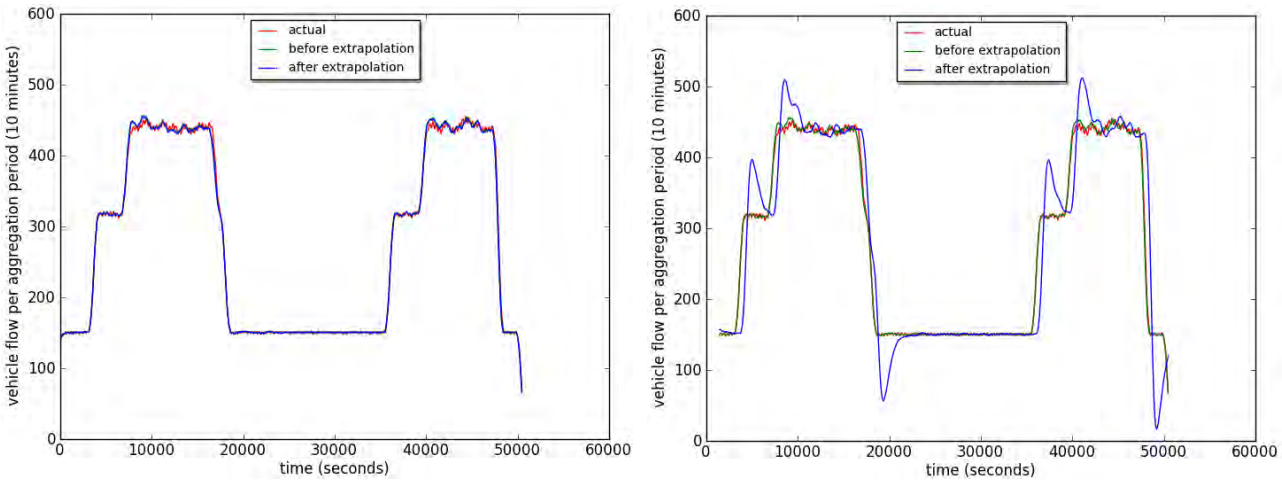
(d) Scenario S4



(e) Scenario S5

Figure 4.3: Percentile Flow Error

The exponential moving average based temporal extrapolation is relatively erroneous with mean flow error approaching 10%. It is observed that there is a huge difference between mean and median flow error (mean flow error is approximately double than the median flow error) indicating the presence of bias in the flow error. The detailed analysis revealed some interesting facts. The vehicle flow on edges varies with time and gets more noticeable transitions from off-peak hour to peak hour and vice versa. The exponential moving average based temporal extrapolation gives more weightage to recent history to estimate the vehicle flow after h time steps (h=10 minutes in this case). During steady flow periods, the extrapolated flow values are in very good agreement with the actual flow value. During the transition periods, the variations in flow are inflated leading to increase in flow error. For example, if the vehicle flow has been increasing recently, the exponential moving average method predicts increase even after the flow stabilizes.



(a) Flow Variations with Regression based Extrapolation (b) Flow Variations with Exponential Moving Average based Extrapolation

Figure 4.4: Flow Variations on an Edge in Scenario S2

To support the above arguments, Figure 4.4 shows actual and estimated flow (before and after extrapolation using exponential moving average and regression) on an edge in 132-feet ring road scenario (S2). It can be seen that the actual and estimated flow (before and after extrapolation) are in good agreement during steady flow periods regardless of the method used for extrapolation. The regression based extrapolation (Figure 4.4a) captures flow variations well and makes accurate flow estimation even during transition periods (peak hours to off-peak hours and vice versa). The exponential moving average based extrapolation (Figure 4.4b) does not perform well during transition periods and

the increase or decrease in flow is overshot.

The least flow error among all the simulation scenarios is observed in Naroda-Narol highway scenario (S3), specifically with exponential moving average based extrapolation (less than 4%). This is due to two reasons: first, there are no sharp transitions of peak hour and off peak hour traffic; and second, there are no traffic light controlled junctions permitting relatively smooth (uninterrupted) traffic movement.

4.4.2 Edge Space Occupancy Estimation

The edge space occupancy is computed using vehicle flow and count data. Figure 4.5 shows mean error in non-real time space occupancy estimation (before temporal extrapolation of flow) and real time space occupancy estimation (after temporal extrapolation of flow using exponential moving average and regression, each) for all the simulation scenarios with maximum location error of 250m and 500m. Figure 4.6 shows percentile error in space occupancy estimation in all the scenarios. The variations in space occupancy error are denoted by error bars in the graph (Figure 4.5). The height of an error bar equals the standard deviation in space occupancy error.

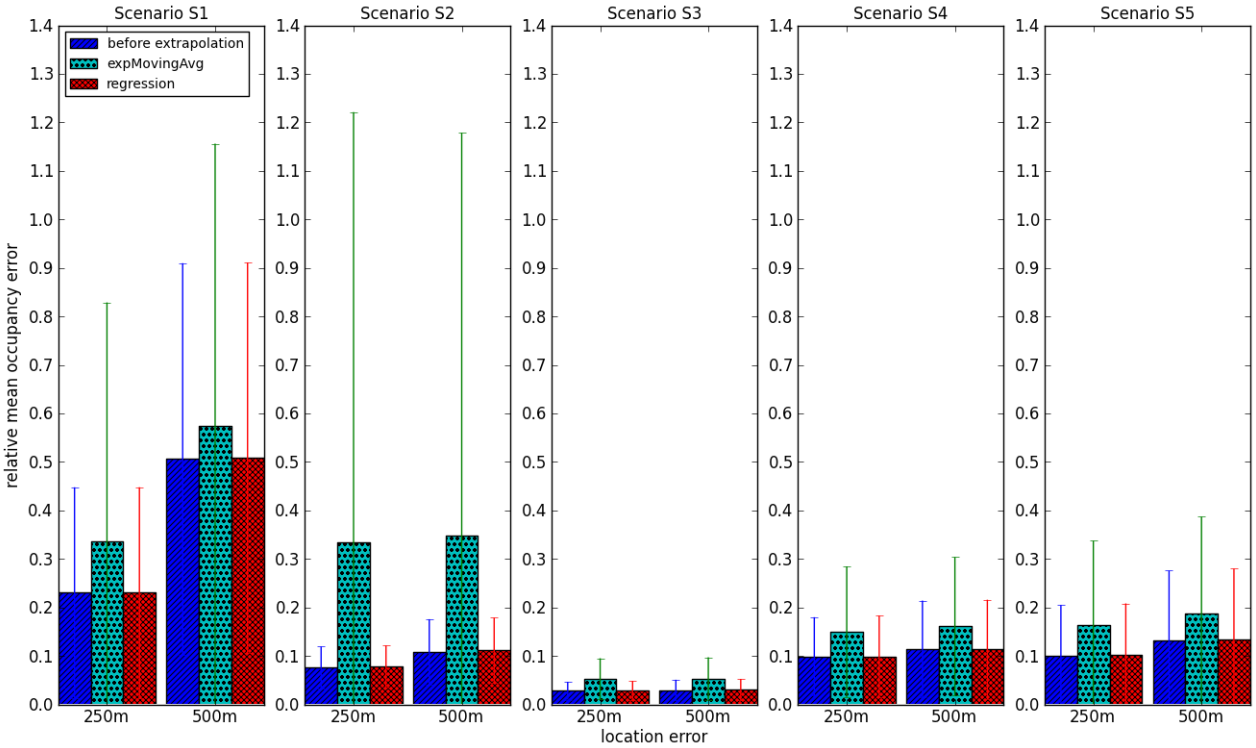
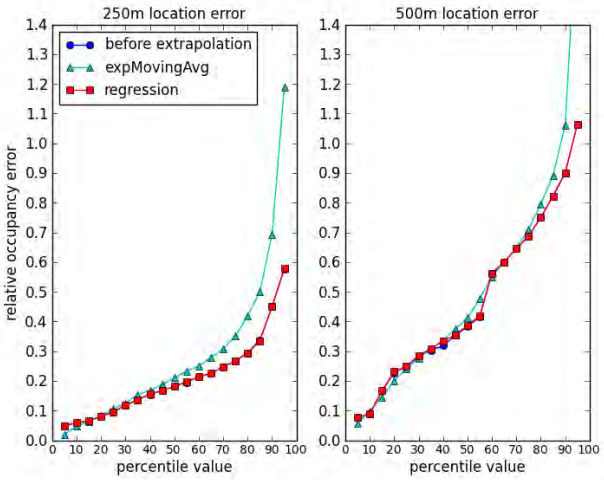
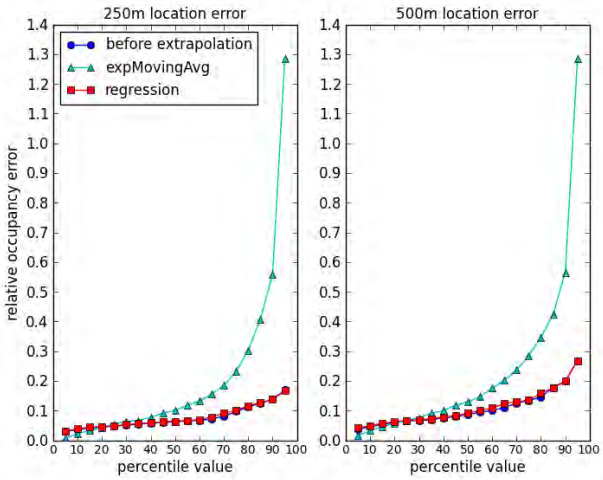


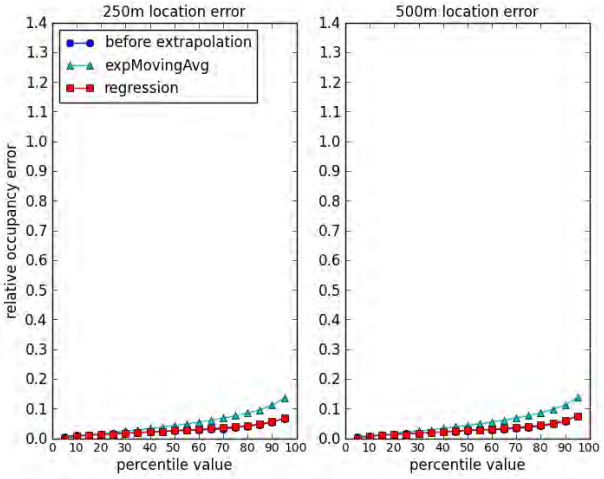
Figure 4.5: Mean Space Occupancy Error



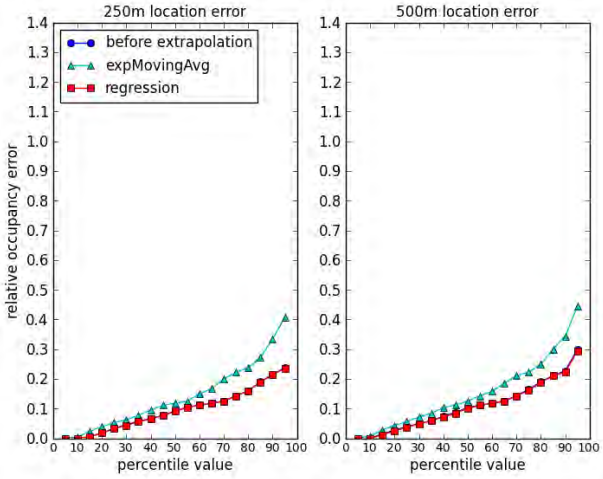
(a) Scenario S1



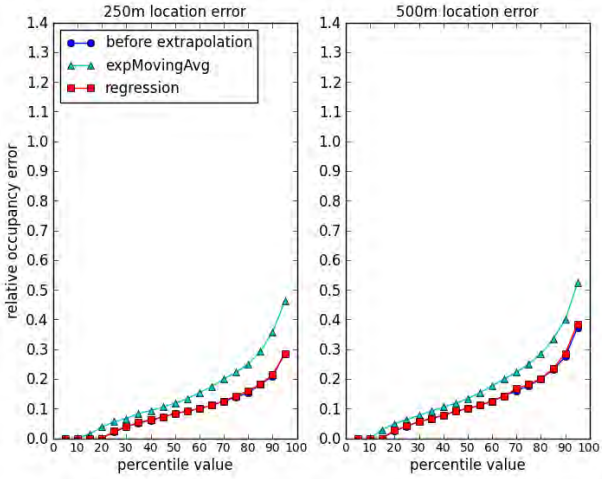
(b) Scenario S2



(c) Scenario S3



(d) Scenario S4



(e) Scenario S5

Figure 4.6: Percentile Space Occupancy Error

The error in non-real time space occupancy estimation is significantly higher (more than double) than flow error. This is due to the fact that flow estimates are computed for an aggregation period (10 minutes) reducing flow error significantly. The space occupancy estimates, on the other hand, are instantaneous and specify edge space occupancy at the end of an aggregation period. Except for CG road scenario (S1), the mean and median error in space occupancy estimation is less than 10%. In S1, the mean and median space occupancy error is 23% and 18%, respectively, for 250m location error scenario, and more than 50% and 38%, respectively, for 500m location error scenario. The detailed analysis of simulation results revealed some interesting facts. S1 has junctions with and without traffic lights. The vehicles' movement on traffic light controlled edges is different than the edges without traffic lights. As the traffic light deployment information is not available (realistic assumption in India), the map matching algorithm generates erroneous vehicle-edge mapping. It is observed that the space occupancy estimation is biased. For a pair of adjacent edges with and without traffic lights, the traffic light controlled edge gets under-estimate of space occupancy whereas the edge without traffic light gets over-estimate of space occupancy. It needs to be clarified that the similar effect is not observed in flow estimation error due to large aggregation period. The Pearson correlation coefficient between actual space occupancy and estimated space occupancy is more than 0.95 for all the edges in the road network. This implies that the space occupancy estimation data can be conveniently used in subsequent stages (e.g. edge level speed estimation) in the proposed model.

As mentioned earlier, the real time and non-real time space occupancy estimation is computed using corresponding vehicle flow and count data. The accuracy of non-real time flow data and real-time flow data computed using regression model are comparable. Hence, the error in non-real time space occupancy data and real time space occupancy data computed using regression based flow data are analogous. Due to high error in real time flow data computed using exponential moving average based extrapolation, the corresponding space occupancy estimation has high error (about three times higher than non-real time occupancy estimation error in S2, about two times higher in S3, and about one and half times higher in S4 and S5).

The least occupancy error among all the simulation scenarios is observed in Naroda-Narol highway scenario (S3), specifically with exponential moving average based extrap-

olation (less than 6%). The reason lies in the fact that the flow estimation in the scenario is very accurate.

4.4.3 Traffic Congestion Estimation

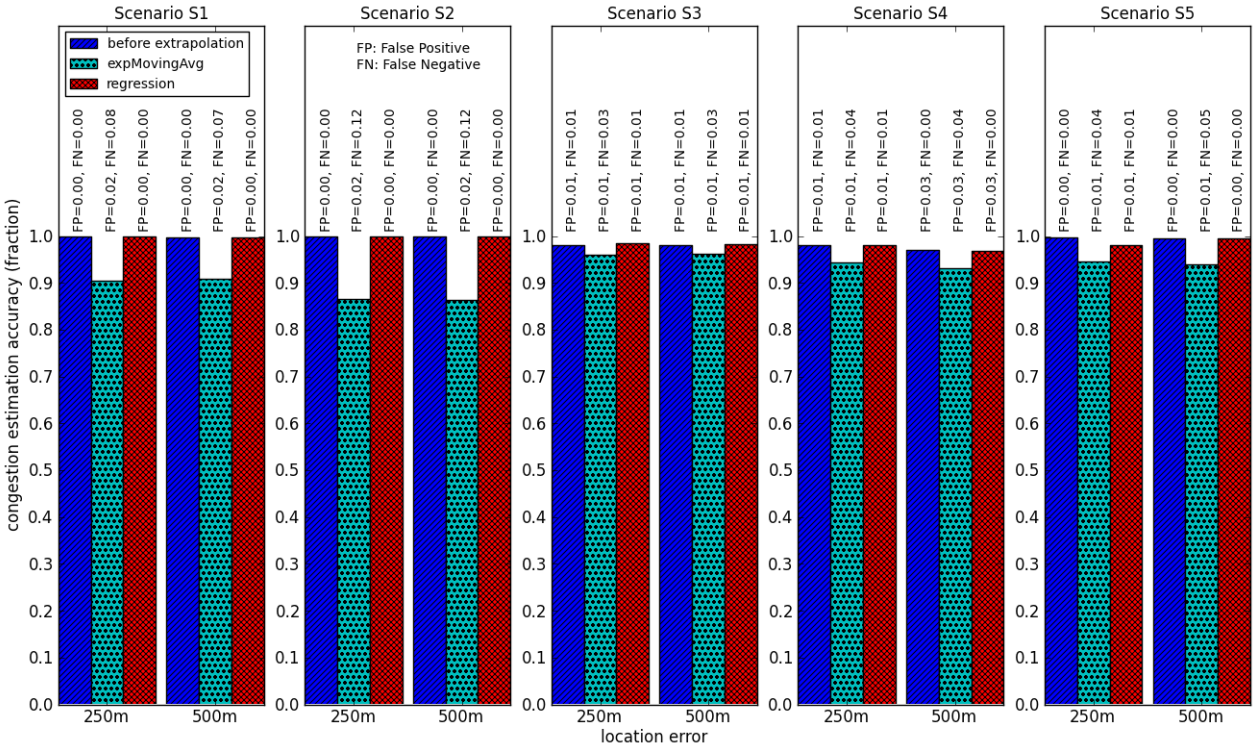


Figure 4.7: Congestion Estimation Accuracy

Traffic condition on an edge during an aggregation period is classified among six congestion levels (A-F). The volume to capacity ratio (V/C ratio) is used to compute congestion level on an edge (section 3.3.2 elaborates the mechanism). Figure 4.7 summarizes the accuracy of non-real time congestion level estimation (before temporal extrapolation of flow) and real time congestion level estimation (after temporal extrapolation of flow using exponential moving average and regression, each) for all the simulation scenarios with maximum location error of 250m and 500m. The fraction of false positives and false negatives are also computed for all the scenarios. A classification result is considered false positive if the estimated congestion level is higher than the actual congestion level on an edge; similarly, a false negative is recorded if the estimated congestion level is lower than the actual congestion level.

The non-real time congestion estimation is very accurate in all the scenarios with

maximum error of 3%. There are two reasons for this high accuracy: first, the vehicle flow data used for congestion level computation is accurate (mean and ninety percentile error of less than 5%); and second, to avoid misclassification at the boundary value, range estimation of the V/C ratio with $\epsilon = 0.1$ is used (elaborated in section 3.3.2), reducing false positives and false negatives significantly.

As mentioned earlier, the accuracy of non-real time flow data and real time flow data computed using regression model are comparable. Hence, the error in non-real time congestion data and real time congestion data computed using regression based flow data are analogous, showcasing accurate real time congestion estimation.

Table 4.12: Flow Error and Congestion Level Classification Error with Exponential Moving Average

Scenario	Congestion Error	Flow Error
S1	09-10%	<09%
S2	14%	<12%
S3	04%	<04%
S4	05-07%	<09%
S5	05-06%	<09%

The congestion level estimation using real time flow data computed using exponential moving average based temporal extrapolation is comparatively more erroneous. The reason being, the flow error with exponential moving average is higher than the error in non-real time flow estimation. Table 4.12 shows flow error and congestion level classification error with exponential moving average for all the scenarios. The least classification error of 4% is observed in S3 which has the least flow estimation error of less than 4% as well. Similarly, the highest congestion level classification error of 14% is observed in S2 which has the highest flow estimation error of less than 12%. It can be seen that the error in flow estimation is reflected in congestion level classification accuracy.

4.4.4 Discussion

The simulation results show the feasibility of real time estimation of traffic parameters using cellular network data. The map matching algorithm generates vehicle trajectories using which the vehicle flow and count data is computed for all the edges. The real time vehicle flow estimations are accurate with the mean error of less than 5% for regression

based extrapolation and less than 10% in most of the cases for exponential moving average based extrapolation.

It is assumed that all the vehicles are equipped with a cell phone. While it is a reasonable assumption considering cellular teledensity in urban areas, a small fraction of vehicles not having cell phone does not affect the overall accuracy of traffic parameter estimation.

The vehicle flow data are used to compute edge level space occupancy and congestion level estimation. The error in congestion level classification is affected by the accuracy of flow estimation. It needs to be emphasized that, even though traffic condition on an edge is categorized among six congestion levels, the classification error is notably less. This is considered significant because the proposals in literature attempts to classify the traffic state into fewer levels and report higher error (for example, in [37], the traffic state on an edge is classified into two levels (congested and uncongested) and accuracy of 90% is reported). The historical data of congestion level are used to compute congestion profile of edges which is further used by the COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO) for selecting edges for ITS infrastructure deployment (section 3.5.1 and 3.5.2).

The space occupancy error is significantly higher than flow error, but has good correlation with the actual occupancy estimation. The edge space occupancy data along with the real time congestion level data are subsequently used by the proposed ITS for edge level speed estimation (section 3.5.3).

4.5 Speed Estimation using COngestion COverage MOdel (COCOMO)

For these simulations, it is assumed that six percent vehicles (of bus and car types) are GPS enabled and report their accurate position every second. The collected trajectories of vehicles are aggregated to compute average speed of GPS probes for every congestion level for every edge. The historical data of GPS probes collected in this manner are used to spatial extrapolate speed estimation from an infrastructure edge to corresponding infrastructureless edge as described in section 3.5.3.

The COCOMO uses edge level congestion profile computed using cellular data to select optimal set of edges for ITS infrastructure deployment (section 3.5.1). It aims to provide K-coverage of all the congestion levels in a road network, that is, for every congestion level occurring in a road network, there are at least K edges with ITS infrastructure covering it. $K = 3$ is used in the simulations. When a congestion level occurs on an infrastructureless edge, the infrastructure edges covering the congestion level are used to compute real time speed estimation on the edge (section 3.5.3). The real time space occupancy estimation of infrastructureless edge is provided as an input to the regression model of infrastructure edges covering the congestion level. The regression model along with the speed transition function computed using historical data of GPS probes is used for real time speed estimation on the edge.

The performance of COCOMO is evaluated using scenarios S4 and S5 only. The reason being, the other scenarios (S1, S2, and S3) have fewer edges to provide K coverage for all the congestion levels (S1, S2 and S3 have 4, 4, and 9 edges, respectively). On the other hand S4 and S5 have sufficiently large number of traffic carrying edges (S4 has 40 edges and S5 has 52 edges) and are more suitable for evaluating effectiveness of the model in selecting edges for infrastructure deployment and edge level speed estimation.

4.5.1 ITS Infrastructure Requirement and Speed Estimation Error

Speed estimation is carried out for all the infrastructureless edges for congestion levels A-E. As congestion level F represents jam condition (unstable traffic regime), we do not do any speed estimation for the congestion level, even though infrastructure is deployed to cover all the congestion levels.

Figure 4.8 summarizes the mean speed error for scenarios S4 and S5 with maximum location error of 250m and 500m. The real time space occupancy estimation (which is provided as an input to the speed estimation model) based on real time flow data computed using exponential moving average and regression based temporal extrapolation, each, are considered. The number of infrastructure edges required to 3-cover all the congestion levels is also specified. The variations in speed error are denoted by error bars in the graph. The height of an error bar equals the standard deviation in speed error.

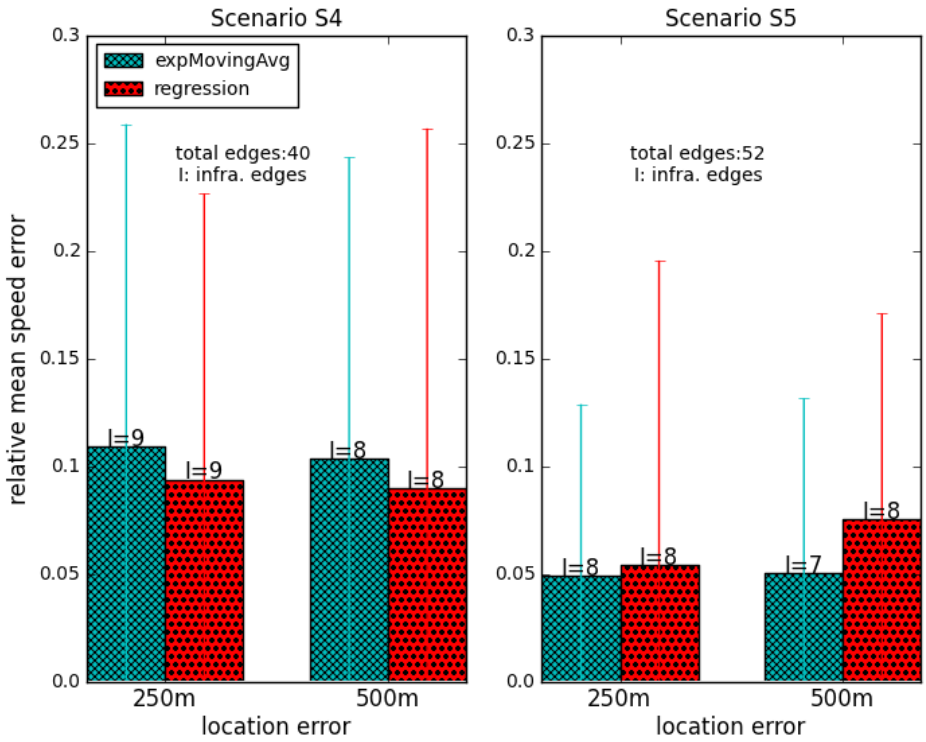
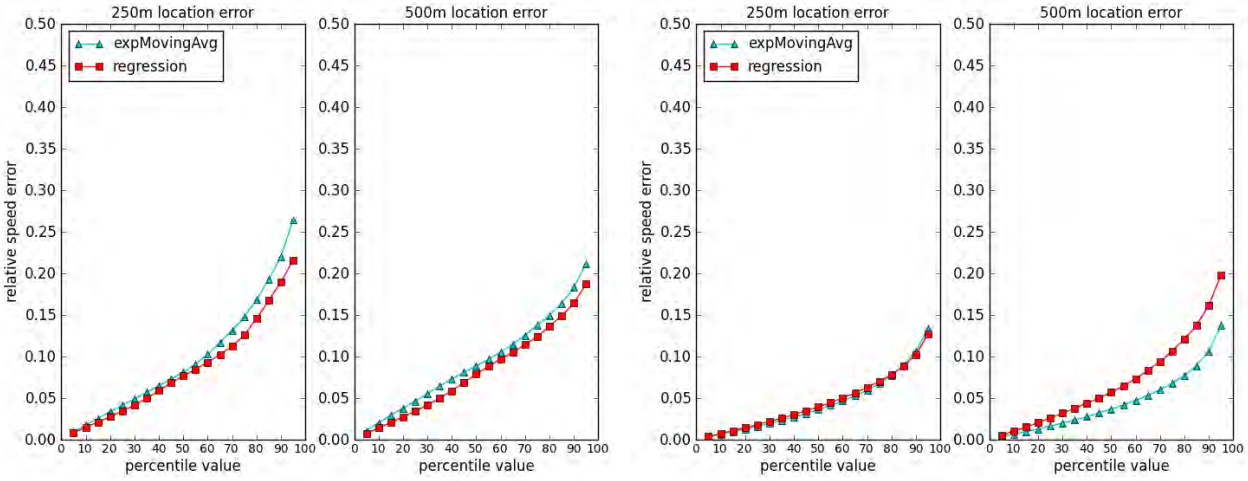


Figure 4.8: Mean Speed Error (COCOMO)



(a) Scenario S4

(b) Scenario S5

Figure 4.9: Percentile Speed Error (COCOMO)

Figure 4.9 shows percentile error in real time speed estimation for S4 and S5. Following are the important observations:

- The mean speed error of 10% and 5.0-7.5% is observed in scenarios S4 and S5, respectively. The median error of less than 10% is perceived in all the speed estimations (7-9% in S4 and 3.5-6.0% in S5). The ninety percentile error of 16-22%

Table 4.13: Error in Space Occupancy and COCOMO based Speed Estimation

Scenario	Location Error	Extra*-polation	Occupancy Error	Mean Speed Error
S4	250m	E1	15.00%	10.92%
		E2	09.90%	09.37%
	500m	E1	16.10%	10.32%
		E2	11.33%	08.99%
S5	250m	E1	16.27%	04.93%
		E2	10.12%	05.40%
	500m	E1	18.68%	05.04%
		E2	13.35%	07.53%
*E1 and E2 are flow extrapolation using exponential moving average and regression, respectively				

and 10-16% is observed in S4 and S5, respectively.

- The error in space occupancy estimation (Figure 4.5) does not propagate in the speed estimation. It is due to the following reasons: first, the regression model for speed estimation is learnt using estimated (erroneous) occupancy information, and second, most of the occupancy errors are systematic with very high correlation with the actual space occupancy. Table 4.13 show error in space occupancy and speed estimation for all the scenarios.

- The speed estimation results do not suggest any specific temporal extrapolation method for real time flow estimation. In S4, regression based extrapolation gives a slightly better speed estimation whereas in S5, specifically with 500m location error, exponential moving average based extrapolation gives better speed estimation. We reiterate that the regression based flow extrapolation generated better occupancy estimation (Table 4.13 and Figure 4.5) and congestion level classification (Figure 4.7) in all the cases. But the similar trend is not present in speed estimation.

- To 3-cover all the congestion levels in scenario S4 having 40 edges carrying traffic, 8-9 infrastructure edges are required, whereas in scenario S5 having 52 edges carrying traffic, 7-8 infrastructure edges are required.

The simulation results show the feasibility of edge level speed estimation with ITS infrastructure deployment on 15-20% edges in a road network. This is considered an important finding because various proposals in the literature require infrastructure deployment on

60-80% edges in a road network (section 2.5).

4.5.2 Effect of Unavailability of Infrastructure Edges

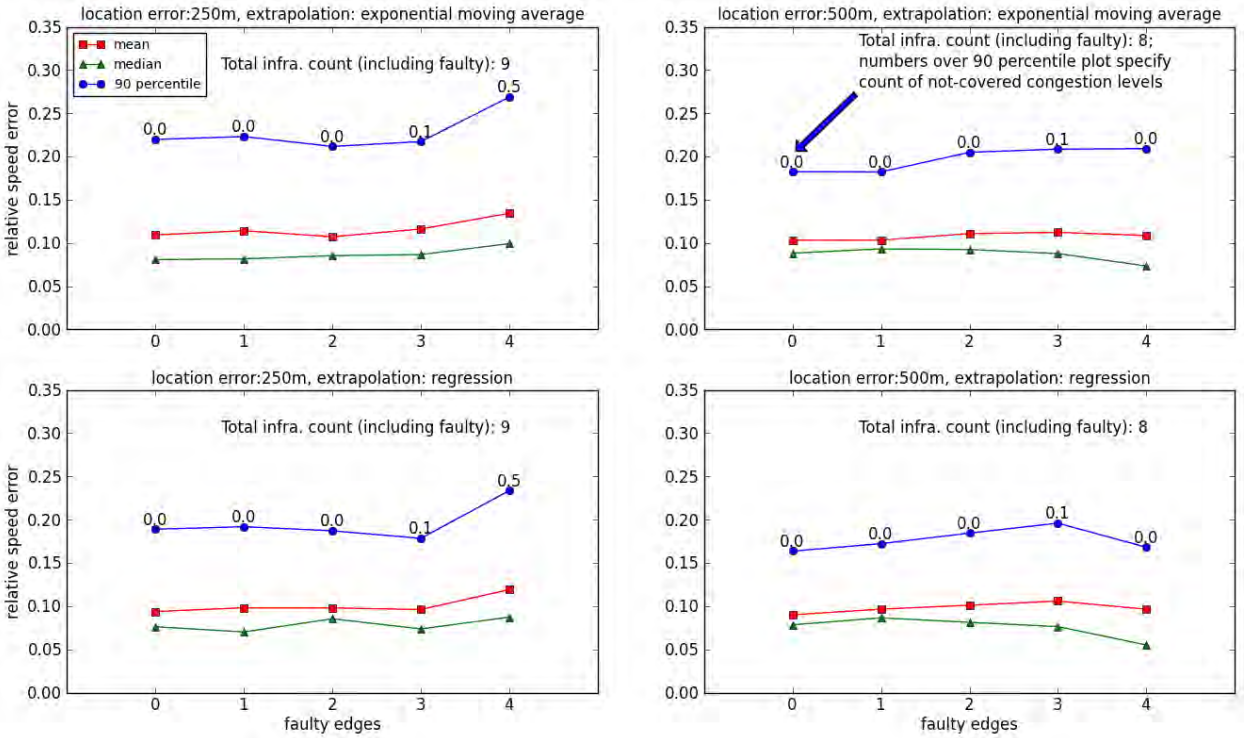
This section evaluates fault tolerance characteristics of the COCOMO by observing the effect of the unavailability of a certain fraction of infrastructure on speed estimation accuracy and congestion level coverage.

An infrastructure edge may be unavailable for speed estimation for given congestion level due to infrastructure failure or due to change in traffic pattern after infrastructure deployment. The unavailable infrastructure edges are treated as infrastructureless edges for the given congestion level. Speed estimation for all the infrastructureless edges is carried out using currently available infrastructure edges. The count of unavailable infrastructure edges is varied from 1 to 50% of total number of infrastructure edges, to analyze its effect on speed estimation and coverage of congestion levels. A sample of a certain number of infrastructure edges is picked and considered as the set of unavailable edges. The congestion level coverage and real time speed estimation is done using remaining infrastructure edges. In order to avoid special cases and to observe the aggregate effect, the process is repeated ten times and average speed error and congestion level coverage is computed.

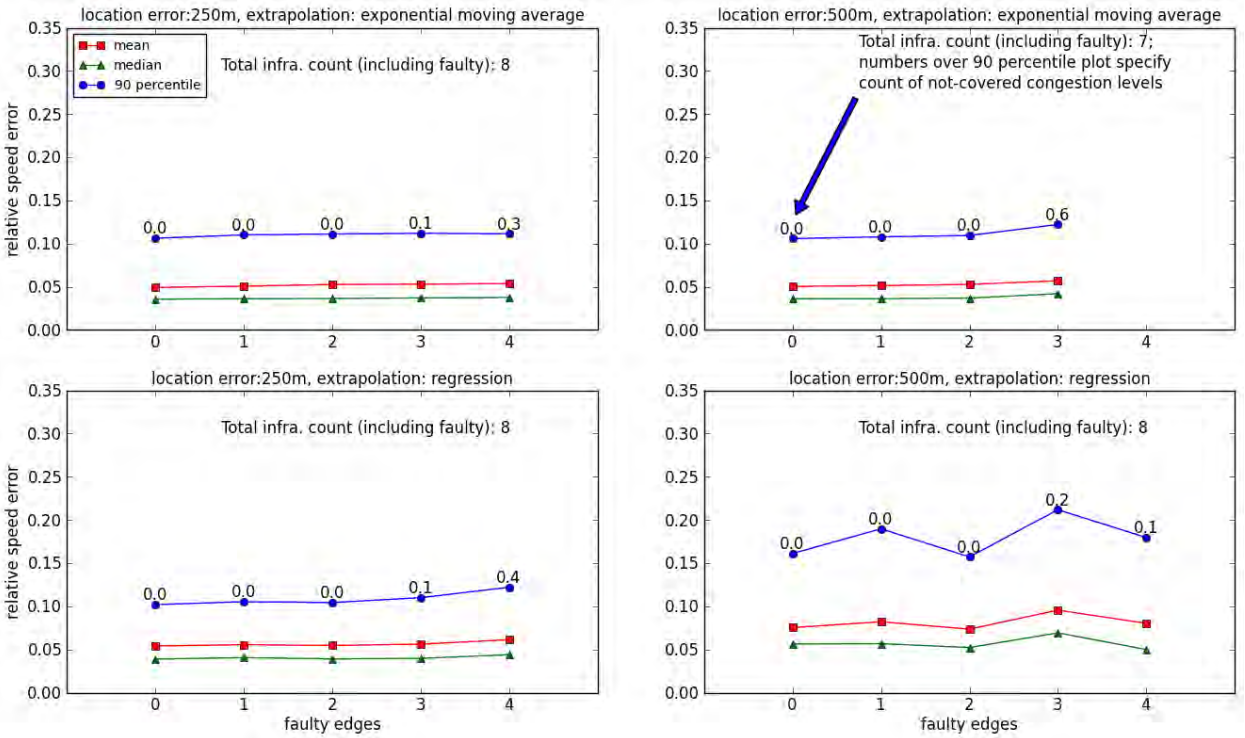
Figure 4.10 shows the effect of unavailability of a certain number of infrastructure edges on speed estimation (mean, median and ninety percentile error) and congestion level coverage for grid network (S4) and random network (S5). The count of not-covered congestion levels for a certain number of unavailable infrastructure edges is specified over ninety percentile speed error plot in the figure. Following are the important observations:

- The congestion level coverage is not much affected by the unavailability of infrastructure edges. Due to 3-cover deployment of infrastructure for every congestion level, most of the congestion levels are covered even with 50% unavailable infrastructure, and speed estimation is possible.

- The speed error does not vary according to a specific trend with the count of unavailable infrastructure edges. Inspecting standard deviation in speed error (error bars in Figure 4.8), and variations in speed error with number of unavailable infrastructure edges (Figure 4.10), it is inferred that there is no impact of unavailability



(a) Scenario S4



(b) Scenario S5

Figure 4.10: Effect of Unavailability of Infrastructure Edges (COCOMO)

of infrastructure edges on speed estimation accuracy. When at least one infrastructure edge is available to cover a congestion level, speed estimation on all the edges

experiencing the congestion level is feasible.

The simulation results exhibit fault-tolerance capability of the proposed model. It is observed that the redundant infrastructure deployment to K-cover congestion levels does not improve accuracy of speed estimation much. However, it serves two purposes: first, it increases fault tolerance capability of the model significantly, and second, it enables computation of anticipated speed error to be reported to the end user.

4.5.3 Effect of Limited Infrastructure Deployment

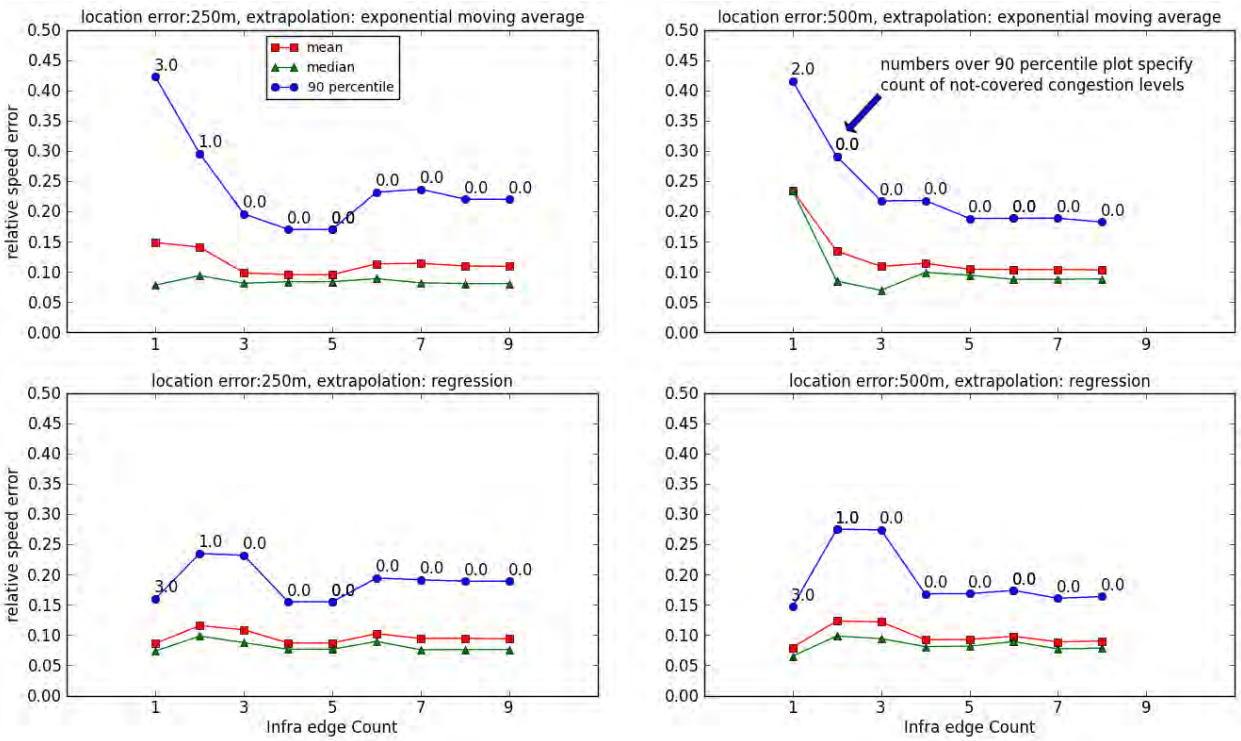
This set of simulations evaluates COCOMO for feasibility of incremental ITS infrastructure deployment. The effect of limited infrastructure deployment on speed estimation accuracy and congestion level coverage is analyzed.

The total ITS infrastructure requirement to 3-cover all the congestion levels in S4 and S5 are known (Figure 4.8). For limited infrastructure deployment, the number of infrastructure units is varied from 10% to 100% of total infrastructure requirement in increments of 10%. Given the certain number of ITS infrastructure units, say N, the algorithm 3.4 is executed to select an optimal set of N infrastructure edges to maximize congestion level coverage.

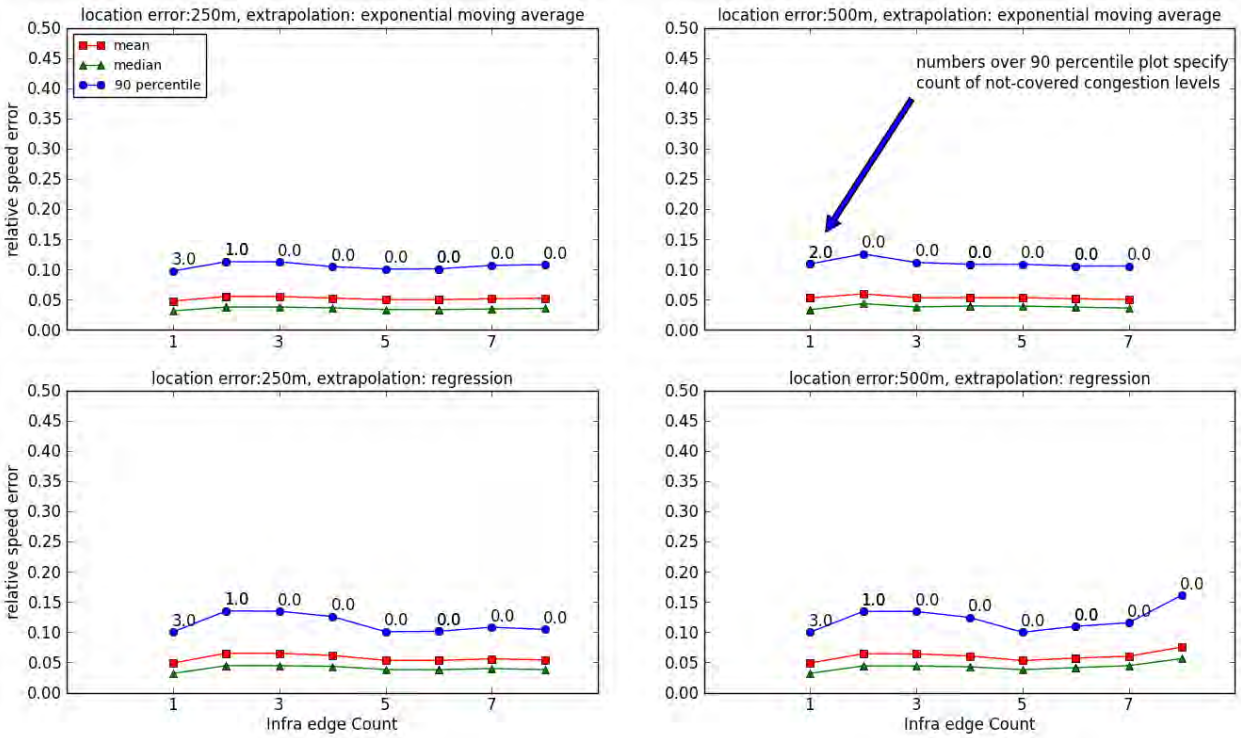
Figure 4.11 shows the effect of limited infrastructure deployment on speed estimation accuracy (mean, median and ninety percentile error) and coverage of congestion levels for grid network (S4) and random network (S5). The count of not-covered congestion levels for a certain number of infrastructure units is specified over ninety percentile speed error plot in the figure. Following are the important observations:

- With three ITS infrastructure units, atleast 1-coverage of all the congestion levels is achieved in both the scenarios in all cases (maximum location error of 250m and 500m, temporal extrapolation using exponential moving average and regression), permitting real time speed estimation on all the edges. The additional infrastructure units improve fault tolerance and enable anticipated error computation.

- When all congestion levels are not covered, high fluctuations in speed estimation error are observed, specifically in S4. The detailed analysis of simulation results pointed out two sources of error: the error in congestion level classification and



(a) Scenario S4



(b) Scenario S5

Figure 4.11: Effect of Limited Infrastructure Deployment (COCOMO)

infrastructure edges generating poor speed estimations. We explain the fact using speed estimation in 250m location error and exponential moving average based

extrapolation scenario in S4 (Figure 4.11a). Table 4.14 shows congestion level coverage and corresponding speed estimation error for the scenario.

With one infrastructure unit, congestion levels C and D are covered. For these congestion levels, speed estimation error is in a reasonable range (mean speed error is 6.46% and 10.9% for C and D, respectively). The congestion levels B and E get speed estimation due to congestion level classification error. Congestion level classification error for this scenario is 5% as reported in Figure 4.7, however the error involving B and E is higher (7.88%). When speed estimation is done in these cases, the significantly high error is observed (mean speed error of 31.50% and 46.10% for B and E, respectively). This is a primary cause of notably high ninety percentile error (42.21%) in the scenario.

The deployment of second infrastructure unit gets coverage for congestion level B reducing speed error for it. The ninety percentile error is still high (29.55%) in the scenario due to high speed error associated with false negatives of congestion level E.

With three infrastructure units, one coverage for all the congestion levels is achieved. The speed error measurements in this case are comparable to the 3-Coverage scenario reported in Figure 4.9 and 4.8.

The further increase in infrastructure units affects speed estimation accuracy based on the quality of speed estimations contributed by the newly selected infrastructure edge.

- The congestion level classification error does not necessarily result in high error in speed estimation in all cases. The speed error depends upon the speed difference between correct and estimated congestion level. The fact is observed in Figure 4.11b for exponential moving average based extrapolation scenario in S5 where the congestion level classification error does not contribute much to the overall speed error.

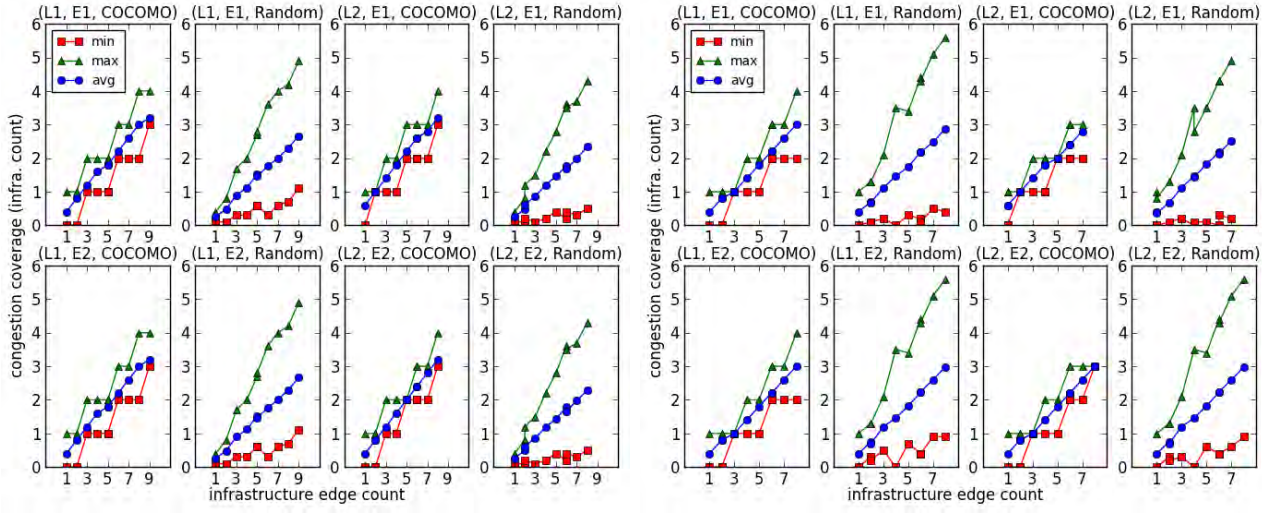
Figure 4.12 compares congestion level coverage (minimum, maximum and average) of COCOMO and random infrastructure deployment for grid network (S4) and random network (S5) in all the cases. Figure 4.13 shows details of congestion level coverage of

Table 4.14: Effect of Congestion Level Coverage on Speed Estimation Error in S4 with 250m Location Error and Exponential Moving Average based Extrapolation

Infra. Units	Congestion Level	Coverage	Speed Error (%)		
			Mean	Median	Ninety Percentile
1	B	0	31.50	02.57	122.25
	C	1	06.46	04.44	13.64
	D	1	10.90	09.10	18.81
	E	0	46.10	43.97	79.62
	All	–	14.85	07.82	42.21
2	B	1	13.56	13.59	22.22
	C	1	06.61	04.64	14.02
	D	1	10.90	09.10	18.81
	E	0	46.10	43.97	79.62
	All	–	14.09	09.38	29.55
3	B	1	13.55	13.59	22.22
	C	1	06.56	04.62	13.83
	D	2	08.84	07.82	15.56
	E	1	13.43	13.12	22.74
	All	–	09.84	08.11	19.57
4	B	2	09.15	08.30	14.17
	C	2	08.59	07.06	16.53
	D	2	10.25	09.11	17.46
	E	1	13.43	13.12	22.74
	All	–	09.55	08.38	17.03
6	B	2	09.15	08.30	14.17
	C	2	09.37	07.26	18.39
	D	3	19.64	19.34	33.09
	E	2	07.75	06.77	14.97
	All	–	11.34	08.88	23.17

COCOMO with exponential moving average based flow extrapolation and random deployment for S4 and S5 scenarios. The following methodology is adopted to compute the congestion level coverage of random deployment: a sample of given size is picked randomly from the set of edges in a road network and considered as the set of infrastructure edges; the congestion level coverage is computed using this set of edges; in order to avoid special cases and to observe the aggregate effect, the process is repeated ten times and average congestion level coverage is computed. Following are the important observations from this set of simulations:

- The congestion level coverage of COCOMO increases systematically with an increase in ITS infrastructure units ensuring maximal coverage of all the congestion



(a) Scenario S4 (b) Scenario S5

Figure 4.12: Comparison of Congestion Level Coverage: COCOMO and Random Deployment (L1 and L2 are location error of 250m and 500m, respectively; E1 and E2 are flow extrapolation using exponential moving average and regression, respectively)

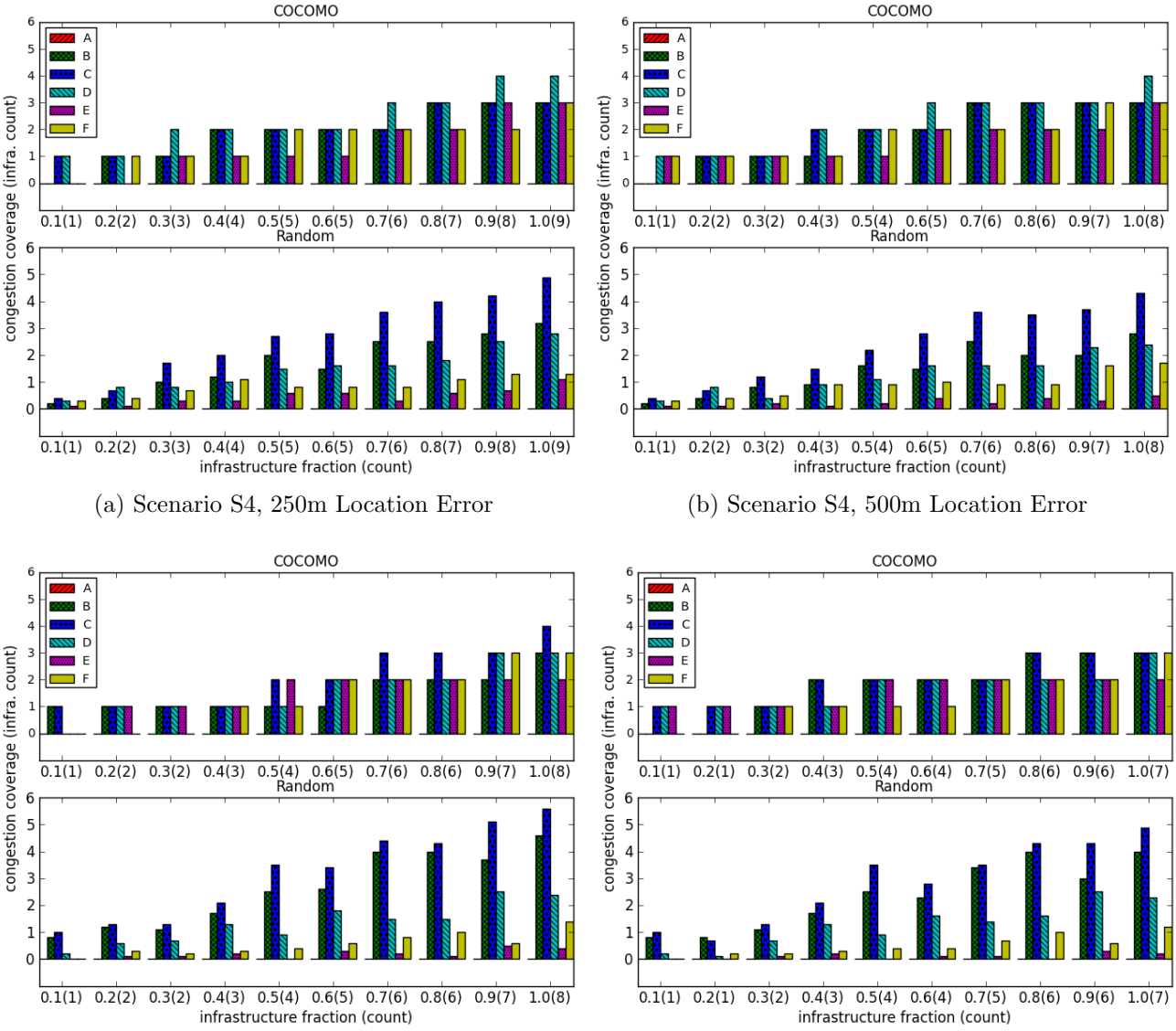
levels using available infrastructure.

- As expected, the random deployment fails to ensure maximal coverage of all the congestion levels. It has high coverage for some congestion levels (max plot in Figure 4.12) and very low coverage of other congestion levels (min plot in the figure). The reason being, some congestion levels occur more frequently in a road network than the others.

- In S5 (Figure 4.13c and 4.13d), there are only two edges that spend sufficient time under the congestion level E and are the only candidates for having the infrastructure to cover the congestion level. The COCOMO systematically assigns infrastructure to allow 2-coverage of the congestion level, whereas random deployment face difficulty in ensuring even 1-coverage of the congestion level.

- To ensure 1-coverage of all the congestion levels in all the scenarios, the COCOMO requires 3 infrastructure units, whereas the random deployment does not achieve 1-coverage even with 9 infrastructure units (infrastructure requirement to 3-cover all the congestion levels in COCOMO) in some cases.

  Figure 4.14 reiterates the fact and shows that COCOMO requires 2-3 infrastructure units to reduce the non-coverage of congestion levels to zero, whereas random



(a) Scenario S4, 250m Location Error

(b) Scenario S4, 500m Location Error

(c) Scenario S5, 250m Location Error

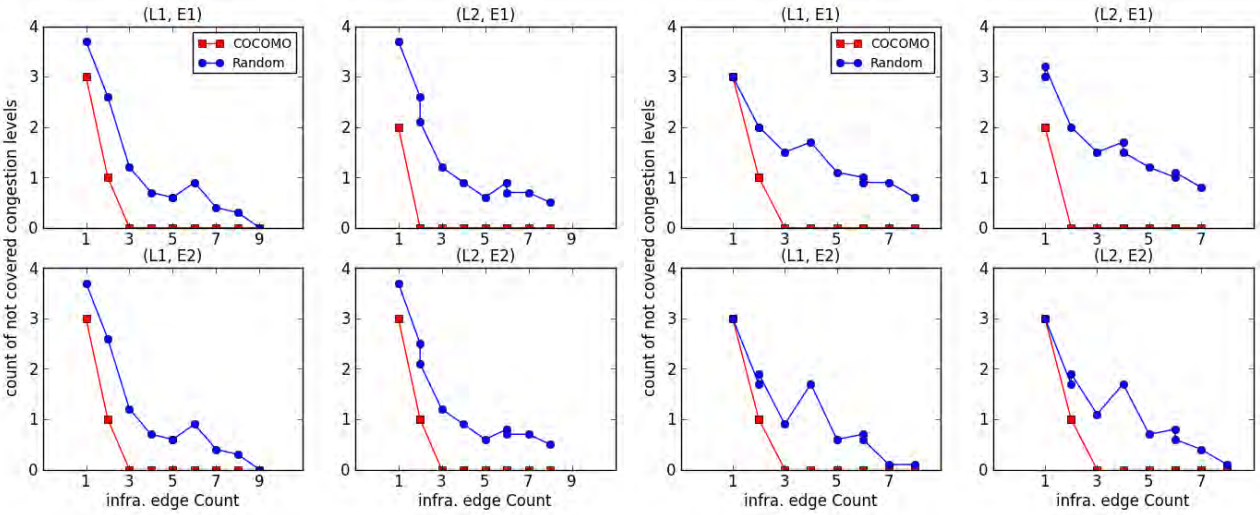
(d) Scenario S5, 500m Location Error

Figure 4.13: Details of Congestion Level Coverage: COCOMO and Random Deployment

deployment fails to do so.

4.6 Speed Estimation using Edge COverage MOdel (ECOMO)

The GPS probe data collection and processing is the same in COCOMO and ECOMO. In this set of simulations, it is assumed that six percent vehicles (of bus and car types) are GPS enabled and report their accurate position every second. The collected trajectories



(a) Scenario S4 (b) Scenario S5

Figure 4.14: Non-coverage of Congestion Level: COCOMO and Random Deployment (L1 and L2 are location error of 250m and 500m, respectively; E1 and E2 are flow extrapolation using exponential moving average and regression, respectively)

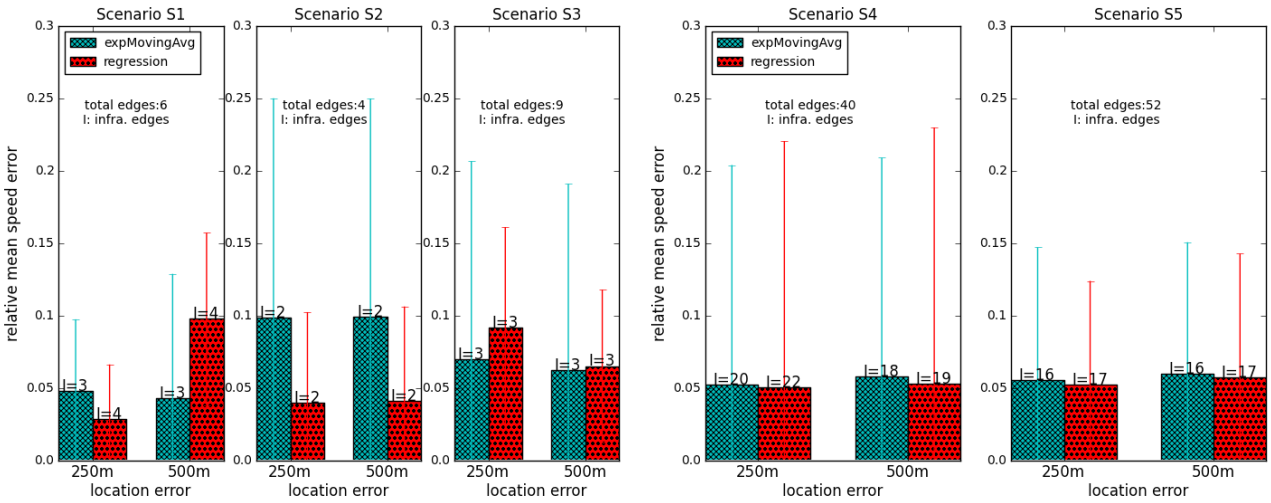
of vehicles are aggregated to compute average speed of GPS probes for every congestion level for every edge. The historical data of GPS probes collected in this manner are used to spatial extrapolate speed estimation from an infrastructure edge to corresponding infrastructureless edge as described in section 3.5.3.

The ECOMO aims to select an optimal set of edges for ITS infrastructure deployment (section 3.5.2). To accomplish this, it determines a set of similar edges for every edge in a road network using the congestion profile of edges computed using cellular network data. The aim is to provide K-coverage for all the edges in a road network, that is, for every infrastructureless edge in a road network, there should be at least K edges with ITS infrastructure, similar to it. $K = 3$ is used in the simulations. The infrastructure edges similar to an edge are used for speed estimation on the edge (section 3.5.3). The real time space occupancy estimation of infrastructureless edge is provided as an input to the regression model of infrastructure edges covering it. The regression model along with the speed transition function computed using historical data of GPS probes is used for real time speed estimation on the edge.

The performance of ECOMO is evaluated using all the scenarios, S1 to S5 (Table 4.2). The S1, S2 and S3 have fewer edges to provide 3-coverage for all the edges. Hence, performance evaluation is done with 1-coverage of edges in these scenarios. The S4 and

S5 have sufficiently large number of traffic carrying edges (S4 has 40 edges and S5 has 52 edges) and are more suitable for evaluating effectiveness of the model in selecting edges for infrastructure deployment and edge level speed estimation.

4.6.1 ITS Infrastructure Requirement and Speed Estimation Error



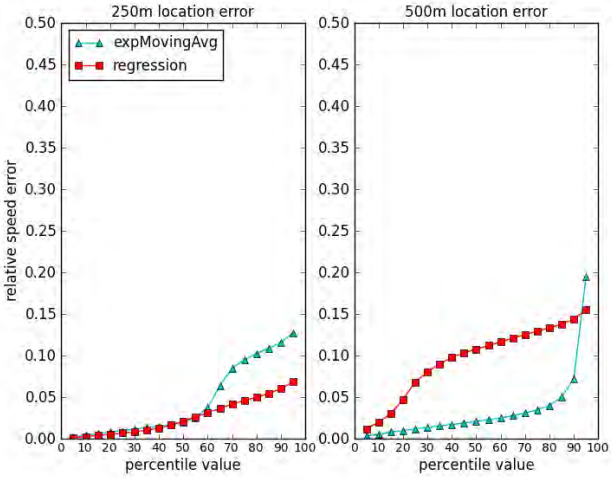
(a) Mean Speed Error (1-coverage)

(b) Mean Speed Error (3-coverage)

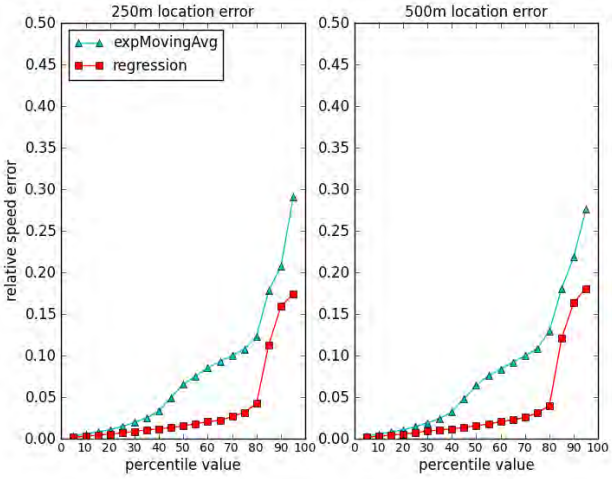
Figure 4.15: Mean Speed Error (ECOMO)

Speed estimation is carried out for all the infrastructureless edges for congestion levels A-E. As congestion level F represents jam condition (unstable traffic regime), we do not do any speed estimation for the congestion level, even though it is considered while computing the similarity of edges.

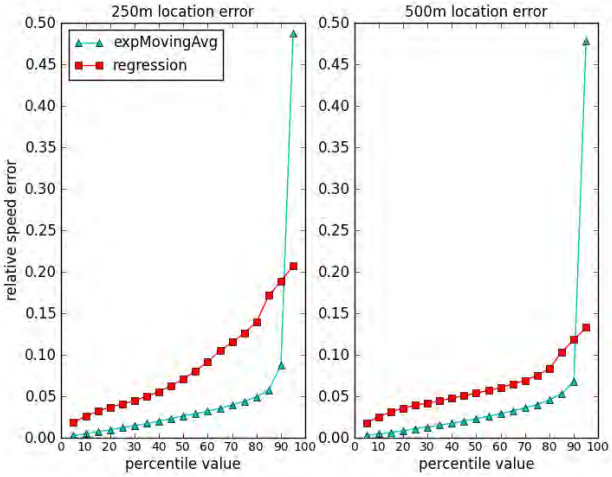
Figure 4.15 summarizes the mean speed error for all the simulation scenarios with maximum location error of 250m and 500m. The real time space occupancy estimation (which is provided as an input to the speed estimation model) based on real time flow data computed using exponential moving average and regression based temporal extrapolation, each, are considered. The number of infrastructure edges required to K-cover all the edges (K=1 for S1-S3 and K=3 for S4-S5) is also specified in the figure. The variations in speed error are denoted by error bars in the graph. The height of an error bar equals the standard deviation in speed error. Figure 4.16 shows percentile error in real time speed estimation for all the scenarios. Following are the important observations:



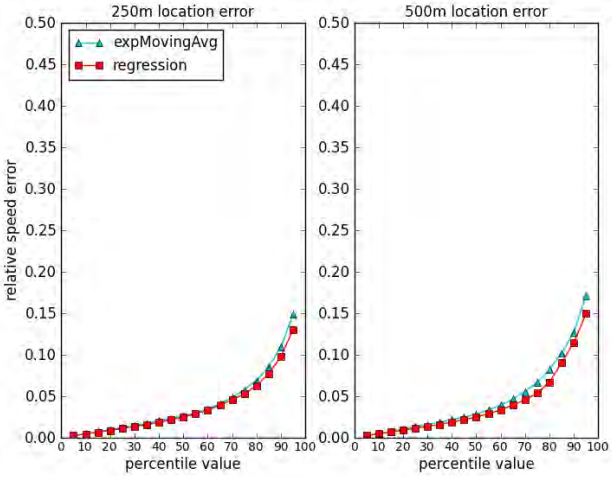
(a) Scenario S1 (1-coverage)



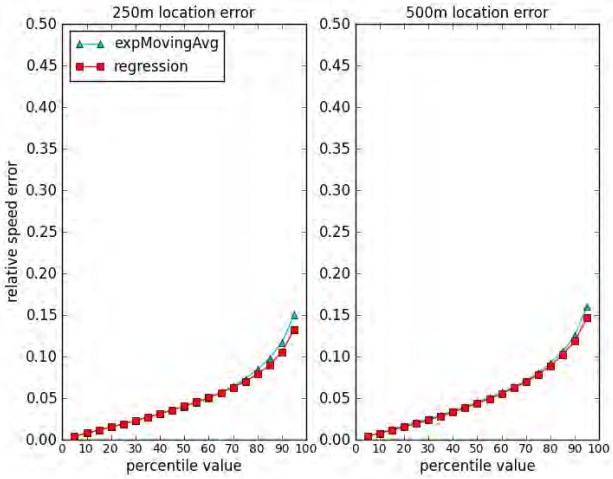
(b) Scenario S2 (1-coverage)



(c) Scenario S3 (1-coverage)



(d) Scenario S4 (3-coverage)



(e) Scenario S5 (3-coverage)

Figure 4.16: Percentile Speed Error (ECOMO)

- The mean speed error of less than 10% is observed in all the scenarios. In the large scale scenarios, S4 and S5, the mean error of 5-6% and ninety percentile error of 12-13% is perceived.

- In S1, S2 and S3, fluctuations in speed estimation accuracy are observed: the mean, median and ninety percentile error varies in the range of 2.8 to 9.8%, 1.5 to 10.7%, and 6.0 to 21.8%, respectively. In these small scale scenarios, the erroneous speed estimations by even one infrastructure edge significantly affect the overall accuracy.

  In S2, there are only two infrastructureless edges for speed estimation. With exponential moving average based flow extrapolation, the congestion level classification error of 15% is observed (Figure 4.7). This leads to comparatively high ninety percentile speed error (20.6%), which in turn increases the mean speed error (9.9%).

  In S3, with 250m location error and regression based flow extrapolation, the mean speed error of 9.1% and ninety percentile error of 18.8% is observed. The congestion level classification error in the scenario is less than 5% (Figure 4.7). The analysis revealed the fact that, the high ninety percentile error occurs due to erroneous speed estimation on one edge (mean error of 21%).

- The error in space occupancy estimation (Figure 4.5) does not propagate in the speed estimation. It is due to the following reasons: first, the regression model for speed estimation is learnt using estimated (erroneous) occupancy information, and second, most of the occupancy errors are systematic with very high correlation with the actual space occupancy. Table 4.15 show error in space occupancy and speed estimation for all the scenarios.

- The speed estimation results do not suggest any specific temporal extrapolation method for real time flow estimation. In S4 and S5, both the extrapolation methods give comparable accuracy in speed estimation. While S2 has better speed estimation with exponential moving average, the speed estimation in S3 are better with regression based extrapolation. We reiterate that the regression based flow extrapolation generated better space occupancy estimation (Table 4.15 and Figure 4.5) and congestion level classification (Figure 4.7) in all the cases. But the similar trend is not present in speed estimation.

Table 4.15: Error in Space Occupancy and ECOMO based Speed Estimation

Scenario	Location Error	Extra*-polation	Occupancy Error	Mean Speed Error
S1	250m	E1	33.67%	04.77%
		E2	23.08%	02.83%
	500m	E1	57.45%	04.25%
		E2	50.81%	09.75%
S2	250m	E1	33.47%	09.86%
		E2	07.81%	03.98%
	500m	E1	34.80%	09.88%
		E2	11.19%	04.08%
S3	250m	E1	5.26%	06.97%
		E2	02.87%	09.13%
	500m	E1	05.34%	06.22%
		E2	03.03%	06.50%
S4	250m	E1	15.00%	05.19%
		E2	09.90%	05.01%
	500m	E1	16.10%	05.78%
		E2	11.33%	05.30%
S5	250m	E1	16.27%	05.50%
		E2	10.12%	05.22%
	500m	E1	18.68%	05.96%
		E2	13.35%	05.72%
E1 and E2 are flow extrapolation using exponential moving average and regression, respectively				

- To 3-cover all the edges in scenario S4 having 40 edges carrying traffic, 18 to 22 infrastructure edges are required (about 50% of total number of edges), whereas in scenario S5 having 52 edges carrying traffic, 16 to 17 infrastructure edges are required (less than one third of the total number of edges).

The simulation results show the feasibility of accurate edge level speed estimation in real time with ITS infrastructure deployment on 35-50% edges in a road network. Here we need to clarify that the ITS infrastructure requirement mentioned above is for 3-coverage of all the edges in a road network. The 1-coverage of an infrastructureless edge is sufficient to generate real time speed estimations for the edge. As detailed in section 4.6.3 subsequently, 1-coverage of all the edges is achieved with 10 to 12 infrastructure edges in both the scenarios, S4 and S5. The 3-coverage of edges ensures high availability, in case of infrastructure failure or change in the traffic profile of edges, and permits computation of anticipated error to be reported to the end user.

Even for 3-coverage of all the edges, the infrastructure requirement of ECOMO is

lower than various proposals in literature that requires infrastructure deployment on 60-80% edges in a road network (section 2.5) just to enable estimation of certain traffic parameter(s).

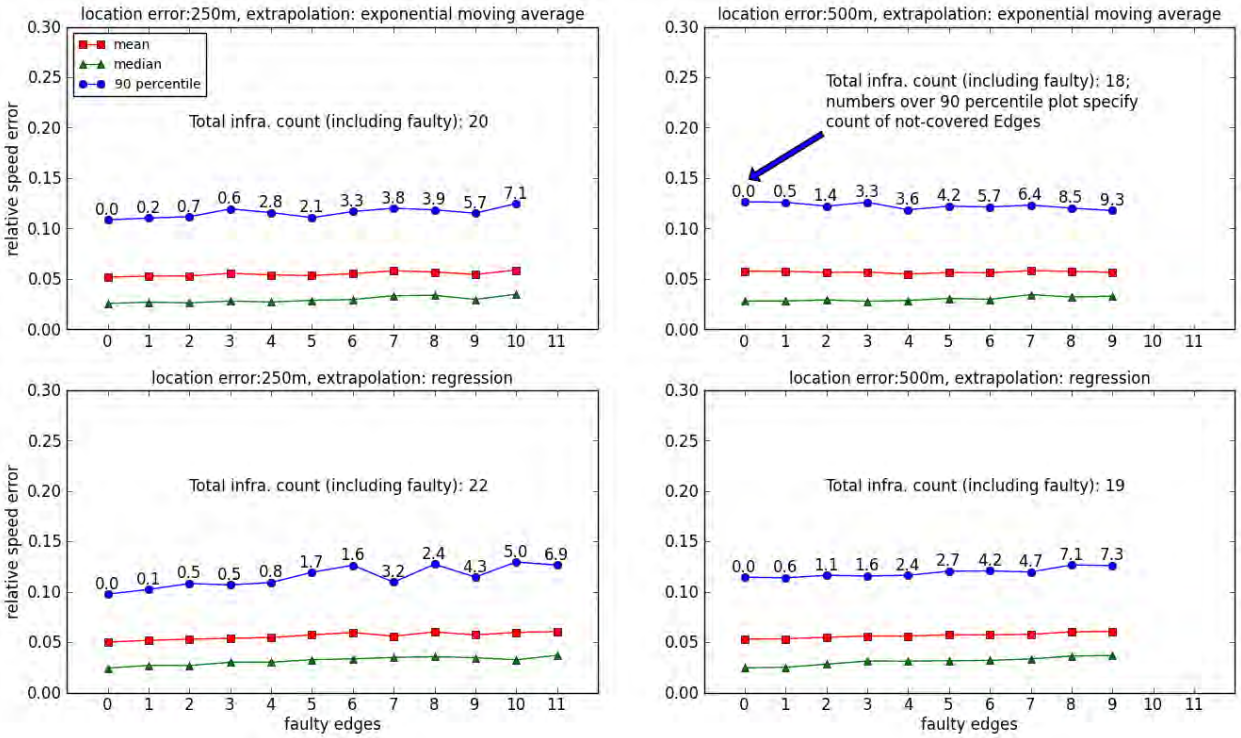
4.6.2 Effect of Unavailability of Infrastructure Edges

This section evaluates fault tolerance characteristics of the ECOMO using large scale scenarios, grid network (S4) and random network (S5). The aim is to analyze the effect of the unavailability of a certain fraction of infrastructure on speed estimation accuracy and edge coverage. The other scenarios (S1, S2, and S3) have 1-coverage of edges and can not tolerate faults. That is, unavailability of an infrastructure edge in these scenarios leads to unavailability of real time speed estimations on the covered edge(s).

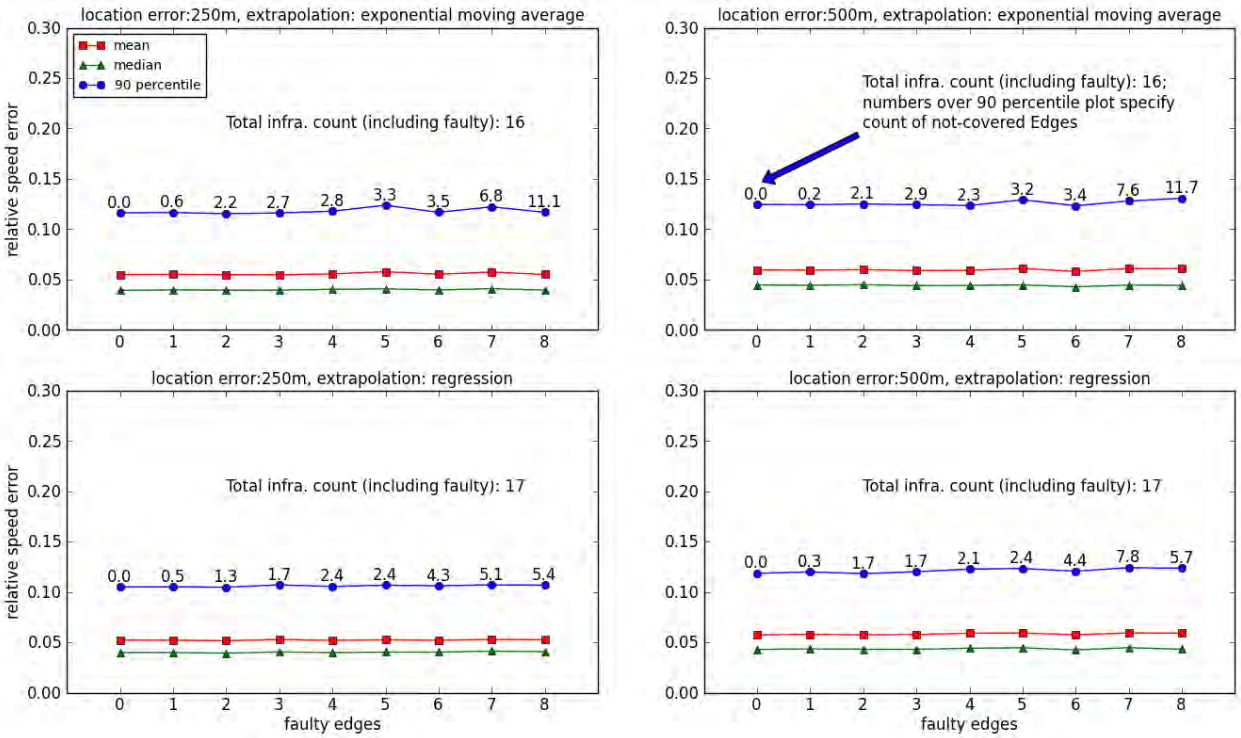
An infrastructure edge may be unavailable for speed estimation on an edge due to infrastructure failure or due to change in traffic pattern after infrastructure deployment. The unavailable infrastructure edges are treated as infrastructureless edges. Speed estimation for all the infrastructureless edges is carried out using currently available infrastructure edges. The count of unavailable infrastructure edges is varied from 1 to 50% of total number of infrastructure edges, to analyze its effect on speed estimation and coverage of edges. A sample of a certain number of infrastructure edges is picked and considered as the set of unavailable edges. The edge coverage and real time speed estimation is done using remaining infrastructure edges. In order to avoid special cases and to observe the aggregate effect, the process is repeated ten times and average speed error and edge coverage is computed.

Figure 4.17 shows the effect of unavailability of a certain number of infrastructure edges on speed estimation (mean, median, and ninety percentile error) and coverage of edges for S4 and S5. The count of not-covered edges for a certain number of unavailable infrastructure edges is specified over ninety percentile speed error plot in the figure. Following are the important observations:

- The unavailability of infrastructure edges has little impact on speed estimation accuracy, and ninety percentile error of less than 15% is achievable in all the cases.
- The speed error does not vary much or according to a specific trend with a count of unavailable infrastructure edges. Inspecting standard deviation in speed error



(a) Scenario S4



(b) Scenario S5

Figure 4.17: Effect of Unavailability of Infrastructure Edges (ECOMO)

(error bars in Figure 4.15b), and variations in speed error with number of unavailable infrastructure edges (Figure 4.17), it is inferred that there is no impact of

unavailability of infrastructure edges on speed estimation accuracy. When at least one infrastructure edge is available to cover an edge, speed estimation on the edge is feasible.

- The fraction of not covered edges increases with the number of unavailable infrastructure edges, but at a slower rate. With 50% unavailable infrastructure, S4 has 17-23% of not-covered edges, whereas S5 has 13-23% of not-covered edges.

The simulation results exhibit fault-tolerance capability of the proposed model. It is observed that the redundant infrastructure deployment to K-cover edges does not improve accuracy of speed estimation much. However, it serves two purposes: first, it increases fault tolerance capability of the model significantly, and second, it enables computation of anticipated speed error to be reported to the end user.

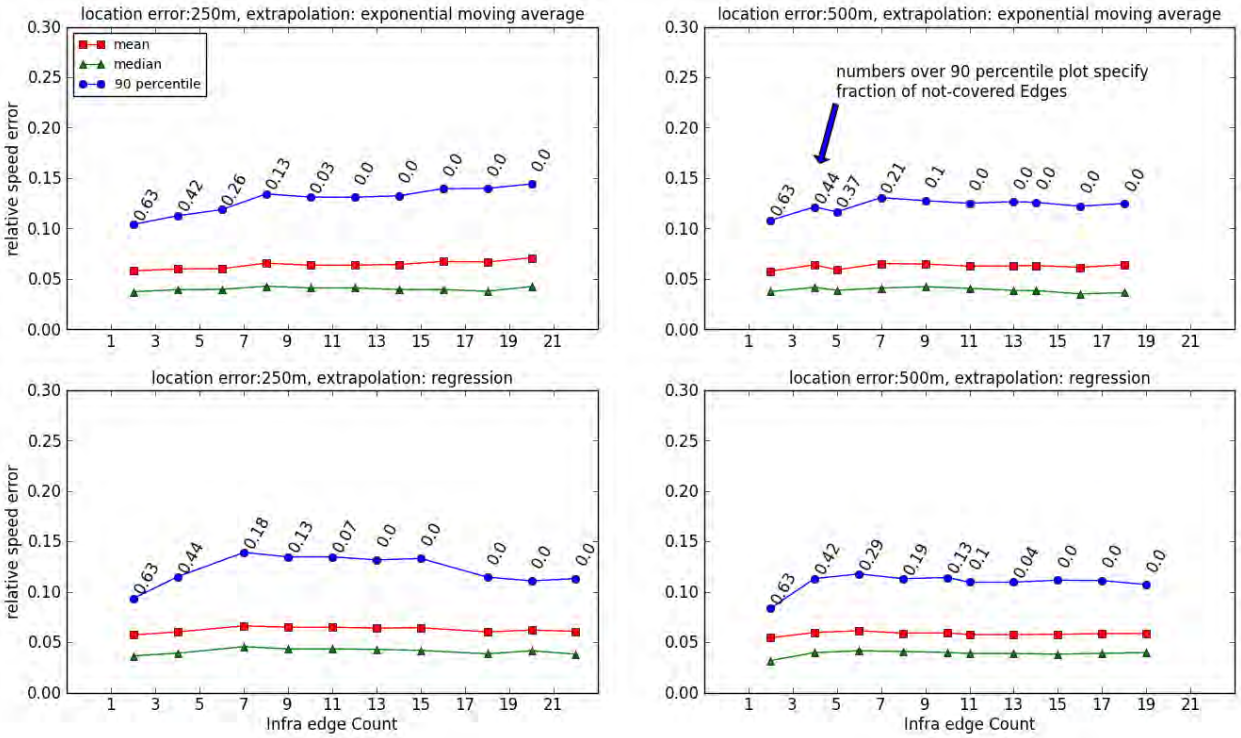
4.6.3 Effect of Limited Infrastructure Deployment

This set of simulations evaluates ECOMO for feasibility of incremental ITS infrastructure deployment. The effect of limited infrastructure deployment on speed estimation accuracy and edge coverage is analyzed. Only the large scale scenarios (S4 and S5) are used for this evaluation.

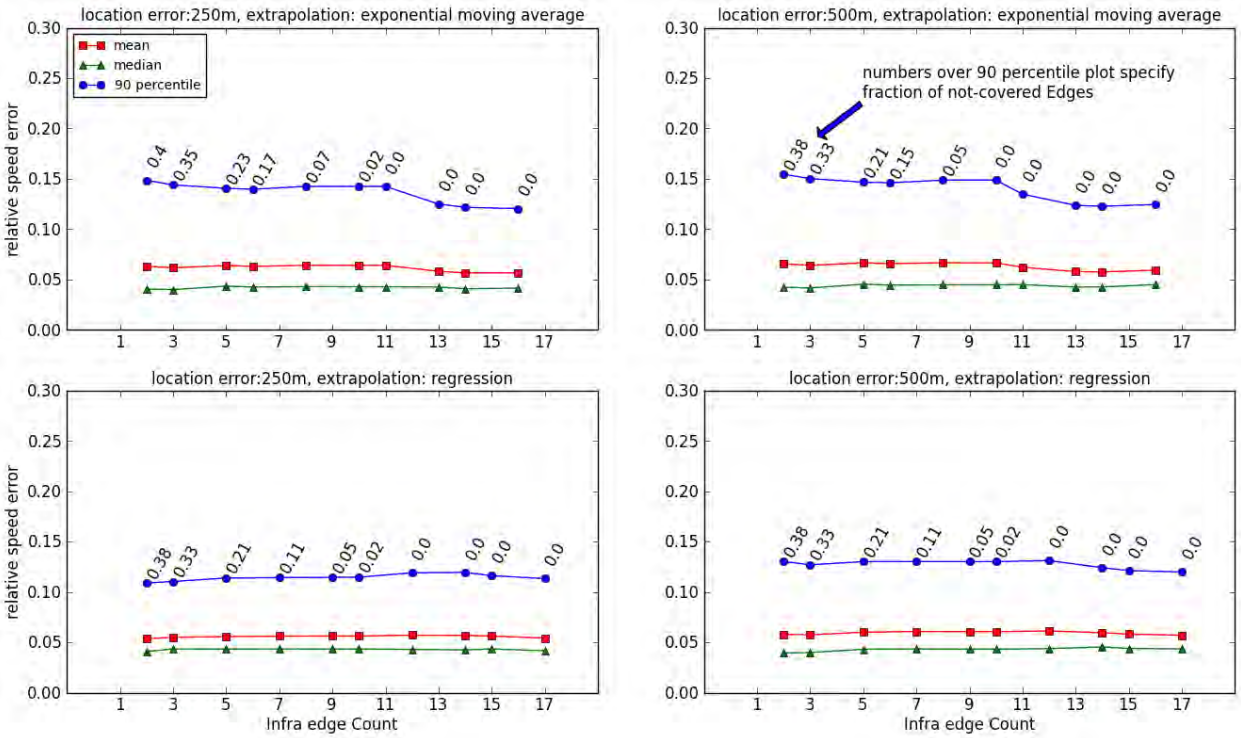
The total ITS infrastructure requirement to 3-cover all the edges in S4 and S5 are known (Figure 4.15b). For limited infrastructure deployment, the number of infrastructure units is varied from 10% to 100% of total infrastructure requirement in increments of 10%. Given the certain number of ITS infrastructure units, say N, the algorithm 3.6 is executed to select an optimal set of N infrastructure edges to maximize edge coverage.

Figure 4.18 shows impact of limited infrastructure deployment on speed estimation accuracy (mean, median and ninety percentile error) and coverage of edges for grid network (S4) and random network (S5). The fraction of not-covered edges for a certain number of infrastructure units is specified over ninety percentile speed error plot in the figure. Following are the important observations:

- The speed error does not vary much or according to a specific trend with the count of infrastructure edges. Inspecting standard deviation in speed error (error bars in Figure 4.15b), and variations in speed error with number of infrastructure



(a) Scenario S4



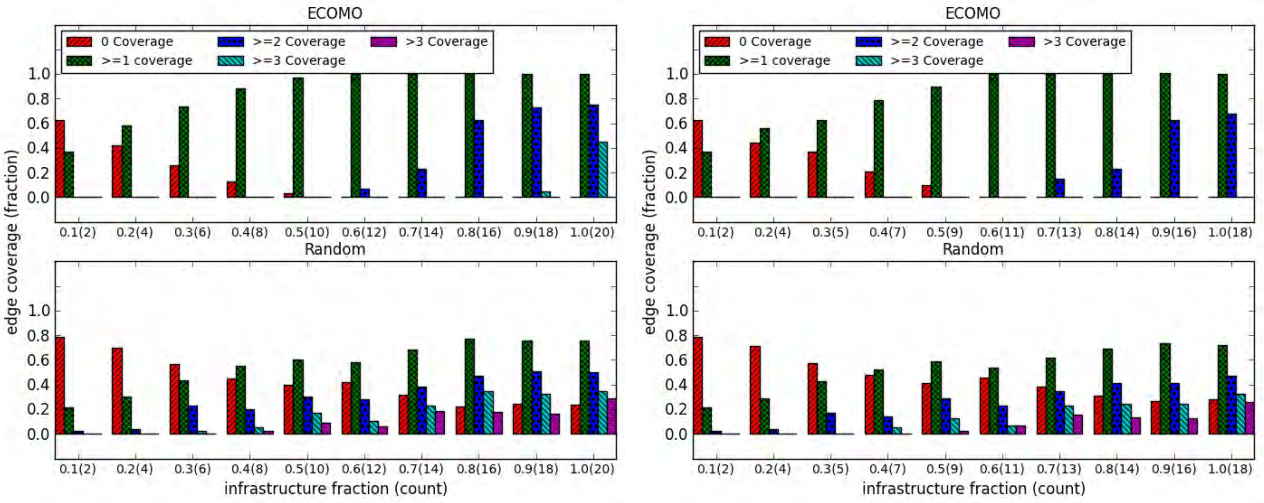
(b) Scenario S5

Figure 4.18: Effect of Limited Infrastructure Deployment (ECOMO)

edges (Figure 4.18), it is inferred that there is no impact of limited infrastructure deployment on speed estimation accuracy. When atleast one infrastructure edge is

available to cover an edge, real time speed estimation on the edge is feasible.

- With 10 to 12 infrastructure units, atleast 1-coverage of all the edges is achieved in both the scenarios in all cases (maximum location error of 250m and 500m, temporal extrapolation using exponential moving average and regression), permitting real time speed estimation on all the edges. The additional infrastructure units improve fault tolerance and enable anticipated error computation.



(a) Scenario S4, 250m Location Error

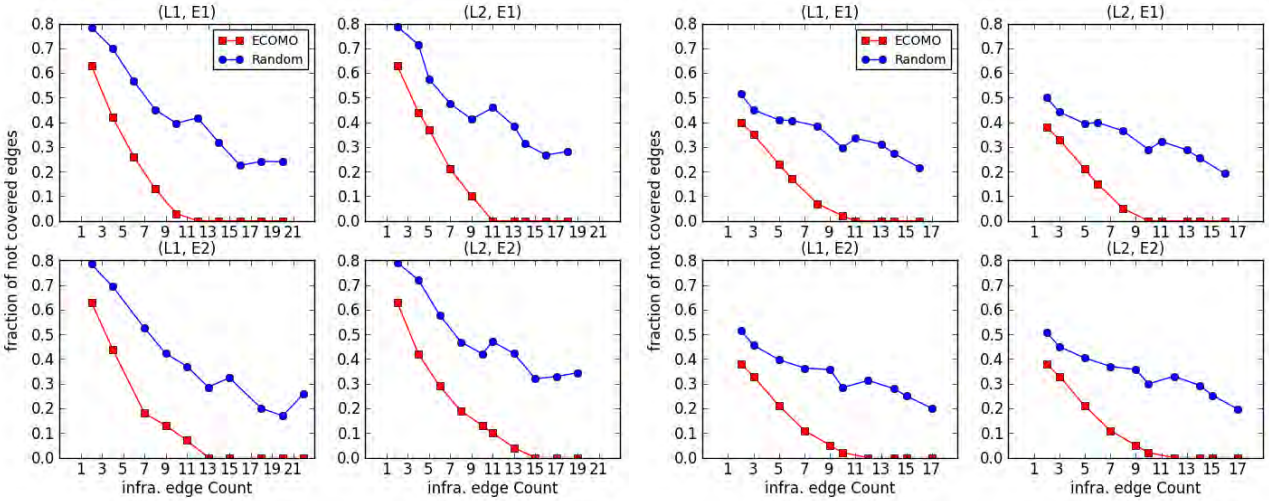
(b) Scenario S4, 500m Location Error

(c) Scenario S5, 250m Location Error

(d) Scenario S5, 500m Location Error

Figure 4.19: Details of Edge Coverage: ECOMO and Random Deployment

Figure 4.19 shows edge coverage details of ECOMO with exponential moving average based flow extrapolation and random deployment for S4 and S5 scenarios. Figure 4.20 compares edge coverage of ECOMO and random infrastructure deployment for grid



(a) Scenario S4 (b) Scenario S5

Figure 4.20: Comparison of Edge Coverage: ECOMO and Random Deployment (L1 and L2 are location error of 250m and 500m, respectively; E1 and E2 are flow extrapolation using exponential moving average and regression, respectively)

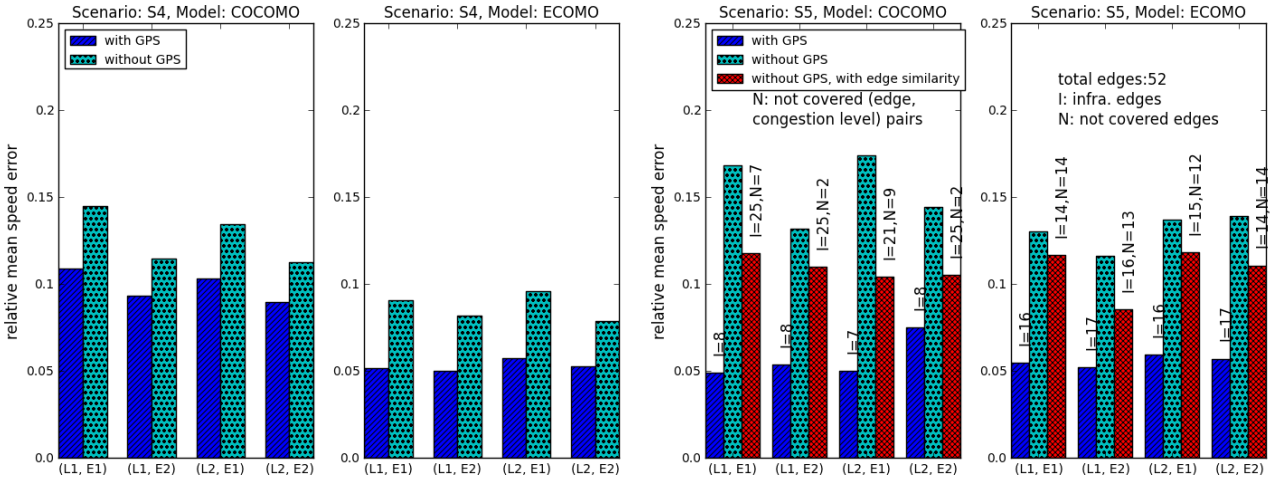
network (S4) and random network (S5) in all the cases. The following methodology is adopted to compute edge coverage of random deployment: a sample of given size is picked randomly from the set of edges in a road network and considered as the set of infrastructure edges; the edge coverage is computed using this set of edges; in order to avoid special cases and to observe the aggregate effect, the process is repeated ten times and average edge coverage is computed. Following are the important observations from this set of simulations:

- The edge coverage of ECOMO increases systematically with an increase in ITS infrastructure units ensuring maximal coverage of all the edges using available infrastructure. The model tries to achieve 1-coverage of all the edges with minimal infrastructure, followed by 2-coverage and 3-coverage of all the edges.

- As expected, the random deployment fails to ensure coverage of all the edges (has high coverage for some edges and very low coverage of other edges, with some edges not covered by any infrastructure edge). For example, even with 16-17 infrastructure units in S5 (the infrastructure requirement to 3-cover all the edges in ECOMO), there are approximately 20% not-covered edges, whereas more than 50% edges have $>$ 3-coverage (Figure 4.19c and 4.19d). The reason being, congestion profile of some edges occurs more frequently than the others in a road network.

- Figure 4.20 reiterates the fact and shows that ECOMO requires 10-12 infrastructure units to reduce the non-coverage of edges to zero in S4 and S5. With the same number of infrastructure units, the random deployment provides 1-coverage for less than 75% edges. The 1-coverage of all the edges is not achieved even with randomly deployed 16-20 infrastructure units (the infrastructure requirement to 3-cover all the edges in ECOMO for S4 and S5).

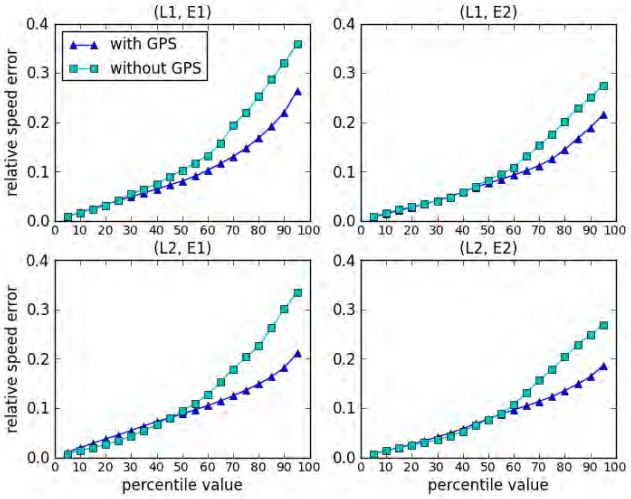
4.7 Speed Estimation when GPS Probe Data is Not Available



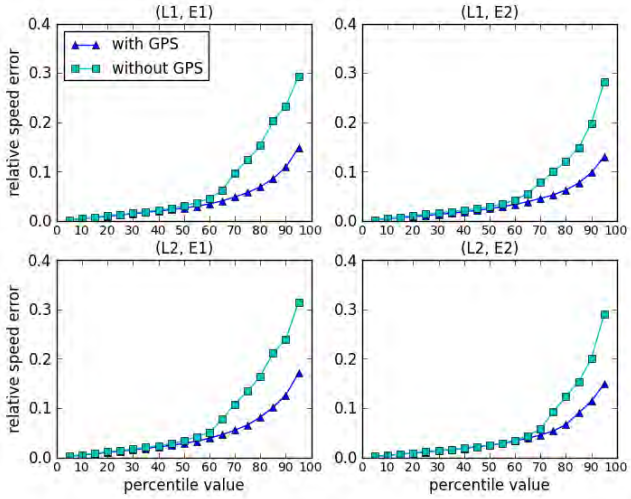
(a) Scenario S4 (b) Scenario S5

Figure 4.21: Effect of GPS Probe Data on Mean Speed Error (L1 and L2 are location error of 250m and 500m, respectively; E1 and E2 are flow extrapolation using exponential moving average and regression, respectively)

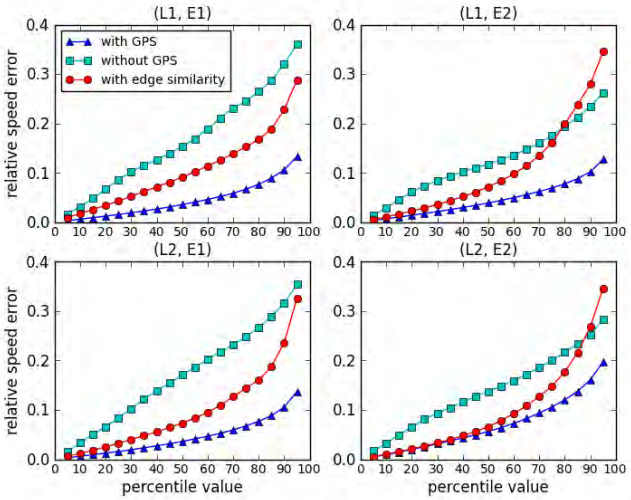
This set of simulations evaluates the effect of unavailability of historical GPS probe data on speed estimation. Also, the effect of considering static parameters of edges on speed estimation accuracy is examined. As mentioned earlier, the vehicle movement speed on an edge is affected not only by space occupancy, but also by many other parameters. For example, the edge length and edge speed are highly correlated (Pearson's correlation coefficient >0.9) for edges with traffic lights, according to our simulation results. The COCOMO and ECOMO use historical data of GPS probe speed to compute a speed transition function for every pair of edges, (infrastructure edge and associated infras-



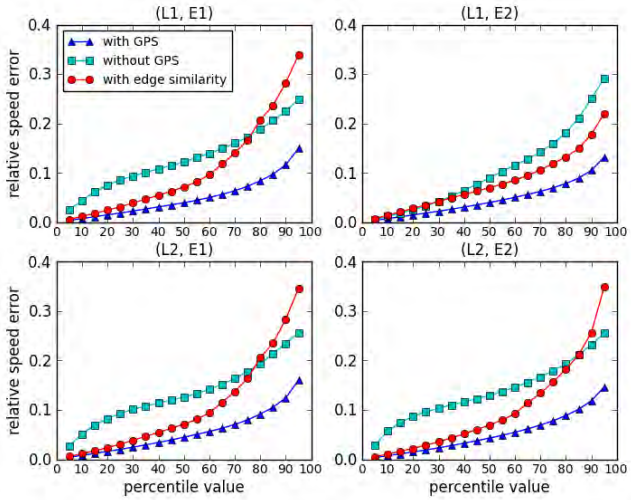
(a) Scenario S4: COCOMO



(b) Scenario S4: ECOMO



(c) Scenario S5: COCOMO



(d) Scenario S5: ECOMO

Figure 4.22: Effect of GPS Probe Data on Percentile Speed Error

tructureless edge), for every congestion level. The speed transition function is used to spatially extrapolate speed estimation from the infrastructure edge to the infrastructure-less edge. Also, GPS probe data is used to compute anticipated speed estimation error, reported to the user along with the speed estimation, when fewer infrastructure edges for speed estimation are available (section 3.5.3).

When historical data of GPS probes is not used, the speed estimation without applying speed transition function is reported to the end user. Figure 4.21 and 4.22 show the effect of (not) using GPS probe data on mean speed error and percentile speed error, respectively, for grid network scenario (S4) and random network scenario (S5).

The COCOMO and ECOMO are modified to use static parameters of edges, in addi-

tion to congestion profile, for infrastructure edge selection (section 3.5.5). Figure 4.21b show the mean speed error and infrastructure requirement of the modified models for S5. The figure also compares performance of the modified models against the version of COCOMO and ECOMO that (do not) use GPS probe data. It needs to be mentioned that the grid network scenario (S4) has same static parameters for all the edges. Figure 4.22c and 4.22d compare percentile speed error of modified models and the COCOMO and ECOMO that (do not) use GPS probe data. Following are the important observations:

- The mean speed error increases significantly when GPS probe data is not used. In scenario S4, the mean speed error of 9-11% and 5-6% is observed with COCOMO and ECOMO, respectively, when GPS probe data is used. Without GPS probe data, the mean speed error increases to 11.2-14.5% and 7.9-9.6% with COCOMO and ECOMO, respectively.

  In scenario S5, the mean speed error of 5-7.5% and 5-6% is observed with COCOMO and ECOMO, respectively, when GPS probe data is used. Without GPS probe data, the mean speed error increases to 13.2-17.4% and 11.6-13.9% with COCOMO and ECOMO, respectively.

  In grid network (S4), all the edges are identical as per static parameters. The reason for the speed error increase in S4 is that the vehicle movement speed on an edge is affected by many other parameters (e.g., traffic condition on the adjacent edge) which is not captured by the models without speed transition function.

  In random network (S5), the increase in speed error is higher than S4. The reason being, the edges in random network are different as per static parameters (specifically length), leading to difference in vehicle movement speed which is not captured by the models without the speed transition function.

- The results of ninety percentile error are no different and the speed estimations without GPS probe data are significantly more erroneous than with GPS probe data.

- When static parameters of edges are considered in modified COCOMO and ECOMO, the speed estimation error improves (Figure 4.21b, 4.22c and 4.22d). It is due to the fact that the modified models are able to capture the effect of static parameters on

speed estimation. In scenario S5, the mean speed error of 10.4-11.8% and 8.6-11.9% is observed with modified COCOMO and ECOMO, respectively.

The speed estimation error is still higher than that achieved using GPS probe data. The reason is, the modified models are unable to apprehend effect of other parameters (e.g., traffic conditions on the adjacent edges and systematic error in space occupancy estimation).

- The improvement in mean speed error comes at the cost of higher infrastructure requirement. The modified COCOMO (that considers static parameters of edges) requires about three times more infrastructure units than COCOMO (that uses GPS probe data) for scenario S5 and still has a few (edge, congestion level) pairs not-covered. The modified ECOMO has about 25% (12-14 edges out of 52) not-covered edges with similar infrastructure deployment as ECOMO.

- The percentile error plots (Figure 4.22) show some interesting facts. In S4, the median speed error with and without GPS probe data is comparable in COCOMO and ECOMO. This is indicative of the fact that on certain edges the vehicle movement speed is similar and speed transition using GPS probe data is not necessary.

  In S5, when GPS probe data is not used, the modified COCOMO and ECOMO that use static parameters of edges, show better performance upto seventy five percentile speed error. However, beyond that (eighty to ninety percentile regime) the performance is unpredictable. In some cases, due to unknown reasons, the modified models have higher speed error than COCOMO and ECOMO without speed transition.

The simulation results show that the historical data of GPS probes are extremely useful to reduce the infrastructure requirement and enable better speed estimation. While the COCOMO and ECOMO are modified to consider static parameters of edges when GPS probe data is not available, the infrastructure requirement of the modified models is significantly high. Also the modified models have higher speed error than COCOMO and ECOMO. Hence, we conclude that the GPS probe data are very important and integral part of COCOMO and ECOMO.

4.8 Comparison of COCOMO and ECOMO

In previous sections (section 4.5, 4.6, and 4.7), the performance of COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO) is evaluated for ITS infrastructure requirement and speed estimation accuracy. We conclude this chapter by highlighting important features of the models:

- The COCOMO and ECOMO use congestion profile of edges for selecting edges for ITS infrastructure deployment. The congestion profile of an edge is a 6-tuple specifying the percentage of overall time an edge observes each congestion level.

  The COCOMO aims to cover all the congestion levels occuring in a road network using ITS infrastructure. It processes congestion profile of edges to determine the congestion level(s) covered by every edge. An edge spending sufficient time observing a congestion level is said to cover the congestion level. The edges for ITS infrastructure deployment are selected to ensure K-coverage of all the congestion levels. Every infrastructure edge learns a regression model of occupancy-speed relationship for every congestion level that it covers.

  For real time speed estimation on an infrastructureless edge, the real time congestion level and space occupancy information computed using cellular network data are used. The congestion level is used to select infrastructure edges for speed estimation. The space occupancy is provided as an input to the regression model learned on the infrastructure edge for the congestion level. It should be noted that a different set of infrastructure edges may be used for speed estimation at different times based on the congestion level observed on the infrastructureless edge.

  The ECOMO, on the other hand, processes congestion profile of edges to determine the set of similar edges for every edge in a road network. The set of edges having congestion profile similar to an edge are said to be covered by the edge. The edges for ITS infrastructure deployment are selected to ensure K-coverage of all the edges in a road network. Every infrastructure edge learns a regression model of the occupancy-speed relationship.

  For real time speed estimation on an infrastructureless edge, the real time space occupancy information is provided as an input to the regression model of covering

infrastructure edges. The set of covering infrastructure edges used for speed estimation on an edge is fixed and does not depend upon the real time congestion level observed on the edge.

- The infrastructure requirement of ECOMO is higher than that of COCOMO. The ECOMO uses edge similarity based on congestion profile for infrastructure deployment. As the number of edges increases in a road network, the diversity in congestion profiles of these edges also increases, leading to increase in the infrastructure requirement. That is the infrastructure requirement increases with the number of edges in a road network.

  The COCOMO, on the other hand, aims to K-cover all the congestion level. Hence, the maximum infrastructure requirement is K times the number of congestion levels (6), independent of the road network size.

- The real time speed estimations with ECOMO are more accurate than COCOMO. In scenario S4, the mean speed error of 9-11% and 5-6% is observed with COCOMO and ECOMO, respectively, whereas the ninety percentile error of 16-22% and 10-13% is perceived with COCOMO and ECOMO, respectively.

  The results with scenario S5 are no different wherein the mean speed error of 5-7.5% and 5-6% is achieved with COCOMO and ECOMO, respectively, and the ninety percentile error of 10-16% and 10-13% is perceived with COCOMO and ECOMO, respectively.

  While the speed estimations with both the models are reasonably accurate, we point out the following reason for slightly high error in COCOMO. In COCOMO, the infrastructure edges for real time speed estimation are picked based on the real time congestion level observed on an infrastructureless edge. The congestion level classification error may result in use of an incorrect set of infrastructure edges for speed estimation, increasing error in the estimation. In ECOMO, on the other hand, the infrastructure edges to be used for speed estimation on an edge are fixed and does not depend upon the real time congestion level on the edge.

  In chapter 6 it is shown that the speed estimation with accuracy achieved by both the models is adequate for use by an Advanced Traveler Information System (ATIS).

- The 1-coverage of congestion levels (in COCOMO) or edges (in ECOMO) is sufficient for network wide real time speed estimation. The redundant infrastructure deployment does not improve accuracy of speed estimation. However, it improves fault tolerance capability of the models and enables computation of anticipated error to be reported to the end users.

- Both the models, COCOMO and ECOMO, permit incremental infrastructure deployment. While COCOMO requires 2-3 infrastructure units to 1-cover all the congestion levels in scenario S4 and S5, the ECOMO requires 10-12 infrastructure units to 1-cover all the edges in S4 and S5.

  In COCOMO, when fewer infrastructure units (less than that required for 1-coverage) are deployed, speed estimations on all the edges are affected. The reason is that the speed estimations are not available for not-covered congestion levels for all the edges. It should be noted that infrastructure requirement of the model for 1-coverage of congestion levels is nominal and can be satisfied with a maximum of six infrastructure units.

  In ECOMO, when fewer infrastructure units are deployed, the speed estimations are available only for the covered edges. When network wide coverage is not needed, the model permits infrastructure deployment to cover only the edges of interest.

Chapter 5

Distributed Processing and Communication Framework

5.1 Background

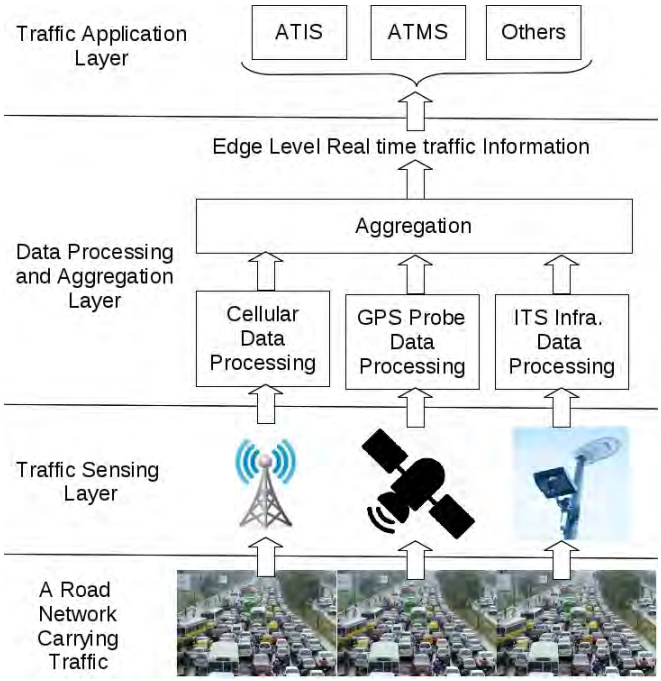


Figure 5.1: Multi-modal Intelligent Transportation System

In chapter 3, the details about various functional components of the proposed Intelligent Transportation System (ITS) are discussed. Figure 5.1 shows a block diagram of the resulting multi-modal ITS. The Traffic Sensing Layer uses cellular network, GPS probes

and ITS infrastructure for generating raw traffic data. The cellular network tracks all the vehicles using active signaling and generates erroneous position data periodically. The GPS probes report their accurate positions periodically. The ITS infrastructure deployed using COngestion COverage MOdel (COCOMO) or Edge COverage MOdel (ECOMO) reports edge level speed estimation periodically for all the infrastructure edges.

The Data Processing and Aggregation Layer computes traffic parameters for all the edges in a road network using the raw traffic data. The Cellular Data Processing involves execution of the map matching algorithm (section 3.2) to compute the trajectory of all the vehicles in real time. The vehicle trajectories in turn are used to compute edge level traffic parameters such as vehicle flow, space occupancy and congestion level (section 3.3). The trajectory computation is done for every vehicle, whereas the traffic parameters are computed for every edge in the road network periodically. The system must be scalable due to the real time processing requirement.

It is assumed that some fraction of vehicles of bus and car types are GPS-enabled and report their accurate locations periodically to the system. The GPS Probe Data Processing involves computation of a speed transition function for every pair of infrastructure edge and the corresponding infrastructureless edge. The GPS probe data is also used to compute anticipated error in speed estimation. The system uses historical data of GPS probes and there is no real time processing requirement.

The COCOMO and ECOMO determine edges for ITS infrastructure deployment using the congestion profile of edges. As a part of ITS Data processing, the occupancy-speed relationship is learned on every infrastructure edge. The process is done offline using historical data. The learned model is used for real time speed estimation on corresponding infrastructureless edges. The real time processing is not computationally intensive and involves execution of regression model and speed transition function.

The Traffic Application Layer uses the real time traffic information generated by the multi-modal ITS in a variety of traffic applications such as Advanced Traveler Information System (ATIS) and Advanced Traffic Management System (ATMS).

This chapter addresses issues related to large scale deployment of the proposed multi-modal ITS. The MapReduce framework proposed by Google [24] is used to model the ITS. The MapReduce framework permits real time traffic data processing on a large cluster of commodity machines. It works well under heavy computation load condition

involving large amount of data. The communication and storage requirement of the model is analyzed to assess feasibility of deployment.

The chapter is organized as follows: section 5.2 introduces the MapReduce framework; the cellular network data processing using MapReduce framework is discussed in section 5.3; the MapReduce framework for the GPS probe data processing and the edge level speed estimation is discussed in section 5.4 and section 5.5, respectively; the major observations and conclusion are described in section 5.6.

5.2 The MapReduce Framework

The MapReduce framework [24] proposed by Google provides an abstraction layer to design and implement programs for execution on a large cluster of commodity machines. The input data set is specified as key-value pairs. The programmer defines a map function and a reduce function for a distributed application. The map function processes a unit data element (a key-value pair) to generate a set of intermediate key-value pairs. The reduce function aggregates intermediate values associated with an intermediate key to compute the final result. The programs written in this manner are parallelized automatically and the run time system takes care of parallelization, fault tolerance, data distribution and load balancing.

Figure 5.2 shows functional block diagram of the MapReduce framework. The data to be processed is available as [key, value] pairs in a distributed file system (DFS), e.g. HDFS - Hadoop Distributed File System. The user program specifies a map function and a reduce function to process the data. After invocation, the master node coordinates the overall program execution in the distributed environment. It identifies two sets of worker nodes to execute the map task and the reduce task. The master node also determines a set of [key, value] pairs to be processed by each worker node. A map worker node accesses the assigned [key, value] pairs from the DFS and processes each using the user defined map function. It stores the computed [intermediate key,value] pairs in its local file system, and reports their location to the master node. The master node communicates these locations to the reduce worker nodes. A reduce worker node accesses the assigned [intermediate key, value] pairs from local file system of map worker nodes using remote procedure calls and processes each using user defined reduce function. The computed

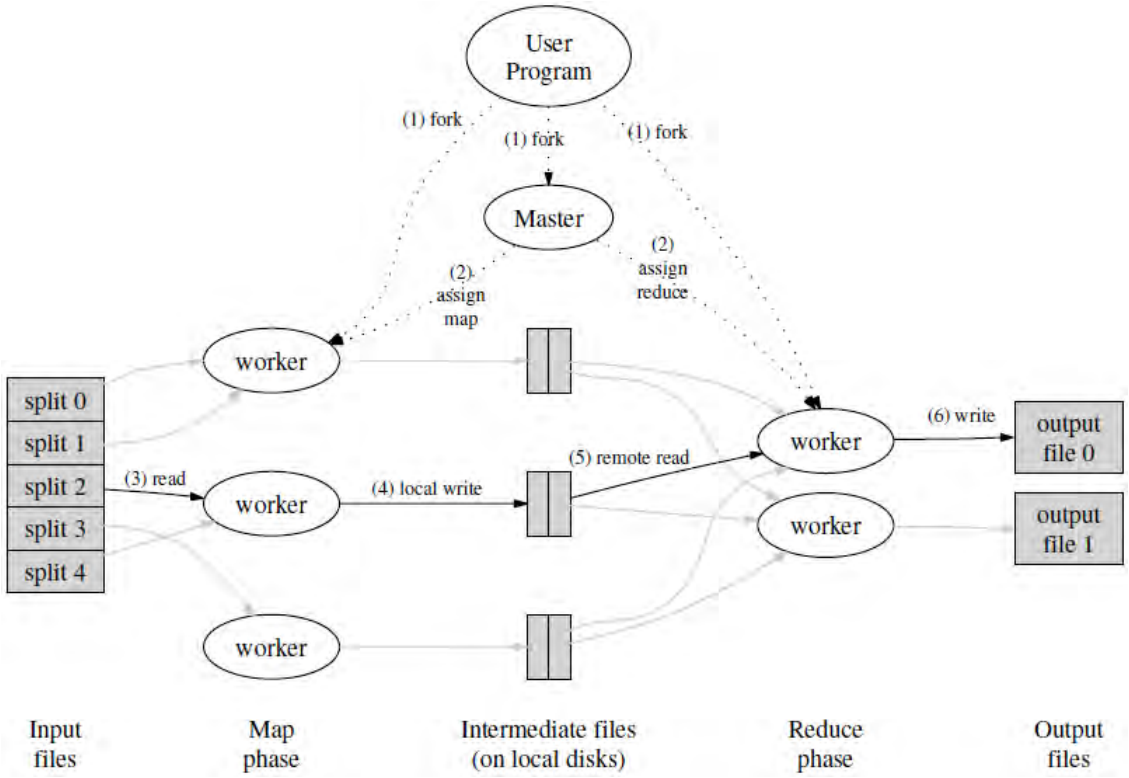


Figure 5.2: Program Execution under MapReduce Framework

final output is stored in the DFS. The MapReduce operation completes when all worker nodes complete the assigned task.

In case of a worker node failure, the master node reschedules the task and assigns the [key, value] pairs to other available node(s). In case of the master node failure, the program execution is aborted.

In this discussion the following terminology is used: a key-value pair is represented as [(key), (value)]. When a key or value has a single attribute, the enclosing brackets are optional.

5.3 Cellular Data Processing using MapReduce Framework

The framework takes as input the periodic position updates of all the vehicles and computes vehicle flow, space occupancy and congestion level data for every edge. Figure 5.3 shows a functional block diagram of the cellular data processing using the MapReduce framework.

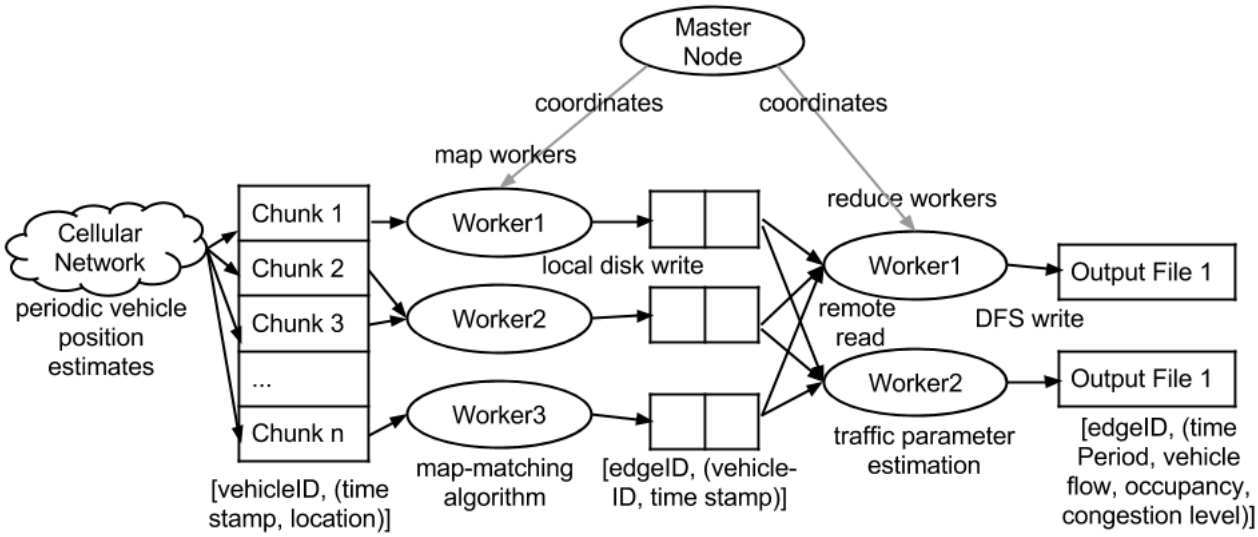


Figure 5.3: Cellular Data Processing using MapReduce Framework

It is assumed that the periodic position estimates for all the vehicles are available in the form of [vehicleID, (time stamp, location)] through active signaling in cellular network. The information is stored in the DFS and is accessible to all the nodes. The map matching algorithm is associated with the map function and the edge level traffic parameter computation is associated with the reduce function. The traffic parameters for every edge are stored as [edgeID, (timePeriod, vehicle flow, space occupancy, congestion level)] pair in the DFS. The

The MapReduce processing of the cellular data is elaborated in the following:

1. The master node identifies a set of worker nodes, M map worker nodes and R reduce worker nodes, that execute the map function and the reduce function, respectively. It also defines two hash functions, the one to map a vehicleID to a unique map worker node, and the other to map an edgeID to a unique reduce worker node.

2. A map worker receives the [vehicleID, (time stamp, location)] pairs for all the vehicles assigned to it and processes them using the map matching algorithm (section 3.2) to generate the vehicle trajectory in the form of [edgeID, (vehicleID, time stamp)] pair. It stores the data in its local file system, partitioned into R regions using edgeID. The locations of these files are communicated to the master, which then forwards the locations to reduce workers.

3. A reduce worker, after getting these locations, uses remote procedure calls to read

data of all the edgeIDs assigned to it from local disk of the map workers. The data is sorted by edgeID so that all occurrences of the same edgeID are grouped together. Sorting is required because typically many different edgeIDs are assigned to a reduce worker.

The reduce worker iterates through the set of values for given edgeID and computes vehicle flow, space occupancy and congestion level information using the method elaborated in section 3.3. The reduce worker writes [edgeID, (timePeriod, vehicle flow, space occupancy, congestion level)] pairs to an output file in the distributed file system. The data can be accessed by any node for further processing by querying the master node for the location of the data with key, edgeID.

5.3.1 Communication Overhead Analysis

Data communication is required at the following stages of MapReduce processing:

- A cellular network node (i.e. MSC- Mobile Switching Center) generates a location estimate of a vehicle and stores [vehicleID, (time stamp, location)] pair in the distributed file system. It is reported in the literature that active tracking of all the vehicles every 30 seconds is feasible with approximately 5% location capacity of the cellular carrier [15]. If the MSC and the data server are not collocated, the additional communication overhead is incurred to store the location estimates in the DFS. Let N_e be the total number of edges covered by an MSC, f_e be the number of vehicles served by an edge in unit time (vehicle flow), D be the periodicity of location updates, and S be the size of location record. The data generation rate is,

$$\frac{N_e \times f_e \times S}{D} \tag{5.1}$$

Typically, if an MSC covers 500 edges of a road network (approximately 2^9 edges), flow on 2-lane edges is approximately 2500-2800 vehicles per hour and that on 3-lane edges is approximately 3000-4000 vehicles per hour during peak hours (approximately 2^{12} vehicles per hour), D is 30 seconds, and S is 1 KB (2^{10} bytes), then less than 256 GB of data is generated every hour. The data is communicated over network if the MSC and the data server are not collocated. Communication of this

size data is feasible using available communication technology. For example, the gigabit Ethernet provides bandwidth of few giga bits per second for short distance communication (tens of kilometers). The optical fiber cables provide bandwidth of 100 tera bits per second over longer distances (hundreds of kilometers). The raw data is retained in the DFS only for the duration of computing traffic parameters (vehicle flow, space occupancy, etc.). For a typical data aggregation period of ten minutes, the storage capacity of less than 50 GB is sufficient.

- A map worker node fetches [vehicleID, (time stamp, location)] information from the DFS for all the vehicles assigned to it. A reduce worker node fetches [edgeID, (vehicleID, time stamp)] information from the local file system of map worker nodes for all the edges assigned to it. The communication overhead is of the same order as equation 5.1 and is distributed among M map worker nodes and R reduce worker nodes.

- The reduce worker node writes [edgeID, (timePeriod, vehicle flow, space occupancy, congestion level)] information in the DFS after every update period. The amount of data generated by each reduce worker in unit time equals $(N_e \times S)/(U \times R)$ bytes, where U is the update period. If there are 500 edges in a road network, and the traffic parameter data of 1KB size is recorded every minute, the storage of 720 MB per day is sufficient (less than 75 GB for keeping three months traffic data).

5.4 GPS Data Processing using MapReduce Framework

The GPS probe vehicles report their position updates periodically which are processed to compute edge level GPS probe speed. The COCOMO and ECOMO do not use the GPS probe speed data in real time. Instead, the average GPS probe speed is computed for all the edges and congestion levels using data collected over few days period (section 3.4). As the processing of GPS probe data is a non-real time task, it can be carried out when the load is less in the system, e.g. during off peak hours.

It is assumed that periodic position updates of GPS probes are available in the form of [vehicleID, (time stamp, location)] pair. The information is stored in DFS and is

accessible to all the nodes. The GPS probe trajectory generation is associated with the map function and the average GPS probe speed computation is associated with the reduce function. The MapReduce processing of GPS probe data is elaborated in the following:

- The master node identifies a set of worker nodes, M map worker nodes and R reduce worker nodes, that execute the map function and the reduce function, respectively. It also defines two hash functions, the one to map a vehicleID to a unique map worker node, and the other to map an edgeID to a unique reduce worker node.

  The set of worker nodes (and their role) and the hash functions can be the same as that used in the MapReduce processing of cellular network data. The use of same hash function and reduce worker nodes decreases communication overhead through the effective use of caching. As described in subsequent steps, a reduce worker node fetches the traffic parameters (computed using cellular data processing) of various edges from DFS. Using the same reduce worker node for cellular and GPS probe data processing of an edge permits caching of the computed traffic parameters.

- A map worker receives [vehicleID, (time stamp, location)] pairs for all the GPS probes assigned to it and generates their trajectory as [edgeID, (vehicleID, time stamp)] pairs. The map matching process is very simple as accurate position estimates are available. The map worker stores the data in its local file system, partitioned into R regions using edgeID. The locations of these files are communicated to the master, which then forwards the locations to reduce workers.

- A reduce worker, after getting these locations, uses remote procedure calls to read data of all the edgeIDs assigned to it from the local disk of map workers. The data is sorted by edgeID to group together all the occurrences of an edgeID.

  The reduce worker iterates through the set of values for given edgeID and computes average GPS probe speed for every time period. It writes [edgeID, (timePeriod, average speed of GPS probes)] pairs to an output file in the distributed file system.

  The following processing is done by a reduce worker node every few days. The reduce worker fetches [edgeID, (timePeriod, average speed of GPS probes)] and [edgeID, (timePeriod, congestion level)] information of the most recent N days from DFS (or from local disk if it is cached) to compute the average speed of GPS probes

for every congestion level on every edge. It sorts the data by edgeID and congestion level and iterates through the GPS probe speed values to compute the average. It writes [edgeID, a 6-tuple of (congestion level, average GPS probe speed)] pairs to an output file in the distributed file system.

5.4.1 Communication Overhead Analysis

Data communication is required at the following stages of MapReduce processing:

- A GPS probe vehicle periodically communicates [vehicleID, (time stamp, location)] to a data server. Let N_e be the total number of edges covered by a data server, f_e be the number of vehicles served by an edge every hour (vehicle flow), P_G be the penetration of GPS probes in the vehicle population, D be the periodicity of location updates, and S be the size of location record. The data generation rate is,

$$\frac{N_e \times f_e \times P_G \times S}{D} \tag{5.2}$$

  Typically, if a data server covers 500 edges of a road network (approximately 2^9 edges), flow on 2-lane edges is approximately 2500-2800 vehicles per hour and that on 3-lane edges is approximately 3000-4000 vehicles per hour during peak hours (approximately 2^{12} vehicles per hour), D is 1 second, penetration of the GPS probes in vehicle population is 5%, S is 1 KB (2^{10} bytes), then less than 360 GB of data is generated every hour. While the storage of few tera bytes is available in personal computers, storage requirement can be significantly reduced with simple compression techniques. Alternatively, considering the low complexity of GPS data processing, the MapReduce processing of GPS probe data can be done in real time along with cellular network data. In that case, for an aggregation period of ten minutes, the storage capacity of less than 60 GB is sufficient.

- A map worker node fetches [vehicleID, (time stamp, location)] information from the DFS for all the vehicles assigned to it. A reduce worker node fetches [edgeID, (vehicleID, time stamp)] information from the local file system of map worker nodes for all the edges assigned to it. The communication overhead is of the same order as equation 5.2 and is distributed among M map worker nodes and R reduce worker

nodes.

- The reduce worker node writes [edgeID, (timePeriod, average speed of GPS probes)] information in the DFS after every update period. The amount of data generated by each reduce worker in unit time equals $(N_e \times S)/(U \times R)$ bytes, where U is the update period. If there are 500 edges in a road network, and the traffic parameter data of 1KB size is recorded every minute, the storage of 720 MB per day is sufficient (less than 75 GB for keeping three months traffic data). Note that in a typical implementation, the edge level traffic parameters computed using cellular network and GPS probes are stored together and consumes 720 MB per day. A tuple of 1 KB size is sufficient to contain all the traffic parameters.

The following communication overhead is incurred every few days to update average GPS probe speed associated with a congestion level for every edge. A reduce worker node fetches [edgeID, (timePeriod, average speed of GPS probes)] and [edgeID, (timePeriod, congestion level)] information of the most recent N days from DFS. The amount of data fetched by each reduce worker equals $(N \times N_e \times S)/(U \times R)$ bytes. After computing the updated values, a reduce worker writes [edgeID, 6-tuple of (congestion level, average GPS probe speed)] to the DFS. The amount of data communicated by each reduce worker equals $(N_e \times S)/R$ bytes, and is negligible. The process is repeated by a reduce worker node every few days.

5.5 Speed Estimation using MapReduce Framework

5.5.1 Computing Congestion Profile of Edges

This computation is done offline when the system is lightly loaded. The master node selects R reduce worker nodes for computing the congestion profile of edges using the N days' data. A reduce worker processes the [edgeID,(timePeriod, congestion level)] pairs to compute congestion profile of the edges assigned to it and stores [edgeID, 6-tuple of the congestion profile] in DFS. The amount of data fetched by each reduce worker node from the DFS equals $(N \times N_e \times S)/(U \times R)$ bytes. The total amount of data retrieved over network equals 7.2 GB for $N = 10$, $N_e = 500$, $S = 1$ KB, and $U = 1$ minute. The

amount of data stored by each reduce worker node to the DFS equals, $(N_e \times S)/R$ bytes, and is negligible.

5.5.2 Infrastructure Edge Selection

This computation is done by the master node centrally to determine a set of edges for infrastructure deployment. The master node fetches the congestion profile of edges from DFS (amount of data read equals $N_e \times S$ bytes) and selects a set of edges for infrastructure deployment using the COCOMO (section 3.5.1) or ECOMO (section 3.5.2). Also, the set of edges or congestion levels covered by each infrastructure edge are determined.

In case of COCOMO, the master node stores [infrastructure edgeID, a set of covered congestion levels] and [congestion level, set of infrastructure edgeIDs] in the DFS. The amount of data written to DFS equals $(N_e^i \times S) + (6 \times S)$ bytes, where N_e^i is number of infrastructure edges and there are six congestion levels. In case of ECOMO, the master node stores [infrastructure edgeID, set of covered edges] and [edgeID, set of infrastructure edgeIDs] in the DFS. The amount of data written to DFS equals $(N_e \times S)$ bytes. The communication overhead of this operation is negligible and is incurred only once during initial deployment of the system.

The master node also computes a speed transition function using the [edgeID, 6-tuple of (congestion level, average GPS probe speed)] pairs. The amount of data read from DFS equals $(N_e \times S)$ bytes. As elaborated in section 3.5.3, the speed transition function is used to spatially extrapolate the speed estimation from an infrastructure edge to the corresponding infrastructureless edge. The information is stored as [(infrastructure edgeID, covered edgeID), 6-tuple of (congestion level, speed transition function value)] in the DFS. The amount of data written to DFS equals $(N_e \times K \times S)$ bytes assuming K-coverage of all the edges. The communication overhead for this operation is negligible and is incurred every time average GPS probe speed data is recomputed.

5.5.3 Learning Occupancy-Speed Relationship

The infrastructure edges generate [edgeID, (time period, aggregate speed)] information and store in the DFS.

A reduce worker node uses the [edgeID, (time period, space occupancy, congestion

level)] and [edgeID, (time period, aggregate speed)] information to learn the occupancy-speed relationship for all the infrastructure edges assigned to it.

In case of COCOMO, the reduce worker sorts the data by edgeID and iterates through occupancy-speed values to learn the polynomial regression model for every congestion level covered by the edge. It stores [edgeID, 6-tuple of (congestion level, regression parameters of occupancy-speed relationship)] information in the DFS.

In case of ECOMO, the reduce worker sorts the data by edgeID and iterates through occupancy-speed values to learn the polynomial regression model for the edge. It stores [edgeID, regression parameters of the occupancy-speed relationship] information in the DFS.

It should be noted that the computation is done offline using data of the most recent N days and is repeated every few days. Let N_e^i be the number of infrastructure edges, R be the number of reduce worker nodes, U be the periodicity of the edge level data collection, and S be the size of the data record. The amount of data fetched by each reduce node equals $(N_e^i \times N \times S)/(U \times R)$ bytes. The total amount of data retrieved over the network is less than 1.5 GB in case of $N = 10$, $N_e^i = 100$, $S = 1$ KB, and $U = 1$ minute. The amount of data stored by each reduce worker node to the DFS equals $(N_e^i \times S)/R$ bytes, and is negligible.

5.5.4 Real time Speed Estimation

This task is done by the reduce worker nodes for every infrastructureless edge after computing the space occupancy, and congestion level data for the period.

A reduce worker node fetches,

- [edgeID, set of infrastructure edgeIDs],
- [infrastructure edgeID, 6-tuple of (congestion level, regression parameters of occupancy-speed relationship)] (for COCOMO) or [infrastructure edgeID, regression parameters of occupancy-speed relationship] (for ECOMO), and
- [(infrastructure edgeID, covered edgeID), 6-tuple of (congestion level, speed transition function value)]

from the DFS, and computes the speed estimation for the period. It stores speed estimation data along with the other edge level parameters for the period in the DFS. For efficiency purpose, the reduce worker node caches all of the above information in local file system and uses it for speed estimation. Hence, no communication cost is incurred for fetching the data from DFS. The estimated speed for the period is written to the DFS. The data generation rate is $(N_e \times S)/(U \times R)$ bytes. The total amount of data communicated over network by all the reduce worker nodes equals 720 MB for $N_e = 500$, $S = 1$ KB, and $U = 1$ minute.

5.6 Discussion

The previous sections elaborated a methodology for cellular data processing, GPS probe data processing and real time speed estimation using the MapReduce framework. The communication and storage requirement of the framework is analyzed. The vehicle trajectory generation and traffic parameters' computation (vehicle flow, space occupancy, congestion level, and speed) using cellular network require real time processing. The GPS probe data processing to compute edge level GPS probe speed can be done offline but real time processing is preferred in the interest of reducing storage requirement and due to the less complex computation requirement. The computation of congestion profile of edges, occupancy-speed relationship on infrastructure edges and speed transition function is done offline every few days.

In the subsequent discussion, it is assumed that the number of edges (N_e) in the road network are 500, the number of infrastructure edges (N_e^i) are 100, the cellular data sampling period is 30 seconds and that of GPS probes is 1 second, the aggregation period is 10 minutes, the update period (U) is 1 minute, and the size of data record (S) is 1 KB.

The raw data of vehicle position collected from cellular network and GPS probes represent a significant component of the overall communication and storage requirement of the framework. The real time cellular network data processing requires communication and storage of less than 50 GB per aggregation period. The GPS probe data, when processed in real time (preferred), requires communication and storage of about 60 GB per aggregation period. The specified amount of cellular network data needs to be communicated over network only if the MSC and the data server(s) are not collocated. In

that case a dedicated Ethernet link or a fiber link can be used to carry the data. As the raw data is in text form, the use of simple data compression techniques can significantly reduce the communication requirement. The specified amount of GPS probe data comes to data server(s) from whole road network over Internet. A GPS probe reporting its position every second communicates 3.6 MB of data during its trip of an hour. The current wireless communication technology (WiFi or cellular data connection) readily supports the required data rate. A backbone network connecting the data server(s) to Internet can easily carry 60 GB of data during an aggregation period (a Gbps link is sufficient). As the storage for raw data of cellular network and GPS probes is required only for an aggregation period, the storage capacity of 110 GB is sufficient.

The computation of traffic parameters in real time is carried out in a distributed manner using the MapReduce framework. The traffic parameters are recorded every update period for all the edges in a road network. The communication and storage requirement is 720 MB per day (less than 75 GB for storing three months traffic data).

The computation of congestion profile of edges, speed transition function and occupancy-speed relationship is done every few days using historical traffic data stored in the DFS. If traffic data of ten days are used for the computation, the amount of data retrieved equals 7.2 GB. The computed information collectively consumes less than 1 MB of storage space.

The analysis of communication, computation, and storage requirements shows the feasibility of large scale deployment of the proposed multi-modal ITS.

Chapter 6

Application: Advanced Traveler Information System

6.1 Background

In the previous chapters a multi-modal Intelligent Transportation System (ITS) is designed that generates edge level traffic information (vehicle flow, congestion, speed) in real time for the whole road network. The two models, namely COngension COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO) are developed for ITS infrastructure deployment and edge level speed estimation. The performance of the proposed ITS is evaluated for quality of generated traffic information (error in traffic parameters' estimation) and availability of traffic information (spatio-temporal coverage). Also, the feasibility of large scale deployment of the ITS is established by designing a distributed computing framework and analyzing computation, communication and storage requirement.

The network wide real time traffic information can be used by traffic applications for a variety of purposes. For example, an Advanced Traveler Information System (ATIS) uses real time traffic information to help the commuters in trip planning. An Advanced Traffic Management System (ATMS) uses traffic information for adaptive traffic light control, enforcing speed limits and lane control, or suggesting diversions to avoid congested regions in a road network. The Advanced Public Transportation System (APTS) uses real time traffic information to estimate the arrival time of public transport buses at different

stations and plan trip schedule of buses.

The aim of this chapter is to assess the utility of the real time traffic information generated by the proposed ITS. As a proof of concept, an Advanced Traveler Information System (ATIS) is designed that uses the traffic information generated by the proposed ITS. The traffic information has a certain amount of error, and also, the traffic information may not be available for a set of congestion levels (COCOMO) or for a fraction of edges (ECOMO). This chapter analyzes the effect of these parameters and ATIS penetration on the average trip duration and congestion distribution in a road network.

The chapter is organized as follows: design of the ATIS is discussed in section 6.2; simulation model of the application is elaborated in section 6.3; section 6.4 discusses analysis of the simulation results; the chapter is concluded in section 6.5.

6.2 Introduction

With the availability of real time traffic information, the commuters can make informed decisions to optimize their trips (in terms of travel time, driving comfort, fuel consumption, etc.). An Advanced Traveler Information System (ATIS) has potential to make the individual commuter and the road network as a whole more productive [75].

Bertini et al. [76] survey various ITS deployments in different parts of the world and analyze their performance. The authors evaluate freeway and arterial management systems, traveler information and transit management systems, incident or emergency management systems, and various crash prevention and safety systems for their deployment cost and effectiveness in improving the traffic condition. While an ITS deployment improves traffic condition in general, the authors observe that the quantification of ITS benefits has not been statically sophisticated. For example, when an ITS claiming certain cost benefit in one city is deployed in another city, there is no guarantee that the similar benefits will be achieved.

A large number of ATIS applications that consume real time traffic information generated by an ITS are reported in the literature. Xiao and Lo [77] develop an ATIS that processes real time traffic information updates to enable re-routing of vehicles during their commute. The authors use probabilistic dynamic programming and claim better performance than deterministic routing specifically for road networks with highly vari-

able travel times. It is assumed that the penetration of application is low and the overall network traffic assignment is not affected.

Kumar et al. [78] propose a Geographic Information System (GIS) based ATIS for Hyderabad city in India. The authors report twelve modules of their system, such as public transport services (buses, rail, air) in the city, searching facilities in the city, a distance based shortest path for Origin-Destination pair (O-D pair), etc.. The current version of the ATIS does not use real time traffic information, but the extension is feasible.

Ben-Elia et al. [79] analyze the effect of traffic information accuracy and travel time uncertainty on route choice made by ATIS users. The authors show using experiments that the decrease in accuracy shifts choices from the riskier route (lower mean travel time with high variation) to the reliable route (higher mean travel time with low variation).

We design an ATIS that uses the real time speed estimations generated using the proposed models (COCOMO or ECOMO). The proposed models enable real time estimation of edge level traffic parameters (vehicle flow, space occupancy, congestion level, and speed). However, the generated traffic information has the following properties: (i) it has a certain amount of error, and (ii) the traffic information may not be available for some fraction of edges or for some congestion levels due to limited infrastructure deployment. The application considers the above properties in its design.

The application classifies the road commuters in two categories: the informed commuters (intelligent vehicles) receive traffic information updates every minute and adapt their routes as suggested by the system; and the uninformed commuters (non-intelligent vehicles) do not receive or process the traffic information updates and always follow the shortest distance path.

Algorithm 6.1: Route Query Processing by the ATIS Server

Input: a query (s, d), where s specifies current location of the vehicle, and d is the intended destination, road network topology, G, containing travel time estimation for all the edges

Output: an optimal route $P_{s,d}$ from s to d along with travel time estimate $T_{s,d}$

1: $P_{s,d}$ = Dijkstra(s, d, G)

2: $T_{s,d} = \sum_{e \in P_{s,d}} \frac{length(e)}{speed(e)}$

The application is simple and less computationally intensive. It does not make any forecast about the changes in traffic condition in road network due to rerouting of the

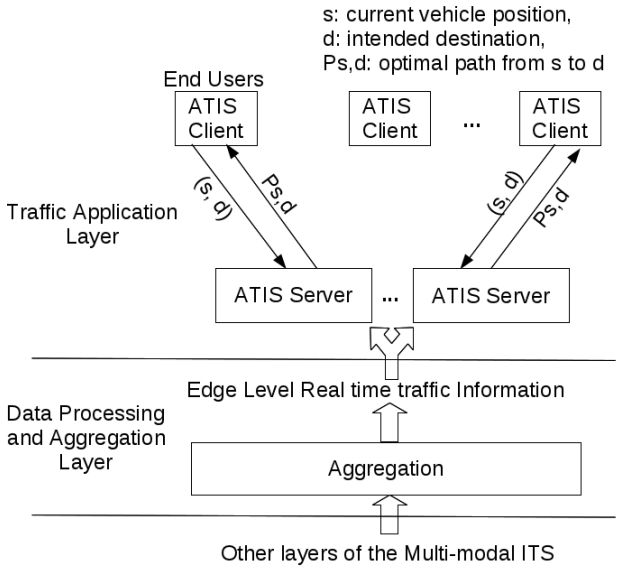


Figure 6.1: Advanced Traveler Information System

Algorithm 6.2: Route Query Response Processing by the ATIS Client

Input: $P_{s,d}^{cur}$, $T_{s,d}^{cur}$, $P_{s,d}^{new}$, $T_{s,d}^{new}$, g_{th}, where $P_{s,d}^{cur}$ and $T_{s,d}^{cur}$ are current path and corresponding travel time, respectively, $P_{s,d}^{new}$ and $T_{s,d}^{new}$ are the new path and corresponding travel travel time received in the query response, g_{th} is the minimum gain threshold

Output: $P_{s,d}$, the path to be followed by the vehicle

1: **if** $T_{s,d}^{new} < (1 - g_{th}) \times T_{s,d}^{cur}$ **then**
2: $\quad P_{s,d} = P_{s,d}^{new}$
3: $\quad T_{s,d} = T_{s,d}^{new}$
4: **end if**

informed commuters, and uses only current traffic state of the road network for computing and suggesting alternate routes. This simple approach works well because different vehicles make rerouting decisions at different times: every vehicle selects the most optimal path at the beginning of its trip, and makes rerouting decision when it is about to reach a junction (about to complete the current edge traversal). Also, a vehicle changes route only when there is significant gain - travel time improves at least by a fraction of g_{th}. $g_{th} = 0.05$ is used in the simulations.

Figure 6.1 shows a schematic diagram of the ATIS. An ATIS client queries the server with its current position and intended destination. The ATIS server computes an optimal route for the query using real time traffic information available from the multi-modal ITS and sends the query response. The Dijkstra's shortest path algorithm is used to compute the shortest path with respect to travel time from vehicle's current position to

its destination. Algorithm 6.1 and 6.2 show the steps carried out by the ATIS server and client for a query.

6.3 Simulation

The application is simulated using Simulator for Urban MObility (SUMO) version 0.22 [80]. The Traffic Control Interface (TraCI) of SUMO is used to dynamically change the route of vehicles. The simulation scenarios are described in Table 6.1. The simulations are done using city wide large scale scenarios namely, Grid network (S4) and Random network (S5). The scenarios are same as used in chapter 4 for evaluating accuracy of traffic parameters computed by the proposed ITS. The heterogeneous traffic dynamics of peak hours and off peak hours are simulated. The same vehicle distribution as specified in chapter 4 (Table 4.1) is used. All data processing is done using Python version 2.7.3 scripts.

A set of informed commuters (intelligent vehicles) is randomly selected at the beginning of simulations according to the penetration of ATIS among users. An intelligent vehicle gets optimal path from the ATIS server at the beginning of its trip. It also receives and processes route updates from the ATIS server. If an alternate path suggested by the server has travel time gain more than g_{th}, the vehicle reroutes to the alternate path from next junction. $g_{th} = 0.05$ is used in the simulations. The other vehicles (uninformed commuters) follow the shortest distance path statically assigned to them at the beginning of the trip.

Simulations are carried out by varying application penetration from 0% to 50% (in increments of 10%) to evaluate its effect on the average trip duration and congestion distribution in the road network.

The following speed error model is used in the application: for the congestion levels or edges covered by infrastructure edges (i.e. periodic speed estimation is available), a normal random variate $N(\bar{\epsilon}, \sigma_\epsilon)$ is generated as the speed error, where $\bar{\epsilon}$ equals the mean speed error of the model (COCOMO or ECOMO) and σ_ϵ is selected such that,

$$\text{ninety percentile speed error} = \bar{\epsilon} + 1.65 \times \sigma_\epsilon \tag{6.1}$$

The speed error values for COCOMO and ECOMO are taken from simulation results in section 4.5 and 4.6, respectively. A uniform random variate is used to decide whether the speed error is added to or subtracted from the correct speed. If the uniform random variable is less than 0.5, the speed error is added to the correct speed, otherwise it is subtracted.

Due to limited infrastructure deployment, the real time speed estimation may not be available for a set of congestion levels (COCOMO) or for a fraction of edges (ECOMO). For edges or congestion levels not covered by infrastructure edges (real time speed estimates are not available), the real time congestion level data along with historical data of GPS probe speed are used: the average GPS probe speed on the edge for the congestion level is used as a real time speed estimate.

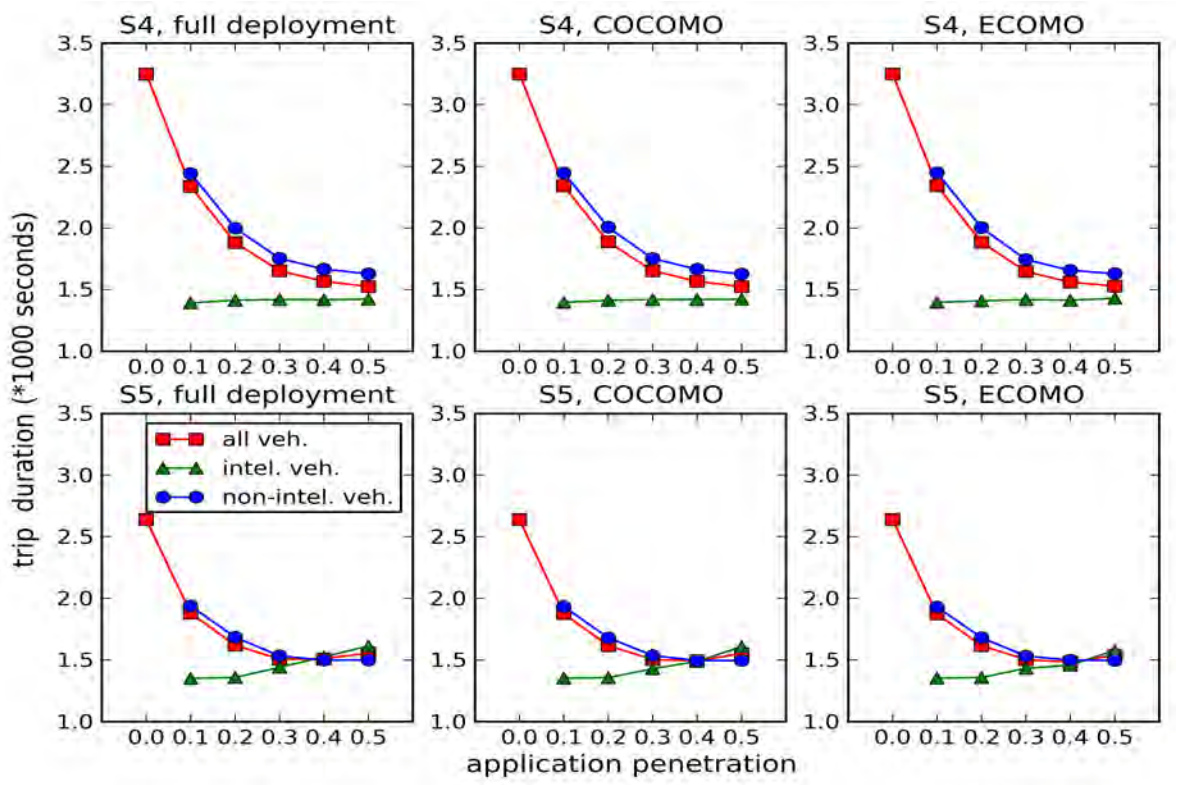
Table 6.1: Simulation Scenarios

Scenario Name	Description	Location error	Vehicle Insertion Rate
Grid Network (S4)	a $15km \times 15km$ road network with road segment length of 1km each, with traffic light at each junction	250m, 500m	2100-4200 veh/hr in seven merging/diverging flows for peak and off peak duration of 14 hrs (total 40950 vehicles)
Random Network (S5)	a $18km \times 16km$ road network with road segment length of 0.5-1.5km each, with traffic light at each junction	250m, 500m	2400-4800 veh/hr in eight merging/diverging flows for peak and off peak duration of 14 hrs (total 46800 vehicles)

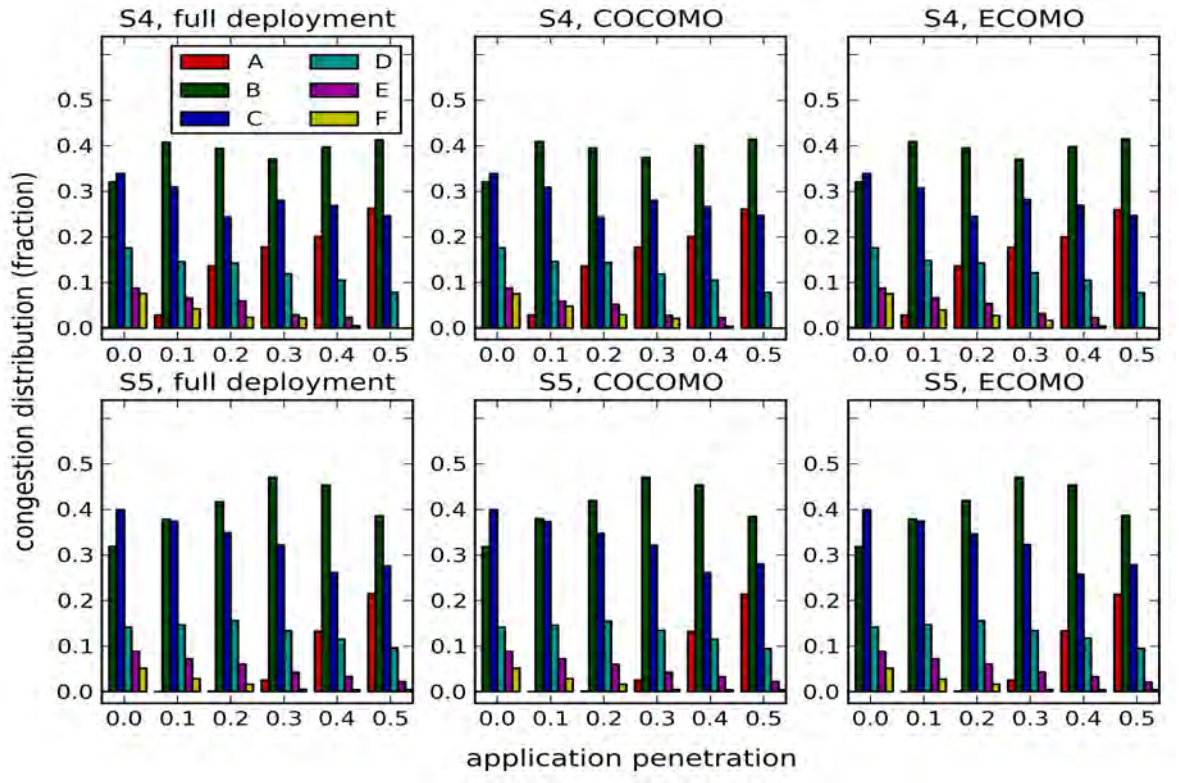
6.4 Analysis of Simulation Results

Simulations are carried out to evaluate the effect of application penetration and speed estimation error on the average trip duration and congestion distribution in a road network. The metrics are compared for the full deployment model and the proposed models (COCOMO and ECOMO). In the full deployment model, ITS infrastructure and hence accurate speed estimation is available for all the edges in the road network.

Figure 6.2a shows the effect of application penetration on average trip duration of intelligent vehicles, non-intelligent vehicles (uninformed commuters), and all the vehicles for full deployment model, COCOMO and ECOMO. Figure 6.2b shows the effect of application penetration on congestion distribution in the road network, for all the scenarios.



(a) Effect on Trip Duration



(b) Effect on Congestion Distribution

Figure 6.2: Effect of Application Penetration

Following are the important observations:

- The variations in trip duration and congestion distribution with ATIS penetration for COCOMO and ECOMO are comparable to the full deployment model. The edge level speed error has little or no impact on the rerouting decisions of the vehicles. The reason being, when trip duration is computed by adding the erroneous travel time estimates on individual edges, a certain fraction of errors cancels out, leading to accurate trip duration calculation. That is, the accurate trip duration estimation is available for making rerouting decision in all the scenarios. The results are in agreement with other works in literature, such as VTrack [47] wherein it is observed that accurate suggestions for shortest route are feasible even with erroneous (25% median error) edge level travel time estimations. The ATIS works effectively with erroneous traffic information and establishes utility of traffic information generated by the proposed models.

- With low penetration of the ATIS (e.g. 10%), intelligent vehicles get the most benefit with reduction in trip duration by more than 50% in the grid network (S4), and more than 45% in the random network (S5). Overall trip duration in the network (including that of uninformed commuters) also improves with reduction in trip duration by more than 25% (more than 20% for non-intelligent vehicles) in S4 and S5. This happens because of the fact that as the intelligent vehicles choose better alternate route to improve their trip, traffic condition on their original route also become better leading to improved trip duration for non-intelligent vehicles as well.

- As ATIS penetration increases, benefits to the intelligent vehicles decrease, but still they achieve reduction in trip duration by more than 40% in S4 and S5.

  For ATIS penetration of 30-50%, the average trip duration of intelligent vehicles and non-intelligent vehicles are comparable. In some of the cases (application penetration of 40-50% in S5), average trip duration of uninformed commuters is slightly better than that of intelligent vehicles. The careful analysis of simulation traces revealed some interesting facts. With increase in application penetration beyond 20%, the overall network traffic assignment is affected. As mentioned earlier, the

ATIS considers only current traffic state for suggesting alternate route to intelligent vehicles. It does not make any forecast about resultant traffic state due to rerouting of intelligent vehicles. With a large number of intelligent vehicles making rerouting decisions, the traffic conditions on alternate routes also change which is not considered by the ATIS while suggesting the alternate route. This results in intelligent vehicles taking longer time than expected for completing their trip. The similar observations are reported by other researchers as well (e.g. [75] and [81]).

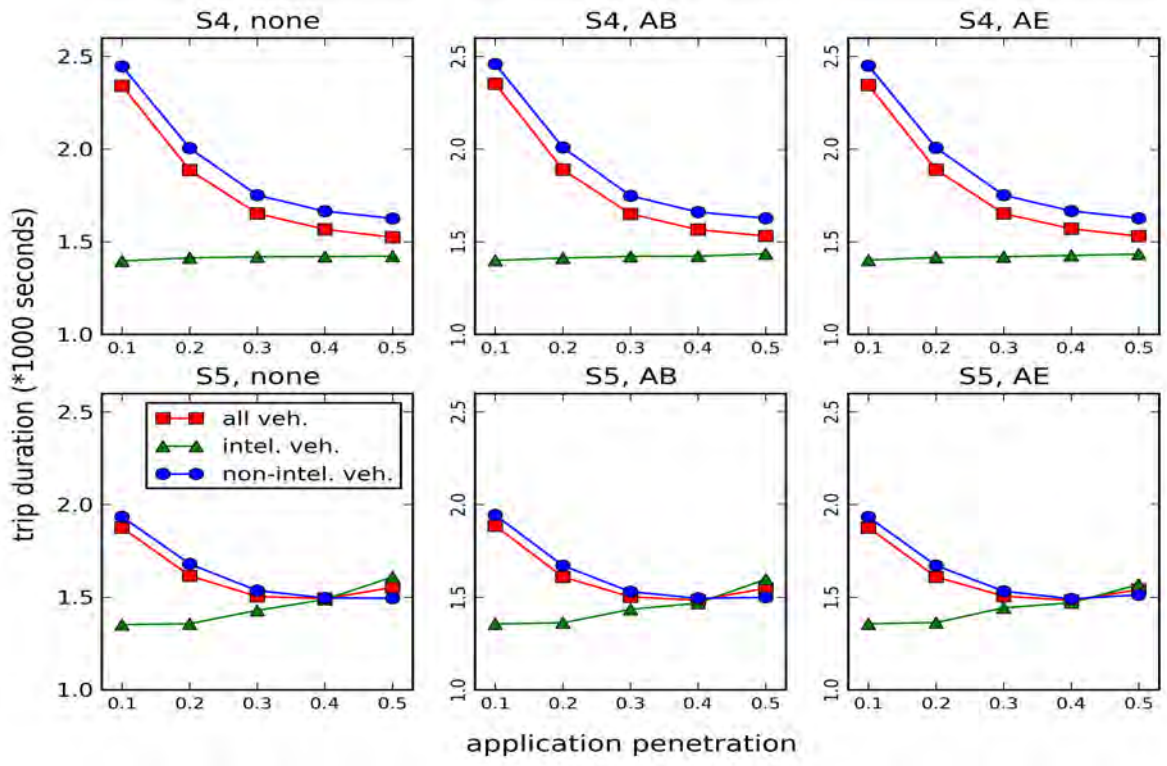
We do not consider it as a limitation of the ATIS due to two reasons: first, the information about the fraction of vehicles changing their route based on suggestions from the ATIS is difficult to determine as commuters themselves make a final decision about the route to be taken for a trip; second, as the ATIS has to respond to a large number of queries in real time, the use of computation intensive forecast mechanisms based on machine learning may not be feasible.

- Figure 6.2b summarizes the effect of ATIS penetration on congestion distribution in a road network for full deployment model, COCOMO and ECOMO. The effect of application penetration is similar in all the cases. The availability of the traffic information improves congestion in the road network.

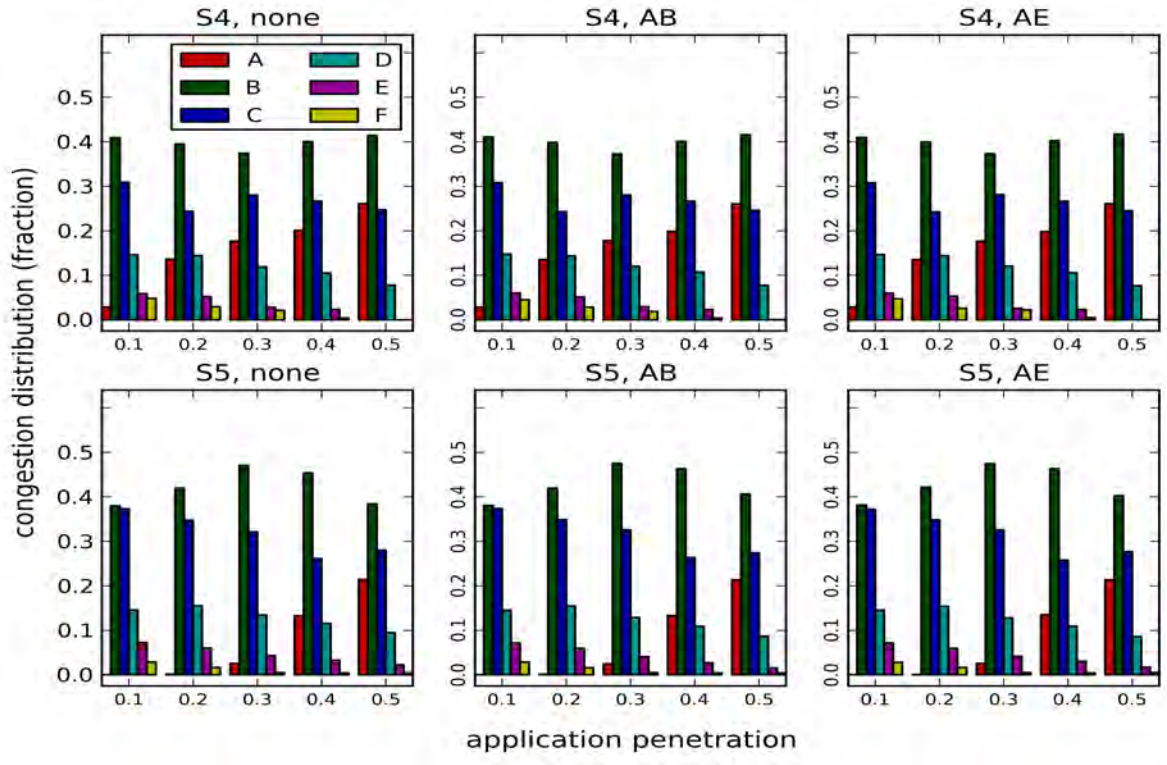
  Without ATIS, the road network faces congestion level F (unstable traffic regime) in about 8% cases (considering time and space) in grid network scenario (S4) and in 5% cases in random network scenario (S5). With the increase in penetration of ATIS, the occurrence of congestion level F reduces and reaches zero percent with application penetration of 40-50%.

  The occurrence of congestion level E reduces from 8-9% without ATIS to 0-2% with ATIS penetration of 40-50%. The traffic condition in road network moves towards better congestion levels (A-D) as the application penetration increases.

Figure 6.3 shows effect of not covered congestion levels (COCOMO) on trip duration and congestion level distribution in scenario S4 and S5. A set of not covered congestion levels is taken from section 4.5.2 where the effect of unavailable ITS infrastructure on congestion level coverage is analyzed. It is observed that there is little or no impact of non-coverage of congestion level(s) on trip duration (Figure 6.3a) and congestion distri-



(a) Effect on Trip Duration



(b) Effect on Congestion Distribution

Figure 6.3: COCOMO: Effect of Not Covered Congestion Levels (subplot title specifies scenario and not covered congestion level(s))

bution (Figure 6.3b) and the results are comparable to the scenario when full coverage is available.

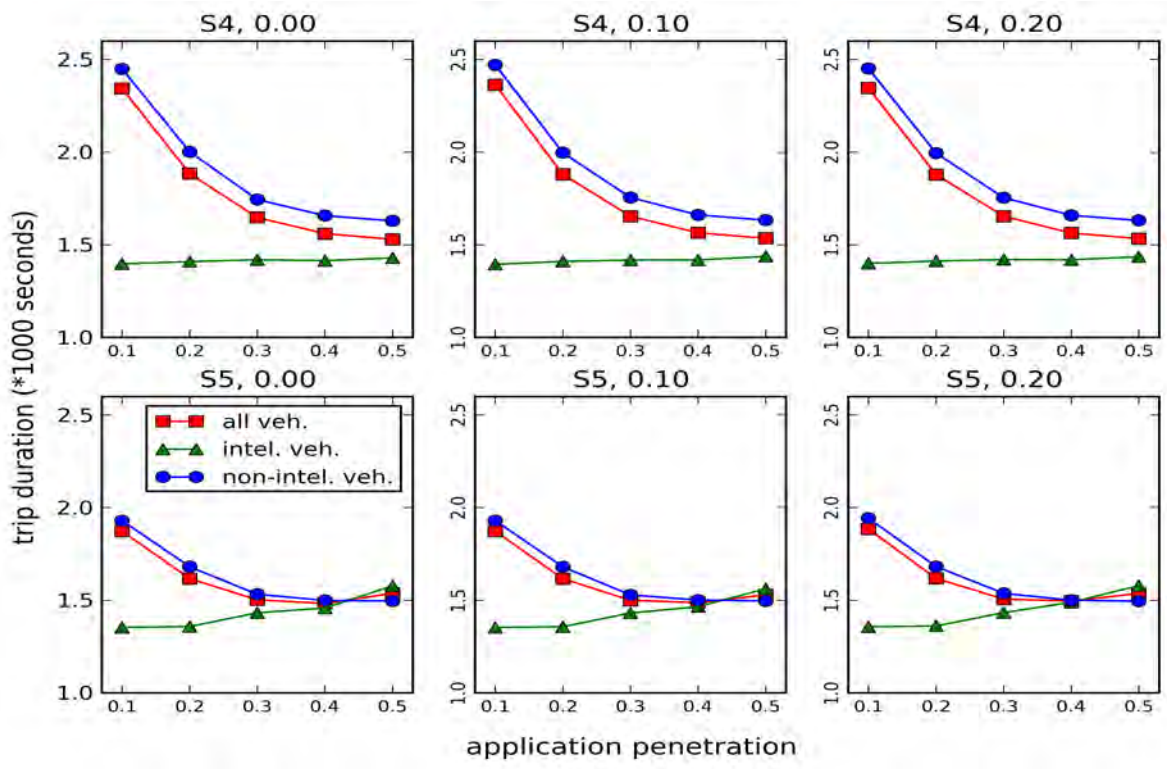
The non-coverage of a congestion level leads to unavailability of real time speed information for all the occurrences of the congestion level in a road network. As mentioned earlier, in all such cases, the corresponding historical data of GPS probe speed is used. While the GPS probe speed is not a good representative of all vehicle speed on an edge, it incorporates the effect of current congestion level on vehicle movement speed. As the real time speed information is available for all the covered congestion levels, the use of GPS probe speed on few edges does not affect the overall trip duration computation. Hence, even with few not-covered congestion levels, a reasonably accurate trip duration estimation is available for making rerouting decision. The historical data of GPS probe speed combined with real time congestion level serves as a good substitute in case of unavailability of real time speed estimation.

Another observation made from simulation traces is that when the trip duration of alternate routes for a trip, each having few edges observing a not-covered congestion level, are compared, usually the resultant path selection is the same as the case when real time traffic information is available for all the congestion levels.

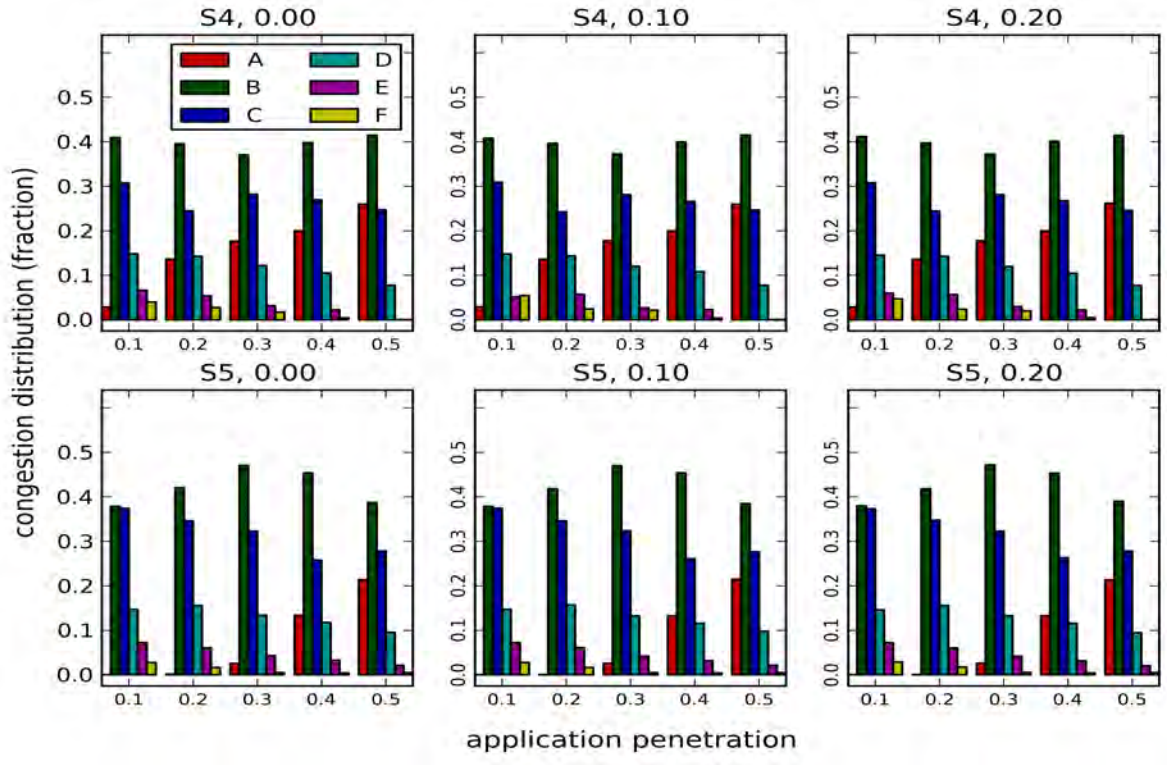
Figure 6.4 shows effect of not covered edges (ECOMO) on trip duration and congestion distribution in scenario S4 and S5. The fraction of not covered edges is varied from 0.00 to 0.20 in increment of 0.10 in scenario S4 and S5. The range is selected based on analysis in section 4.6.2 where the effect of unavailable ITS infrastructure on edge coverage is analyzed. It is observed that there is little or no impact of non-coverage of edges on trip duration (Figure 6.4a) and congestion distribution (Figure 6.4b) and the results are comparable to the scenario when full coverage is available.

When an edge is not covered through infrastructure edges, the real time speed estimations for the edge are not available. As mentioned earlier, in all such cases, the historical data of GPS probe speed associated with the real time congestion level on the edge is used. As the real time speed information is available for all the covered edges, the use of GPS probe speed on few edges does not affect the overall trip duration computation. Hence, even with few not-covered edges, a reasonably accurate trip duration estimation is available for making rerouting decision.

It is also observed from simulation traces that when the trip duration of alternate



(a) Effect on Trip Duration



(b) Effect on Congestion Distribution

Figure 6.4: ECOMO: Effect of Not Covered Edges (subplot title specifies scenario and fraction of not covered edges

routes for a trip, each having few not-covered edges, are compared, usually the resultant path selection is the same as the case when real time traffic information is available for all the edges.

6.5 Conclusion

This chapter elaborated the design of an Advanced Traveler Information System (ATIS) using traffic information generated by the proposed models (COCOMO and ECOMO). Simulation results show that the ATIS improves average trip duration and congestion in the road network and the performance is comparable to the full deployment of ITS infrastructure. Hence, we conclude that the useful traffic information can be generated using limited infrastructure deployment in a road network.

The effect of unavailability of traffic information for a set of congestion levels (COCOMO) or a fraction of edges (ECOMO) in a road network on trip duration and congestion distribution is evaluated. Simulation results show that the application tolerates unavailability of real time speed information and shows improvement in trip duration and congestion distribution which are comparable to the scenario when all the congestion levels or edges are covered. The historical data of GPS probe speed combined with real time congestion level plays an important role in case of unavailability of real time speed estimation.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Due to urbanization, the vehicular traffic has increased tremendously in the road network. The available road infrastructure is stretched to its capacity, specifically during peak hours. Due to increased traffic congestion, the average time spent on the road by commuters has increased significantly. An Intelligent Transportation System (ITS) generates fine grained vehicular traffic information in real time which can be used to optimize traffic movement in a road network. It has potential to reduce traffic congestion and make the commute more efficient and safer.

However, deployment and maintenance of ITS sensors and related infrastructure on all the edges in a road network is costly. With this reasoning, a variety of alternative sources have been examined in literature for quality of generated real time traffic information. The cellular network is widely deployed world wide but has large location error (150-500 meters) reported in the literature. When these location traces are used for generating vehicle trajectories, the edge level speed estimation has a mean error of more than 15%. We observe that most of the related work focus on processing of cellular signaling data (to reduce location error) with very little emphasis on designing algorithms for generating traffic information.

The GPS probe data are shown to be a feasible source of traffic information for road networks carrying relatively homogeneous traffic (only four wheelers). However, in developing countries like India, where the traffic is heterogeneous and dominated by

majority of GPS-less two wheelers, the GPS probes data alone is not a viable solution for generating real time traffic information.

In the literature, a variety of sensors (e.g. video cameras, RF transceivers, acoustic sensors, accelerometers, etc.) are evaluated for their deployment feasibility and quality of generated traffic information. However the deployment and maintenance cost of the dedicated sensors is high and is not feasible for developing countries.

The state of the art also contains works on fusion of traffic data from multiple data sources to improve accuracy and coverage of the generated traffic information. We observe that fusion of cellular network data with other data sources is not reported in the literature to the best of our knowledge. We consider it as a major gap in the literature due to large scale availability of cellular network and inability of generating accurate traffic information using cellular network data alone. Based on this observation, the following goal was set up for this thesis: to generate accurate traffic information in real time with minimum additional ITS infrastructure while exploiting the large scale availability of cellular network data and GPS probe data.

As a first step towards fulfilling the goal and to assess the cellular environment in India, a localization experiment was set up. An Android application was developed to collect cellular network traces and corresponding GPS location traces. The data collection was done for ten days on an 18 km long road stretch in Ahmedabad, India. The data set was processed using fingerprinting based localization algorithm. The mean location error of less than 50 meters is observed in a region with dense deployment of cellular infrastructure and less than 200 meters is observed in a region with relatively sparse deployment of cellular infrastructure.

To generate vehicle trajectories using erroneous location data collected from cellular network, a map matching algorithm was developed. To our surprise, no map matching algorithm was found in the literature that processes location data with such high error presented by cellular network. The map matching algorithm processes a series of location estimates of a vehicle before associating it to an edge. This enables the accurate vehicle trajectory generation but also adds a time lag. The vehicle trajectory is used to compute vehicle flow, space occupancy and congestion estimation. To remove the time lag in the traffic information and to make it real time, the temporal extrapolation using exponential moving average and polynomial regression is attempted. The simulation results show

that polynomial regression based extrapolation is more accurate than exponential moving average. In both the cases, the mean estimation accuracy of more than 90% is achieved for all the traffic parameters.

It was observed that edge level speed estimation using cellular network data alone is highly erroneous and there is need of additional accurate data sources. To enable the edge level speed estimation in whole road network, two models of ITS infrastructure deployment are proposed: COngestion COverage MOdel (COCOMO) and Edge COverage MOdel (ECOMO). Both the models exploit the availability of traffic information (congestion profile of edges) from the cellular network to minimize the ITS infrastructure requirement.

The COCOMO deploys ITS infrastructure on a set of edges to cover all six congestion levels that can occur in a road network. The infrastructure requirement of the model is constant and is independent of the number of edges in a road network (to 3-cover all six congestion levels, the ITS infrastructure deployment on atmost 18 edges is required). To the best of our knowledge, COCOMO is the only model with such unique characteristic.

The ECOMO forms clusters of similar edges using their congestion profile and deploys ITS infrastructure on few representative edges in each cluster. The model permits coverage of all the edges with infrastructure on less than 30% edges. We consider this as a major finding as most of the models proposed in the literature require infrastructure deployment on 60-80% edges in a road network.

It is assumed that accurate speed estimation is available on all the edges with ITS infrastructure. To enable speed estimation on all the edges, the occupancy-speed relationship is learned on infrastructure edges using polynomial regression model. The regression model and historical data of GPS probes are used for speed estimation on infrastructureless edges. The simulation results show that speed estimation with ninety percentile error of 10-22% and 10-13% with COCOMO and ECOMO, respectively, is achievable. The comparison of ITS infrastructure requirement and speed estimation accuracy in COCOMO and ECOMO show that, the higher infrastructure requirement of ECOMO is compensated by better accuracy in speed estimation.

The models are also evaluated for fault tolerance capability and feasibility of incremental ITS infrastructure deployment. Simulation results show that the models permit speed estimation for an edge or a congestion level when at least one covering infrastructure

edge is available.

The thesis presents a unique way of fusing traffic data from cellular network, GPS probes, and ITS infrastructure for accurate edge level speed estimation in real time.

To examine feasibility of large scale deployment, a MapReduce based distributed computing framework is designed for the proposed ITS. The detailed analysis of computation, communication, and storage requirements show the feasibility of large scale deployment of the proposed ITS using currently available technology.

An Advanced Traveler Information System (ATIS) is developed to assess the utility of generated traffic information. The ATIS uses real time traffic information generated by the proposed models to enable trip planning and suggest route changes during a trip based on traffic dynamics. The simulation results show that the ATIS improves average trip duration and congestion in road network significantly. The improvements are comparable to the scenario when accurate traffic information is available for all the edges (all the edges are equipped with ITS infrastructure).

The thesis elaborates design of a multi-modal Intelligent Transportation System (ITS) using cellular network, GPS probes and limited ITS infrastructure. The proposed infrastructure deployment models (COCOMO and ECOMO) exploit the availability of cellular network data and GPS probe data to reduce infrastructure requirement. The models permit incremental infrastructure deployment and try to maximize coverage using available infrastructure. The system does not make any assumption about traffic sensors and can support variety of them. These unique characteristics make the proposed ITS suitable for deployment in developing countries like India.

7.2 Future Work

A set of realistic simulations is used to evaluate performance of the proposed Intelligent Transportation System (ITS). As a future work and a step towards large scale deployment, a prototype system can be built and evaluated.

It is assumed that all the vehicles are equipped with a cell phone and are tracked using cellular infrastructure. As a future work, we aim to evaluate the signaling overhead incurred in cellular network for tracking all the vehicles. The effect of tracking only a fraction of the vehicles (to reduce signaling overhead) on accuracy of edge level traffic

estimation needs to be analyzed. Detecting presence of traffic lights at junctions by processing historical data of GPS probes can be attempted.

In the proposed ITS, deployment of static (non-moving) ITS infrastructure is assumed. Recently the interest has been developed in using mobile sensors and certain combinations of static and mobile sensors for traffic monitoring. The proposed infrastructure deployment models can be extended to consider these new developments.

A data set containing cellular fingerprints along with corresponding GPS location is not available for city wide or larger region in India or rest of the world. The availability of such standard data set may enable evaluation of various localization algorithms on common ground and open up doors for research in location based services (including real time traffic estimation) using cellular infrastructure. Development of the data set is part of future research.

In the current work, all the simulations are done using Simulator for Urban Mobility (SUMO), which enforces lane disciplined traffic movement. Eventhough we could get a good approximation of lane sharing heterogeneous traffic for the thesis work, there is a need of traffic simulator that supports lane sharing traffic movement naturally. As an effort in this direction, the SiMTraM (Simulation of Mixed Traffic Mobility) [68] was developed by the Transportation Research Group at IIT Bombay by adapting the SUMO version 0.12. The simulator divides lane width into multiple strips and assigns road space to vehicles in terms of the number of strips, thereby permitting multiple small vehicles to share the lane width. We upgraded the SiMTraM [69] for the SUMO version 0.17, but due to certain unresolved bugs it could not be used in the thesis work. We believe that the SiMTraM has potential to permit large scale simulations of lane sharing heterogeneous traffic. We aim to update and publish a better version of SiMTraM as a part of future work.

Chapter 8

List of Publications

Publications

1. M. Chaturvedi and S. Srivastava, "Real time vehicular traffic estimation using cellular infrastructure," *IEEE Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, December 2013
   This publication is based on the work discussed in section 3.2

2. V. Patel, M. Chaturvedi, and S. Srivastava, "Comparison of SUMO and SiMTraM for Indian traffic scenario representation," *11th International Conference on Transportation Planning and Implementation Methodologies for Developing Countries (TPMDC)*, December 2014
   This work was done while exploring simulators for representing Indian traffic scenario.

3. M. Chaturvedi and S. Srivastava, "Edge level vehicular traffic estimation using cellular infrastructure and other sources," *First workshop on Intelligent Transportation Systems as part of the 7th International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pp. 1–6, January 2015
   This publication is based on the work discussed in section 3.5.5.

4. M. Chaturvedi and S. Srivastava, "Edge-level real-time traffic estimation with limited infrastructure," *IEEE 82nd Vehicular Technology Conference (VTC)*, pp. 1–5, September 2015
   This publication is based on the work discussed in section 3.5.1,3.5.3.

In Communication

1. M. Chaturvedi and S. Srivastava, "Multi-modal design of an Intelligent Transportation System," IEEE Transaction on Intelligent Transportation Systems

Appendix 1

Cellular Fingerprint Sample Data

```
<root>
<celldata>
<datetime>26:March:2015:09:10:05</datetime>
<MCC>404</MCC>
<MNC>24</MNC>
<maincellid>1712</maincellid>
<maincellrssi>-61</maincellrssi>
<neighbours>
<item cid="7222" lac="15000" rssi="-93" />
<item cid="7772" lac="15000" rssi="-95" />
<item cid="532" lac="15000" rssi="-97" />
</neighbours>
<latitude>23.1284138</latitude>
<longitude>72.5422372</longitude>
</celldata>
<celldata>

  ...

</celldata>

 ⋮

</root>
```

Description

- The <celldata> tag contains one record of cellular fingerprint and corresponding GPS coordinates.
- The <datetime> tag specifies time stamp when the data is recorded.

- The MCC (Mobile Country Code) is 404 for India.
- The MNC (Mobile Network Code) is 24 for the IDEA network.
- The <maincellid> tag stores the identifier of the GSM cell to which the mobile terminal is registered (the main cell ID).
- The <maincellrssi> tag stores the measure of received signal strength (RSSI) value in dbm for the main cell ID.
- The <neighbours> tag contains information about the cell ID and corresponding RSSI value for all the neighbor cells visible at the location. The "lac" attribute specifies the Location Area Code of the location.
- The <latitude> and <longitude> tags specify the GPS coordinates at the location.

Bibliography

[1] "India transport report (volume 1)- moving India to 2032," *National Transport Development Policy Committee and Planning Commission, Government of India*, 2014.

[2] "Road transport year book 2009-10 and 2010-11," *Transportation Research Wing, Ministry of Road Transport and Highways, Government of India*, July 2012.

[3] "India transport report (volume 2)- trends in growth and development of transport," *National Transport Development Policy Committee and Planning Commission, Government of India*, 2014.

[4] D. Schrank, T. Lomax, and B. Eisele, "Urban mobility report," *Texas Transportation Institute*, September 2011.

[5] *Code of Federal Regulations(CFR), Chapter 23, Section 940*, 2001.

[6] J. S. Sussman, "Perspectives on Intelligent Transportation Systems (ITS)," *Springer*, 2005.

[7] "Intelligent transportation systems benefits, costs, and lessons learned," *Publication Number: FHWA-JPO-14-159, U.S. Department of Transportation*, June 2014.

[8] "Press release no. 65/2015," *Telecom Regulatory Authority of India*, September 2015.

[9] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical metropolitan-scale positioning for GSM phones," *8th ACM International Conference on Ubiquitous Computing (UbiComp)*, pp. 225–242, 2006.

[10] E. Trevisani and A. Vitaletti, "Cell-ID location technique, limits and benefits: an experimental study," *6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 51–60, December 2004.

[11] K. Raja, W. Buchanan, and J. Munoz, "We know where you are (cellular location tracking)," *IEE Communications Engineer*, vol. 2, issue 3, June-July 2004.

[12] A. Varshavsky, M. Y. Chen, E. de Lara, J. Froehlich, D. Haehnel, J. Hightower, A. LaMarca, F. Potter, T. Sohn, K. Tang, and I. Smith, "Are GSM phones the solution for localization?" *Proceedings of the 7th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 34–42, August 2005.

[13] H. Bar-Gera, "Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel," *Transportation Research Part C, Elsevier*, vol. 15, pp. 380–391, 2007.

[14] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cellular phones: a case-study in Rome," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 141–151, 2011.

[15] R. Cayford and T. Johnson, "Operational parameters affecting the use of anonymous cell phone tracking for generating traffic information," *Transportation Research Board, 82nd Annual Meeting*, January 2003.

[16] J. C. Herrera, D. B. Work, R. Herring, X. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research Part C, Elsevier*, vol. 18, pp. 568–583, 2010.

[17] J. Aslam, S. Lim, and X. Pan, "City-scale traffic estimation from a roving sensor network," *10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 141–154, November 2012.

[18] "Greater Ahmedabad integrated mobility plan - situation analysis interim report, Vol. 1," *Prepared by Centre of Environment and Planning Technology (CEPT) and Urban Mass Transit Company Limited (UMTC)*, 2012.

[19] "Greater Ahmedabad integrated mobility plan - data collection and analysis," *Prepared by Centre of Environment and Planning Technology (CEPT), Urban Mass Transit Company Limited (UMTC), and Lea Associates South Asia Pvt Ltd (LASA)*, 2012.

[20] T. Park and S. Lee, "A bayesian approach for estimating link travel time on urban arterial road network," *International Conference on Computational Science and Its Applications (ICCSA)*, pp. 1017–1025, 2004.

[21] C. H. Wei and Y. Lee, "Development of freeway travel time forecasting models by integrating different sources of traffic data," *IEEE Transactions on vehicular technology*, vol. 56, pp. 3682–3694, November 2007.

[22] R. A. Anand, L. Vanajakshi, and S. C. Subramanian, "Traffic density estimation under heterogeneous traffic conditions using data fusion," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 31–36, June 2011.

[23] D. Valerio, A. D'Alconzo, F. Ricciato, and W. Wiedermann, "Exploiting cellular networks for road traffic estimation: a survey and a research road map," *69th IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, April 2009.

[24] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *6th ACM USENIX Symposium on Operating System Design and Implementation*, pp. 137–149, 2004.

[25] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," *6th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 323–336, November 2008.

[26] D. Valerio, T. Witek, F. Ricciato, R. Pilz, and W. Wiedermann, "Road traffic estimation from cellular network monitoring: a hands-on investigation," *20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 225–242, 2009.

[27] C. Birle, "Traffic online via GSM network data to traffic information in real time," *Presentation Vodafone*, 2007.

[28] R.-H. Liou, Y.-B. Lin, Y.-L. Chang, H.-N. Hung, N.-F. Peng, and M.-F. Chang, "Deriving the vehicle speeds from a mobile telecommunications network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1208–1217, September 2013.

[29] M. G. Demissie, G. H. de Almeida Correia, and C. Bento, "Intelligent road traffic status detection system through cellular networks handover information: An exploratory study," *Transportation Research Part C, Elsevier*, vol. 32, pp. 76–88, 2013.

[30] N. Caceres, J. Wideberg, and F. Benitez, "Deriving origin-destination data from a mobile phone network," *IET Journal of Intelligent Transport System*, vol. 1, issue 1, pp. 15–26, 2007.

[31] B. L. Smith, M. D. Fontaine, J. Beaton, E. Dejarnette, A. Hendricks, and L. L. Tennant, "Private-sector provision of congestion data," *Final Report NCHRP 70-01, Center for Transportation Studies, University of Virginia*, February 2007.

[32] Q. Zhao, Q.-J. Kong, Y. Xia, and Y. Liu, "Sample size analysis of GPS probe vehicles for urban traffic state estimation," *14th IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 272–276, October 2011.

[33] R. K. Balan, N. X. Khoa, and L. Jiang, "Real-time trip information service for a large taxi fleet," *9th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 99–112, 2011.

[34] L. Zhang, S. D. Gupta, J.-Q. Li, K. Zhou, and W. bin Zhang, "Path2go: Context-aware services for mobile real-time multimodal traveler information," *14th IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 174–179, October 2011.

[35] Q. T. Minh, M. A. Baharudin, and E. Kamioka, "Context-aware mobile intelligent transportation systems," *76th IEEE Vehicular Technology Conference (VTC)*, pp. 1–6, September, 2012.

[36] P. Nitsche, P. Widhalm, S. Breuss, N. Brandle, and P. Maurer, "Supporting large-scale travel surveys with smartphones - a practical approach," *Transportation Research Part C, Elsevier*, vol. 43, pp. 212–221, June 2014.

[37] R. Sen, A. Maurya, B. Raman, R. Mehta, R. Kalyanaraman, N. Vankadhara, S. Roy, and P. Sharma, "Kyunqueue: A sensor network system to monitor road traffic queues," *10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 127–140, November 2012.

[38] N. Kassem, A. E. Kosba, and M. Youssef, "RF-based vehicle detection and speed estimation," *75th IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, May 2012.

[39] A. Al-Husseiny and M. Youssef, "RF-based traffic detection and identification," *76th IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, September 2012.

[40] R. Sen, B. Raman, and P. Sharma, "Horn-ok-please," *8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 137–150, June 2010.

[41] R. Sen, P. Siriah, and B. Raman, "Roadsoundsense: Acoustic sensing based road congestion monitoring in developing regions," *8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 125–133, June 2011.

[42] S. S. M. Ali, B. George, and L. Vanajakshi, "An efficient multiple-loop sensor configuration applicable for undisciplined traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1151–1161, September 2013.

[43] S. Taghvaeeyan and R. Rajamani, "Portable roadside sensors for vehicle counting, classification, and speed measurement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 73–83, February 2014.

[44] E. Levenberg, "Estimating vehicle speed with embedded inertial sensors," *Transportation Research Part C, Elsevier*, vol. 46, pp. 300–308, 2014.

[45] M. Stocker, M. Ronkko, and M. Kolehmainen, "Situational knowledge representation for traffic observed by a pavement vibration sensor network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1441–1450, August 2014.

[46] J. Zoto, R. J. La, M. Hamedi, and A. Haghani, "Estimation of average vehicle speeds traveling on heterogeneous lanes using Bluetooth sensors," *76th IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, September 2012.

[47] A. Thiagarajan, L. Ravindranath, K. Lacurts, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 85–98, November 2009.

[48] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," *8th ACM USENIX conference on Networked Systems Design and Implementation (NSDI)*, pp. 267–280, March-April 2011.

[49] A. Bhaskar, T. Tsubota, L. M. Kieu, and E. Chung, "Urban traffic state estimation: Fusing point and zone based data," *Transportation Research Part C, Elsevier*, vol. 48, pp. 120–142, November 2014.

[50] C. Bachmann, B. Abdulhai, M. J. Roorda, and B. Moshiri, "A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling," *Transportation Research Part C, Elsevier*, vol. 26, pp. 33–48, January 2013.

[51] S. R. Hu and H. T. Liou, "A generalized sensor location model for the estimation of network origin-destination matrices," *Transportation Research Part C, Elsevier*, vol. 40, pp. 93–110, 2014.

[52] L. Bianco, G. Confessore, and M. Gentili, "Combinatorial aspects of the sensor location problem," *Annals of Operations Research, Springer*, vol. 144, issue 1, pp. 201–234, 2006.

[53] M. Gentili and P. Mirchandani, "Locating sensors on traffic networks: Models, challenges and research opportunities," *Transportation Research Part C, Elsevier*, vol. 24, pp. 227–255, 2012.

[54] X. Fei and H. S. Mahmassani, "Structural analysis of near-optimal sensor locations for a stochastic large-scale network," *Transportation Research Part C, Elsevier*, vol. 19, pp. 440–453, 2011.

[55] X. Fei, H. S. Mahmassani, and P. Murray-Tuite, "Vehicular network sensor placement optimization under uncertainty," *Transportation Research Part C, Elsevier*, vol. 29, pp. 14–31, 2013.

[56] S. R. Hu, S. Peeta, and C. H. Chu, "Identification of vehicle sensor locations for link-based network traffic applications," *Transportation Research Part B, Elsevier*, vol. 43, pp. 873–894, 2009.

[57] H. Park and A. Haghani, "Optimal number and location of Bluetooth sensors considering stochastic travel time prediction," *Transportation Research Part C, Elsevier*, vol. 55, pp. 203–216, 2015.

[58] N. Zhu, Y. Liu, S. Ma, and Z. He, "Mobile traffic sensor routing in dynamic transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2273–2285, October 2014.

[59] M. Chaturvedi and S. Srivastava, "Data set of 10 days collected over an 18 km road stretch in Ahmedabad city, India," *available at https://goo.gl/Htf7R6*, March 2015.

[60] J. S. Armstrong, "Long-range forecasting: From crystal ball to computer," *Wiley-Interscience Publication*, 1985.

[61] T. Mitchell, "Machine learning," *McGraw Hill*, 1997.

[62] R. Bertini, "You are the traffic jam: An examination of congestion measures," *Transportation Research Board, 85th Annual Meeting*, pp. 1–17, 2006.

[63] C. Patel and G. Joshi, "Capacity and LOS for urban arterial road in Indian mixed traffic condition," *Procedia - Social and Behavioral Sciences: (Transportation Research Arena - Europe), Elsevier*, vol. 48, pp. 527–534, 2012.

[64] V. Vazirani, "Approximation algorithms," *Springer*, 2001.

[65] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, issue 4, 2007.

[66] S. Wood, "Traffic microsimulation - dispelling the myths," *Traffic Engineering and Control (TEC) magazine*, vol. 53, issue 9, pp. 339–344, October, 2012.

[67] G. Tiwari, "Traffic flow and safety: Need for new models for heterogeneous traffic," *5th World Injury Conference, Delhi*, 2000.

[68] T. V. Mathew and A. Bajpai, "SiMTraM user documentation," *Available on web at http://www.civil.iitb.ac.in/simtram*, last accessed on December 07, 2015.

[69] V. Patel, M. Chaturvedi, and S. Srivastava, "Upgraded SiMTraM simulator," *Available on web at https://sites.google.com/site/simtramsimulator/*, last accessed on December 07, 2015.

[70] V. Patel, M. Chaturvedi, and S. Srivastava, "Comparison of SUMO and SiMTraM for Indian traffic scenario representation," *11th International Conference on Transportation Planning and Implementation Methodologies for Developing Countries (TP-MDC)*, December 2014.

[71] B. Maitra, "Lecture 3: Traffic stream characteristics," *NPTEL on Introduction to Transportation Engineering, IIT Kharagpur*, 2008.

[72] S. Krauss, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," *PhD Thesis report*, April 1998.

[73] J. J. Olstam and A. Tapani, "Comparison of car-following models," *Technical report VTI meddelande 960A, Swedish National Road and Transport Research Institute*, 2004.

[74] K. Varmora and P. J. Gundaliya, "Effect of traffic composition and road width on urban traffic stream," *Paripex- Indian Journal of Research*, vol. 2, issue 4, April 2013.

[75] D. Levinson, "The value of advanced traveler information systems for route choice," *Transportation Research Part C, Elsevier*, vol. 11, pp. 75–87, 2003.

[76] R. Bertini and A. El-Geneidy, "Chapter 15: Advanced traffic management system data," *Assessing the Benefits and Costs of ITS, Springer*, pp. 287–314, 2004.

[77] L. Xiao and H. K. Lo, "Adaptive vehicle navigation with en route stochastic traffic information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1900–1912, 2014.

[78] P. Kumar, V. Singh, and D. Reddy, "Advanced traveler information system for Hyderabad city," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 26–37, March 2005.

[79] E. Ben-Elia, R. D. Pace, G. N. Bifulco, and Y. Shiftan, "The impact of travel information's accuracy on route-choice," *Transportation Research Part C, Elsevier*, vol. 26, pp. 146–159, 2013.

[80] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 3-4, pp. 128–138, 2012.

[81] S. Chintu, R. Anthony, M. Roshanaei, and C. Ierotheou, "Intelligent and predictive vehicular networks," *Intelligent Control and Automation, Scientific Research*, vol. 5, pp. 60–71, 2014.

[82] M. Chaturvedi and S. Srivastava, "Real time vehicular traffic estimation using cellular infrastructure," *IEEE Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, December 2013.

[83] M. Chaturvedi and S. Srivastava, "Edge level vehicular traffic estimation using cellular infrastructure and other sources," *First workshop on Intelligent Transportation Systems as part of the 7th International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pp. 1–6, January 2015.

[84] M. Chaturvedi and S. Srivastava, "Edge-level real-time traffic estimation with limited infrastructure," *IEEE 82nd Vehicular Technology Conference (VTC)*, pp. 1–5, September 2015.