RELIABLE BANDWIDTH CONSERVATIVE QUEUE-END DETECTION AND WARNING SYSTEM USING SMART PHONE COLLABORATION TECHNIQUES

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science

In

Electronic System Engineering

University of Regina

By

Shahedur Rahman

Regina, Saskatchewan

July, 2012

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Shahedur Rahman, candidate for the degree of Master of Applied Science in Electronic Systems Engineering**,** has presented a thesis titled, ***Reliable Bandwidth Conservative Queue-End Detection and Warning System Using Smart Phone Collaboration Techniques,*** in an oral examination held on July 16, 2012. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: `	Dr. Samira Sadoui-Mouhoub, Department of Computer Science
Supervisor:	Dr. Mohamed El-Darieby, Electronic Systems Engineering
Committee Member:	Dr. Craig M. Gelowitz, Software Systems Engineering
Committee Member:	Dr. Raman Paranjape, Electronic Systems Engineering
Chair of Defense:	Dr. Mohamed Ismail, Industrial Systems Engineering

*Not present at defense

ABSTRACT

The goal of this thesis is to ensure public safety. Two of the major components of public safety are: threat detection and notification. This thesis provides threat detection through sensor networks, collaboration of devices, and notification through three types of warning messages using cell broadcast technology.

Most emergency conditions, be it natural or manmade, either directly or indirectly create congestion both on the road and in the telecommunication network. The more severe the road congestion is, the more vulnerable the tail end of the congestion is to rear end collisions. In order to avoid rear end collisions, it is important to know the end location of the traffic jam (i.e. the end of the queue) hence the name, queue-end (q-end) problem. If the exact q-end location is detected and made known to the drivers in a timely manner, many more accidents and incidents can be avoided. The q-end problem is resolved in three major steps: congestion detection, exact q-end location determination, and warning. Dynamic nature of the q-end problem makes it an iterative process. The design objective requires the traffic monitoring and warning system to be reliable, available, and use conservative resources.

A middleware has been designed that will allow important messages to reach affected areas during network congestion. Cell broadcast is a one-to-many communication method, also known as Short Message Service-Cell Broadcast (SMS-CB). SMS-CB uses non traffic barring network resources and is unaffected by network congestion and most partial failures. This system also takes advantage of other wireless communication methods (i.e. Zigbee, Wifi, Bluetooth etc), internal stand alone resources, and collaboration capabilities built into smart phones.

The proposed solution is designed as a multi-tiered cloud architecture system. The solution ensures reduced network traffic, multi-level backup mechanisms, as well as localized computing and control. The middleware supports both centralized and standalone operations, minimizing the impact of catastrophic network failure. The queue end or danger detection will be performed through a collaboration of sensors or user devices, which will send information to designated location for validation. In most situations, the validated information will be sent to the users via cell broadcast technology.

Simulations and actual road tests have been performed to prove that smart phones are capable of detecting and sharing congestion with other smart devices and that they have the ability to receive and process warning messages to inform the end user of potential danger (i.e. location of queue-end). The middleware solution is suitable for day-to-day traffic congestion, accidents (i.e. q-end problem) when telecommunication systems are fully functional, as well as during any other major disasters (i.e. terrorist attack, major natural catastrophe, industrial accidents or other major issues) when telecommunication systems may be congested or malfunctioning.

ACKNOWLEDGEMENT

Special thanks to my academic supervisor

Dr. Mohamed El-Darieby

I would also like to extend my gratitude to

SaskTel (http://www.sasktel.com/)

for their support throughout my research and graduate studies

In addition, I would like to thank Dr. Amr Henni, Dr. Amir Ashique, my co-workers, and Bashar Rashid who have provided lots of help during my research and/or writing of my thesis.

Also special thanks to my wife Shiela, my nieces Maysa and Inaya for helping me during drive tests and proof reading my Thesis.

DEDICATION

To my beloved wife Shiela and my parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ANN	Artificial Neural Network
AP	Access Point
API	Application Programming Interface
AS	Access Stratum
BSC	Base Station Controller
BTS	Base Transceiver Station
CAM	Cooperative Awareness Message
CAP	Common Alert Protocol
CB	Cell Broadcast
CBC	Cell Broadcast Centre
CBE	Cell Broadcast Entity
CBS	Cell Broadcast Service
CDMA	Code Division Multiple Access
CMAS	Commercial Mobile Alert System
CS	Circuit Switch
GGSN	Gateway GPRS Support Node
GMLC	Gateway Mobile Location Centre
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
EDGE	Enhanced Data rates for GSM Evolution
E-UTRAN	Evolved UTRAN
HLR	Home Location Register
LA	Location Area
LTE	Long Term Evolution
IMSI	International Mobile Subscriber Identity
MAP	Mobile Application Part (protocol)
MESA	Mobility for Emergency and Safety Applications
MGW	Media Gateway
MCC	Mobile Country Code
MME	Mobility Management Entity
MNC	Mobile Network Code
MSC	Mobile Switching Center
MSS	MSC Server
NAS	Non-Access Stratum
NMEA	National Marine Electronics Association
OBD	On-Board Diagnostic
P-GW	Packet Data Network Gateway

PC	Personal Computer
PLMN	Public Land Mobile Network
PS	Packet Switch
PSAP	Public Safety Answering Point
PSDR	Public Safety and Disaster Relief
PSTN	Public Switched Telephone Network
RA	Routing Area
RAN	Radio Access Network
RF	Radio Frequency
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RSSI	Received Signal Strength Indicator
S-GW	Serving Gateway
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SMLC	Serving Mobile Location Centre
SMS	Short Message Service
SMSC	Short Message Service Center
SMS-CB	Short Message Service – Cell Broadcast
SAE	System Architecture Evolution
SAE-GW	SAE gateway
TA	Tracking Area
TDMA	Time Division Multiple Access
UE	User Equipment
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-Hoc Network
VLR	Visitor Location Register
VoIP	Voice over IP
WARN	W
WCDMA	Wideband Code Division Multiple Access
WMN	Wireless Mesh Networks
XML	eXtensible markup language

CHAPTER 1 INTRODUCTION

Public Safety and Disaster Relief (PSDR) is an important part of everyone's life as the purpose of public safety is to prevent and protect any creature or objects from being injured, harmed or damaged by natural or manmade causes [1]. Public safety mandates codes, standards, rules and regulations, and employs different agencies for their enforcement. This ensures that houses and buildings are safe to dwell in, that roads are safe to ride on, and that bikes, vehicles and airplanes are reliable for travel. For example, traffic rules reduces accidents, building codes prevents injuries (i.e. railings in stair cases, electric shocks etc), a tsunami warning directs people to safe location protecting their lives.

PSDR consists of multiple agencies and groups, including, but not limited to law enforcement agencies (local, provincial, and national police forces), search and rescue agencies (coastal, local volunteer), transportation safety (highway patrol, highway maintenance, sanding and salting crew, aviation), health and long term care (land and air ambulance), and natural resources (park, fire and flood management) just to name a few. The scope of these agencies is not limited to just local municipalities or provincial regions, as they can also have national or even international breadth.

It is imperative that each of these agencies is able to communicate with their own members privately as well as other agencies. Privacy is also needed for reliability, security, and integrity. In many cases different agencies run their own systems within the same region, which can result in ineffective communication in times of disaster when one agency needs help of another agency. In some cases several agencies share the system within the same region, while maintaining their privacy. However, use of private

networks creates difficulties in cooperation when they need help from other national or international agencies.

For these reasons, the need for interoperability and the complexity of the types of telecommunication required are increasing. The communication requirements of Canadian PSDR can be found in [2]. The telecommunication type is no longer limited to voice only service, but has expanded to include data and video. Improvement in technology allows remote operations enabling experts from all over the world to contribute in a crisis without adding any load to local scarce resources or endangering other lives. A project called MESA (Mobility for Emergency and Safety Applications) has been initiated to address worldwide interoperability of emergency response teams. [3].

Although public safety measurements are in place, in terms of traffic signals, warning signs, and codes and standards, many factors remain beyond the control of human kind (i.e. natural disaster), some of which are preventable while others are not. For many crises, the only way to save peoples' live is to have them evacuate the dangerous location (i.e. disruption of harmful gas or volcano, earthquake, tsunami, etc). The faster someone can leave, the better the chance of their survival. However, faster travel also increases the chance for collision. In some situations, people need warnings to avoid certain locations (i.e. a slippery road, a road prone to immediate avalanche) thus ensuring their safety. In other cases, all that the motorists need is to be advised to slow down immediately. For example, a timely warning of an accident on a highway or freeway could prevent rear end collisions whether or not it is visible. Snow ploughs are another

example of how people could benefit from early warning, since snowy conditions and the plough itself can cause confusion and disorientation for the driver.

As noted above both telecommunication and efficient road transportation are necessary in ensuring public safety. With timely communication, it is possible to eliminate many traffic problems. One of the biggest road-hazard is the queue end (q-end) problem, which is one of the root causes of rear end collisions. "The rear-end collision is the primary type of multi-vehicle incident, comprising over 50 percent of crashes" [4]. According to [5], rear end collisions account for one-sixth of all automobile claims and 38% of the money paid for all automobile claims in the United States of America. These q-end problems can be minimized or eliminated by using timely detection and warning.

Q-end problem originate from congestion or road obstructions. It is the end location, the tail end of congestion, or any lineup of vehicles. The more severe the congestion is, the worse the q-end problem is. The worst case of a q-end problem is when traffic is at a standstill in the middle of highway or freeway where traffic is expected to flow at a high speed. It is further aggravated during poor road and weather conditions when visibility is low. It is important to know the exact location of the last stopped vehicle in order to avoid further accidents.

The occurrence of Q-end problems are not limited to toll booths or border crossings, but also to construction zones, traffic accidents, snow ploughing, street cleaning and even rush hour traffic. In most cases, q-end problems and congestion problems go hand-in-hand, often inflating each other. The key to solving q-end problem are: **detecting congestion, locating the queue end (i.e. tail end of congestion), then notifying the**

users. If drivers are pre-warned about q-end location and road conditions, they can adjust their driving speed to avoid rear end collisions. They can also choose to take an alternate route to avoid congested or troubled areas.

1.1 Problem Definition and Identification

In most instances, predicting q-end problems is complex, difficult and dynamic in nature. Currently, warning systems regarding downstream queuing given to drivers are minimal or non-existent [4]. A proper q-end solution will not only be useful for day-to-day traffic problems, but also for emergency situations. Major catastrophes such as, terrorist attacks, natural disasters (earthquake, tsunami, storms, flood, tornado, fire), or other major technological failures tend to cause congestion and traffic problems. Congestion leads to q-end problems which in turn cause more congestion resulting in accidents, anxiety, and road rages. It stands to reason that, if information regarding end location of congestion could be located and used to pre-warn drivers, road accidents and stress related health problems could be reduced, saving valuable resources like time, fuel, and lives.

Currently, there are services that supply drivers with live traffic information and recommends alternative routes to avoid further congestion. These services are available over the internet and accessible through smart phones, navigation systems and vehicles' on board computers. These existing systems do not nullify the purpose of this thesis, because there are still many gaps in these services that should be filled. Existing services focus mainly on congestion detection and notification, and do not provide q-end location. Most of these solutions also rely heavily on the cellular network or additional equipment that needs to be installed. For these reasons, if there is a network problem, congestion notification can also fail.

The main objective of this thesis is to design a secure, robust, reliable and universal public safety and warning system that can automatically identify different threats, and is sustainable to network congestion or outages. Of course, it must also be able to reach all affected people regardless of disabilities (i.e. deafness, blindness, mobility), language barriers, or affordability. To address this issue, [6] recommends that a warning system should have audio, visual, and vibration capabilities. Additionally, the message should also be standard as defined in [7] and universal (i.e. support different languages).

The availability, popularity, and intelligence of mobile phones and PDAs are increasing everywhere in the world. With the growth of mobile devices, cellular coverage areas are also increasing. Newer vehicles are also getting smarter and being equipped with computers (also known as Carputers), multiple communication systems and sensors. Some communication methods for vehicles include IEEE 802.11p (vehicle-to-vehicle/v2v) Wi-Fi, Bluetooth, ZigBee, and cellular modules, which give them the ability to connect with the internet or different cellular networks, to send or receive Short Message Service (SMS) messages (also known as Text Messages), and to communicate with other smart equipment within the vehicle. Many vehicles include cameras, location (GPS) and direction (compass) sensors, road traction sensors, climate (rain, snow, temperature, and light) sensors, as well as speed, roll and sudden stop sensors. Vehicle-to-vehicle (v2v) communication is also becoming a reality. Due to the prevalence of mobile devices, turning them into public safety tools for both detection and warning makes economic and scientific sense. Mobile devices also meet the audio, visual, and vibration notification criteria mentioned in [6].

1.2 Proposed Solution for Q-end and Warning

The solution provided in this thesis is a combination of many existing technologies and ideas, many of which have been used in different forms by similar researchers and solution providers. This thesis provides a network dependent as well as an independent resolution for emergency warnings that is immune to total network outage. That is an uninterrupted collision avoidance service at the time when it is needed the most. In fact, it will also help to reduce the telecommunication network load, so that it is not overloaded at the time of catastrophe.

Ideally, there will be sensors located at every road in the world to monitor vehicle speed, size, direction and identification, as well as road and weather conditions. In addition, there will be smart cameras to identify stolen vehicles, capture plates of traffic violators and detect congestion. Although this is impractical and not the case in the real world, it is feasible to have permanent sensors deployed only in high traffic or high collision areas.

Not all vehicles have smart sensing and computing abilities. However, if different vehicles and smart devices can collaborate with each other, then every road in the world can be considered as being equipped with sensors. The proposed solution takes advantage of these resources by filling the gaps of fixed sensor deployment with these smart devices.

Here is a brief overview of how the system will operate. Like many other solutions this method also uses the cellular network, smart devices (i.e. smart phones, smart cars and sensors), and other unlicensed short range wireless systems. To ensure information is sent only to the relevant parties, the region is divided into *macro grids*. *Macro grids*

provide a sense of physical boundary and coincide with cell broadcast zones. That is it will divide the regions based on cell tower coverage as defined by a cell broadcast zone or area. However, the *macro grid area* is going to be approximated by the closest polygon covering the CB area. Each *macro grids* consists of zero, one or more *micro grids*. Each *macro grid* has a *macro grid manager* that is responsible for its operation. The *macro grid manager* could be a *micro grid* with elevated resources with a *trusted* profile or it could be the *central server cloud* if no such *micro grid* is present. The *central server cloud* is referred to as the "core" in this thesis.

Each *micro grid* can have either a single device (i.e. a sensor, a phone, a smart phone, a laptop or other devices capable of communicating wirelessly) or a combination of devices (i.e. a phone and a GPS, or a phone, a GPS and a computer, or many other combinations). *Micro grids* can be mobile or static. Sensors in the micro grids could be traffic sensors (speed, length, counter etc.), road sensors (slippery, icy, wet, snowy), weather sensors or other type of sensor. All the devices in a *micro grid* will collaborate with each other and act as a single unit in reference to the *macro grid*. On the other hand, all the micro grids in the macro grid collaborate with each other.

Each *micro grid* will either provide one or more services to the *macro grid* or just be a user of the information provided in the *macro grid*. These services will include, but are not limited to sensing speed, road and weather conditions, and be a gateway to different communication methods (i.e. WiFi, ZigBee, WiMAX etc.). For the most part there will be no intra *micro grid* communication, which will enhance security and privacy. A *micro grid* will only inform the *macro grid manager* when a *micro grid* detects a road hazard.

In normal network conditions, most of this upstream communication will take place via SMS or internet protocol using the cellular network. The downstream notification to the rest of the *micro grids* in the *macro grid* will take place using cell broadcast technology, also known as Short Message System – Cell Broadcast (SMS-CB). One instance when there will be inter *micro grid* communication is when the *macro grid* will operate in standalone mode. This will happen when there is cellular network failure or where there is permanent sensor deployment. It is assumed that inter *micro grid* communication will take place using ad-hoc mesh network using v2v, Wi-Fi, Bluetooth, or ZigBee technologies when the cellular network fails. However, detailed information on this subject is beyond the scope of this thesis. Another instance of inter *micro grid* communication is when a *trusted micro grid* enters or exits a *macro grid* providing traffic updates to the *macro grid manager*.

All the streets are broken into *hops*. Hops are sections of the street, which are primarily based on road intersections, speed limit changes or name changes. A *route* consists of one or more hops. Attributes of a hop is designed such that they will all fit in a SMS-CB message. This will ensure successful delivery of an evacuation or re-routing information.

A middleware has been designed to manage inter and intra *macro grid* communication, authentication, and service management. Each *macro grid* will communicate with the *core* which will be centrally managed. The *core* will be responsible for maintaining several repositories of local and distributed resources, provisioning user data, and providing machine to machine or machine to human interfaces. In addition, the *core* will be responsible for deciding which information should be shared with whom.

1.3 Major Thesis Contributions

This section highlights the major contributions of this thesis:

1. Novel scenarios of q-end detection using innovative hybrid of technologies, meaning where and when to use smart user equipments (UE) as sensors or probes verses installing fixed sensors. The attributes of fixed sensors and their placement on the road in a grid environment while optimizing number of sensors used without compromising public safety.
2. A middleware has been designed to address issues of detecting congestion and issuing warnings. The middleware is responsible for the *micro grid* to *macro grid* communication, functionality, and the handover of *micro grid* between different *macro grids*. It also provides authentication, profiles, and service repository management for the *macro grid*.
3. Multiple congestion detection algorithms have been developed that are suitable for mobile and non-mobile devices. The algorithm used in a *micro grid* will depend on the resources available. All of these algorithms will accompany sudden stop detection, either through deceleration rate calculation or retrieval of sensor reading (i.e. vehicles air bag deployment) in order to allow a very quick detection of the accident. These algorithms are listed below:

a) The first proposed algorithm will compare the speed limit with the actual speed. If the actual speed is below a decided threshold, then it will initiate and wait for 3 consecutive timers before it declares the road to be congested. Each timer is dynamically calculated based on the current hop's (road section) attribute. After all expiry of all three timers if the traveling speed of a vehicle is still below the threshold

the *micro grid* will send a message including the location and speed to the *macro grid manager*, which will first verify the profile of the sender. If the message comes from a legitimate *micro grid* the message will be broadcasted via SMS-CB to all the *micro grids* in that *macro grid*. The *micro grids* receiving this message will check their own speed. Any *micro grid* suffering from congestion and that is also behind a threshold distance (i.e. 50 meter) from the reported location will send updated congestion messages to the *macro grid manager*. The *macro grid manager* will then compare all the incoming messages and broadcast the furthest reported location. Each timer will be set dynamically based on the location. This algorithm is for a *macro grid* consisting mainly of mobile devices.

b) The second proposed algorithm is to compare the actual distance traveled with expected distance during a time interval instead of speed. It will also wait for dynamic timers similar to those mentioned above in order to account for traffic signals or other expected documented delays.

c) The third option is to compare calculated time in a hop and / or in a route with the actual time spent in a hop or in a route. If a vehicle has not traveled a threshold distance at a threshold time (say half the distance, at half the expected time) it will consider the location to be congested.

d) The fourth option is based on actual dwell time in a *macro grid* compared with the expected dwell time in the *macro grid*.

e) The fifth option is a combination one or two or all of the above.

f) The sixth option is based on the sensor capabilities. This will apply more towards fixed sensor deployments. For example, two sensors (one at entry point and one at

exit point) will count, classify (i.e. based on size), and read the vehicle speed in order to determine length, location, and speed at the end of the queue, while collaborating with each other and with other mobile *micro grids*.

4. In order to meet compatibility requirement with national and international emergency agencies and to maintain integrity, a message format has been developed. The format follows the criteria laid out by OASIS Common Alert Protocol (CAP) version 1.2 and still meets cell broadcast system (CBS) message requirement, which is based on 3GPP standard.
5. An application has been developed using Blackberry Java Development Kit to partially implement the middleware and to prove its functionality. The developed software was tested in both simulated and in actual UMTS based network on real Blackberry phones. The first algorithm mentioned above was implemented. The phones (i.e. *micro grids*) were able to detect congestion and relay the messages to other *micro grids*. The *micro grids* collaborated with each other to find the final q-end location.
6. Network load analysis using real life data from a live UMTS network as well as some calculations on network resource usage. This includes factors such as how bandwidth will be saved using SMS-CB method verses a device retrieving live traffic information using some other services/methods.

1.4 Dissertation Overview

The second chapter of this thesis provides a literature review of different methods that can be used to find a point on earth and to detect congestion, as well as existing traffic monitoring services, existing work on q-end detection, and existing use of cell-broadcast

technology. The third chapter provides the architecture and design of the overall solutions and methodologies. The fourth chapter provides detailed information on the software and middleware design, different failover and operational scenarios, and congestion detection algorithms. The fifth chapter includes implementation experience, network traffic volume analysis and how its efficiency compares with other solutions. Finally, the sixth chapter concludes the thesis by summarizing the research and recommending future endeavours.

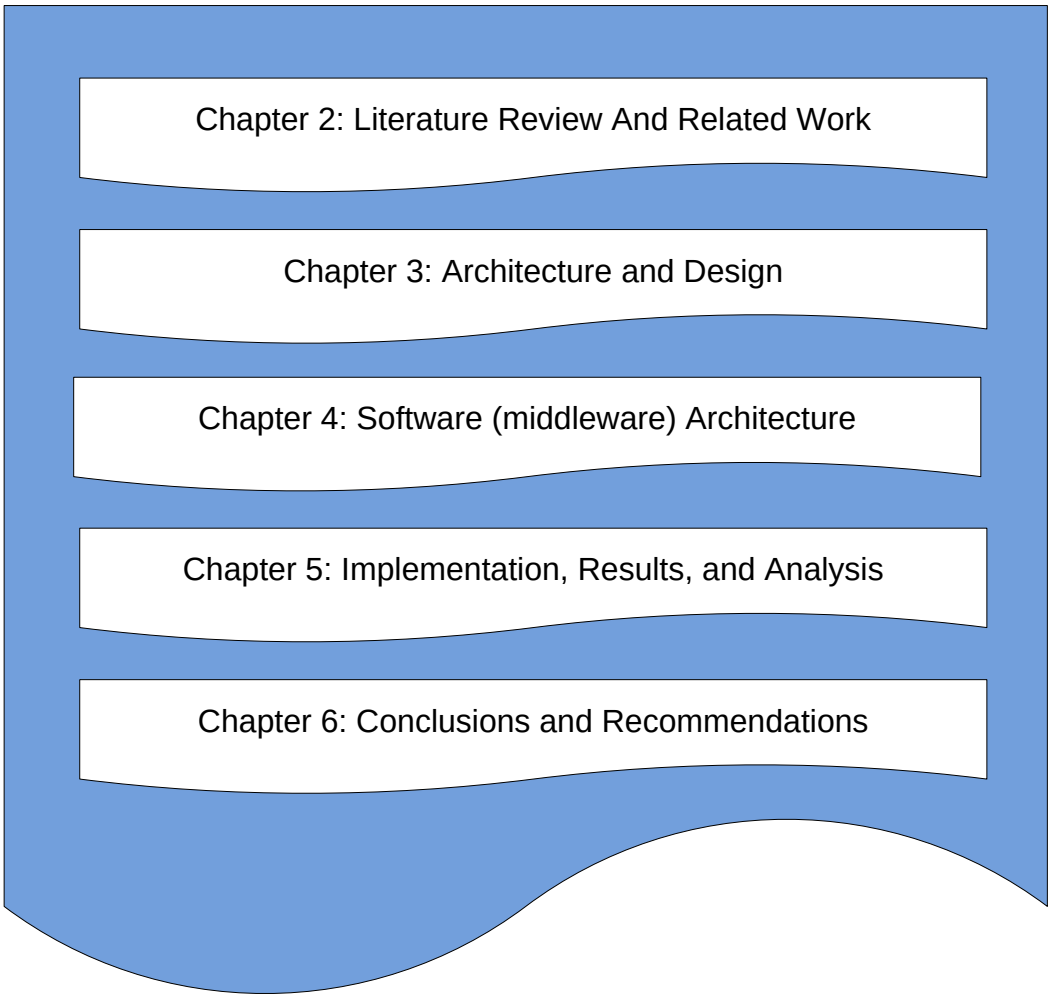

Figure 1.1 Thesis Outline

CHAPTER 2 LITERATURE REVIEW AND RELATED WORK

This chapter consists of five major areas of research. The first part discusses different methods for determining the location of a vehicle. The second part introduces vehicle speed and road traffic congestion detection methods. The third section evaluates existing deployment of traffic monitoring and notification services. The fourth part summarizes existing queue-end detection and notification mechanisms. Finally, the fifth section discusses how cell broadcasting is currently used in public safety and warning systems.

Q-end refers to the tail end of the congestion or the location of standstill vehicles (i.e. stopped traffic). The word "congestion" refers to traffic that is either slower than normal or that has come to a standstill. Congestion warnings may not be enough to prevent rear end collisions. In order to avoid rear-end collisions, it is important to know the exact location of the q-end. Since location is key to both congestion and q-end problems, more emphasis has been made on location detection.

2.1 Different Location Determination Methods

There are many ways to detect location, such as by using handheld tools, signalling strength of wireless devices, by capturing signalling messages of a cellular network, or by looking at maps. Brief description of different location determination methods are presented below:

Handheld devices

Global Positioning System (GPS) – This device is one of the most common ways of finding location. The many technologies within GPS vary in accuracy and in the amount of time they take to determine locations. For example, WAAS enabled GPS, DGPS,

Autonomous GPS, Assisted GPS (A-GPS) all use GPS satellites to find positions, but varies in the techniques they use, the accuracy they provide, and time they take to synchronize with satellite.

User Equipment – Phones and devices that connect to the cellular network will be referred to as User Equipment (UE) or devices. It should be noted that some non-smart phones do not allow the retrieval of location information by the user or by applications running on the phone, and that the network provider can also enable or disable the capability of retrieval of location information. However, most 2.5 and later generation phones are equipped GPS modules. The majority of the smarter phones of these newer generations make the GPS information available to the user or to any of the phone's application regardless of the service provider's location policies. The user can manually send GPS coordinates via SMS, email, or instant messaging to others, and the location information can also be retrieved and sent automatically by applications running on the phone. The GPS information retrieval of smart phones function even when there is a network outage or there is no network coverage. Also the phones' alternate unlicensed wireless modules such as Bluetooth, WiFi, ZigBee or USB connections can be used to retrieve location information from other devices including standalone GPS devices, carputer, or the vehicles' OBD system.

Location determination using signal strength of wireless devices - There are many handheld devices beside cellular phones that are equipped with one or more unlicensed wireless modules like Bluetooth, WiFi, or ZigBee. These devices communicate with either an access point or another wireless device. All wireless devices emit radio

frequency (RF) signals. It is possible to capture signal strengths and other RF properties in order to find their location.

Table 2-1 Comparison of location accuracy using different technologies [8]

Positioning Techniques	Accuracy (meter)	Pros	Cons
Autonomous GPS		- Location information available to UE	- Very slow response time - Not very accurate
Assisted GPS (A-GPS)	5	- Quick response time - Good accuracy - Location information available to UE	- Require some information from the network
Advanced Forward Link Trilateration (AFLT)			
Serving Mobile Location Centre (SMLC)	50	High	- Not available to UE - Requires lot of network equipment resources. - Not design for large volume of location requests
Cell ID	100 – >3000	- Readily available -Easy to retrieve	- Not available to UE - Poor accuracy - Accuracy depends on cell size
Cell ID + Timing Advance (TA)	500		- Not available to UE - Poor accuracy
Angle of Arrival (AOA)	100 – 200		- Not available to UE - Poor accuracy
Time of Arrival (TOA)	100 – 500		- Not available to UE - Poor accuracy
Time Difference of Arrival (TDOA)	50 – 200		- Not available to UE - Poor accuracy
Enhanced Observed Time Difference of arrival (E-OTD)	100 – 400		- Not available to UE - Poor accuracy - Requires modification of UE
Carputer	Unknown		- Often proprietary to vehicle manufacturer - Not available in most vehicles
Maps	Unknown		- Time consuming - Manual process
Standalone GPS Navigator	< 3 WASS 3-5 DGPS 15 Normal	GPS Satellite - Stand alone - Available to user	- Requires line of site

There are different techniques that use different RF parameters to reveal position of a wireless device. The techniques include: Advanced Forward Link Trilateration (AFLT), Cell identity, Cell ID + Timing Advance (TA), Angle of Arrival (AOA), Time of Arrival

(TOA), Time Difference of Arrival (TDOA), Enhanced Observed Time Difference of arrival (E-OTD) and other variations of these methods. Many of these techniques are used by different network types (licensed or unlicensed) to find UE location. Detailed information on these is beyond the scope of this thesis. A summary and comparison of different techniques are provided in Table 2-1.

Location determination by capturing network signalling messages

This subsection begins with a brief introduction to the cellular network followed by an explanation of how signalling information in the cellular network can be used to locate a UE.

The majority of today's cellular networks are based on three major technologies: Code Division Multiple Access 2000 (CDMA2000), Global System for Mobile communications (GSM) and the Universal Mobile Telecommunications System (UMTS). However, all of the major technologies are evolving towards Long Term Evolution (LTE), the next generation of UMTS. GSM first evolved to as UMTS release 1999, then to R4, R5, R6 and R7. In this thesis, UMTS release 1999 to R7 will be referred to as UMTS and release 8 and beyond will be referred to as LTE. Figure 2.1 shows components and logical connections of a basic UMTS network. In this thesis the terms "mobile network" and "cellular network" are used interchangeably. Both refer to Public Landline Mobile Network (PLMN).

All PLMN are divided into Radio Access Network (RAN) and core. RAN is also known as Access Stratum (AS) and the core as non-Access Stratum (NAS). In GSM, the RAN consists of Base Station Controller (BSC), Base Transceiver Station (BTS), and antennas.

The BTS and the antenna(s) make up a cell. A BTS controls one or more antennas. A BSC controls one or more BTSs and is usually located close to the core. In a UMTS network, a BSC is replaced with Radio Network Controller (RNC) and the BTS is replaced with NodeB. A BSC connects to one or more Mobile Switching Center (MSC) and the RNC connects to MSC Server (MSS) and Media Gateway (MGW), or to the MSC in the NAS side. In the case of LTE, RNC is removed and its functionalities are distributed between Enhanced Node B (eNodeB) and the Mobility Management Entity (MME) a component of the core.

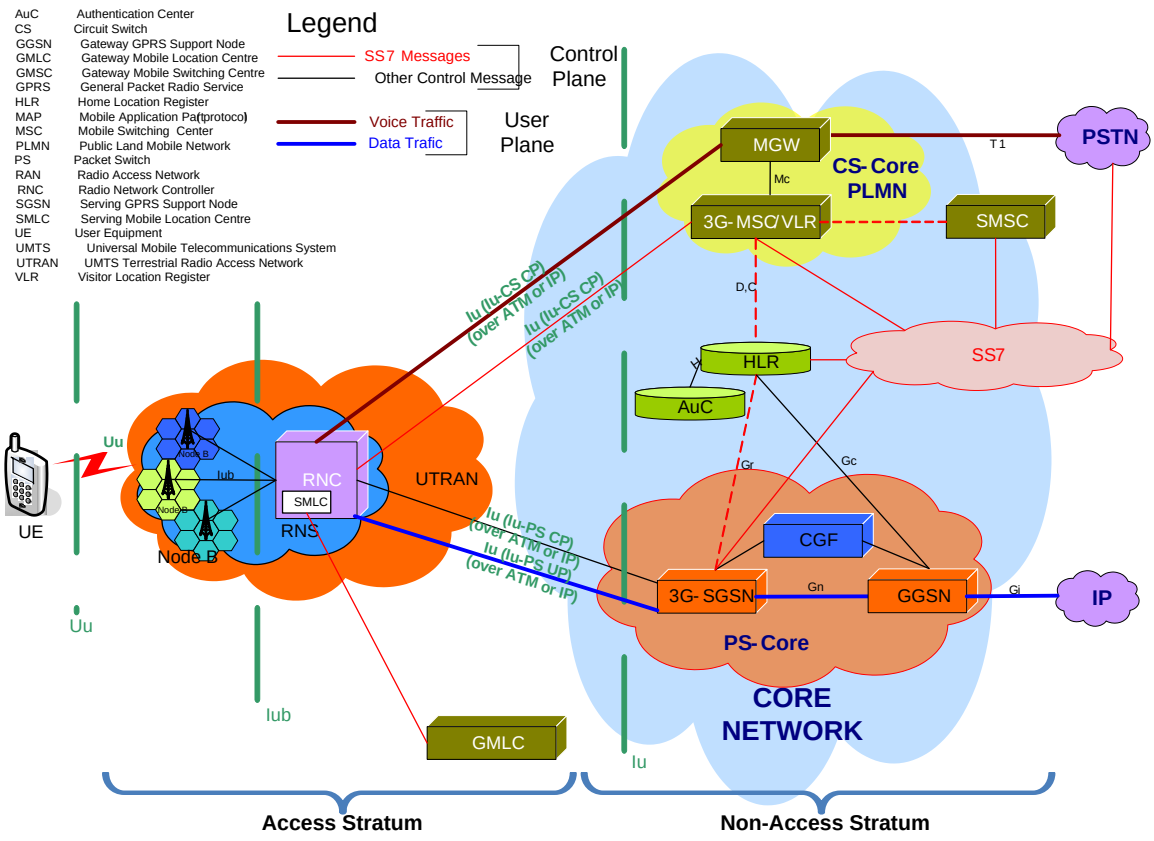

Figure 2.1 Basic UMTS network diagram [9]

A cell is the smallest entity of the PLMN. Every cell in the PLMN has a globally unique cell ID. In case of GSM and UMTS networks multiple cells are grouped to a Location

Area (LA). In UMTS, the circuit core subdivides the LAs into one or more Service Areas (SA), and the packet core subdivides the LAs into one or more Routing Areas (RA). In LTE the cells are grouped in Tracking Areas (TA) instead of SA and RA.

The BTS, BSC, Node B, RNC, and eNodeBs track the unique identity of the cell a UE is registered in as long as the UE is powered on. The UE itself tracks its own dwelling cell ID. Each time the UE moves, it registers to the new cell whether or not it is idle. When a UE is register to a cell the AS elements are notified of the identity of both the UE and the servicing cell. The NAS elements do not necessarily get notified each time a device moves from one cell to another. However, in case of a handover (transfer of a cell during a voice or data conversation), the GSM core tracks both the cell ID and LA identity, whereas the UMTS core tracks only the SA Identity, but not the cell ID, and the LTE core (MME) tracks only the TA. Certain NAS elements also get location updates, be it Cell ID, SA, or TA information during following occasions: periodic location Update, IMSI Attach (phone powered on), IMSI Detach (phone powered off), Mobile-originated call, Mobile-Originated SMS, Mobile-Originated supplementary service operation, response to paging, after release of a call and whenever the UE changes its location area or tracking area.

Every time a UE moves from one cell to another cell there are signalling messages containing the UE ID and the cell ID that takes place between the UE and the BSC, or RNC (via BTS or NodeB) or eNodeB. During handover and location updates there are additional signalling messages that place between the UE and certain NAS elements containing the UE id and either one of cell, SA, RA, or TA identities. Note that, in many UMTS deployments, the numeric value of a 1 octet long SA code and a 1 octet long cell

ID are the same. Therefore, it is possible to locate a UE to cell level from the signalling messages.

```
       Iub - NBAP-1   RadioLinkSetupRequest SCTP    NBAP    RadioLinkSetupRequest
Captured on Thursday, January 27, 2011 15:45:23.496851; Size 222 Octets

Ethernet               Title: IEEE 802.3, Spec: IEEE 802.3, Rev: 2000
IP(Lower)              Title: IPv4&IPv6(Lower), Spec: RFC 791 & RFC 2460, Rev: RFC 791& 2460
...
       28      Protocol=SCTP
...
       63          Payload Protocol Identifier=DUA


NBAP           Title: 3GPP R7 06-2008 NBAP, Spec: 3GPP TS 25.433 (v7.9.0), Rev: Release 7 v7.9.0 (2008-06)
       67      PDU EXTENSION=0; PDU Type=InitiatingMessage;   Spare=0
       68      Procedure Code=id_radioLinkSetup
       69      ddMode EXTENSION=0; ddMode=fdd; Criticality=reject; MessageDiscriminator=common; TransactionID
TYPE=longTransActionId;   Spare=0
       70      LongTransActionId=5242
...
       103     gainFactor_fdd EXTENSION=0; iE_Extensions OPTIONAL=0; betaC=12; betaD=15
       104     UL_DPCCH_SlotFormat EXTENSION=0; UL_DPCCH_SlotFormat=0;   Spare=0
       105     UL_SIR=4.0db
...
       156     TransportFormatSet_ModeDP EXTENSION=0; TransportFormatSet_ModeDP TYPE=notApplicable;
TransportFormatSet_Semi_staticPart EXTENSION=0; codingRate OPTIONAL=1; iE_Extensions OPTIONAL=0;
TransportFormatSet_TransmissionTimeIntervalSemiStatic EXTENSION=0;
TransportFormatSet_TransmissionTimeIntervalSemiStatic=msec_10

...    179     RL_InformationItem_RL_SetupRqstFDD EXTENSION=0; propagationDelay OPTIONAL=1; diversityControlField
OPTIONAL=0; sSDT_Cell_Identity OPTIONAL=0; transmitDiversityIndicator OPTIONAL=0; iE_Extensions OPTIONAL=1; RL_ID1=0
       180       Spare=0
       181     C_ID1=2151
       183     FirstRLS_Indicator EXTENSION=0; FirstRLS_Indicator=first_RLS;   Spare=0
       184     FrameOffset=6
       185     ChipOffset=20992
       187     Propagation Delay=6
       188     FDD_DL_CodeInformation COUNT=1; FDD_DL_CodeInformationItem EXTENSION=0;
transmissionGapPatternSequenceCodeInformation OPTIONAL=0; iE_Extensions OPTIONAL=0; DL_ScramblingCode=0
       189       Spare=0
       190     FDD_DL_ChannelisationCodeNumber=24
       192     initialDL_transmissionPower=-5.7dB
       194     maximumDL_power=0.0dB
       196     minimumDL_power=-15.0dB
...
       208     1st BindingID=44920
...
IP(Lower)              Title: IPv4&IPv6(Lower), Spec: RFC 791 & RFC 2460, Rev: RFC 791& 2460
Ethernet               Title: IEEE 802.3, Spec: IEEE 802.3, Rev: 2000
```

Figure 2.2 Excerpt of signalling message between node B and RNC

A cell could be as small as part of a building (pico or femto cell) or as large as 40 km in diameter, but typically it measure 1.5 km in diameter in most urban areas. However, the location accuracy of a UE can be improved by other parameters present in these signalling messages. Figure 2.2 contains an excerpt of a Node B Application Part (NBAP) packet that was captured from a live UMTS network. This messaging took place between a node B and RNC and was captured from the Iub interface of the RNC. Notice that the captured packet contains the cell ID, transmission power levels, and propagation

delays. These transmission power levels and propagation delays can be used to calculate the distance from the tower to the UE more accurately. The distance can then be cross-referenced with known tower location to find more accurate UE position.

BTS, NodeBs, and eNBs have a direct logical connection to the BSC, RNC, and MME respectively. Therefore, by tapping network links only in the core or at RNC/BSC it is possible to capture the required messages to determine a UE's location. The majority of all signalling messages between different network elements are standardised (by 3 GPP for GSM, UMTS, and LTE), making it possible to deploy this solution almost anywhere in the world relatively quickly and cheaply. An additional benefit is that this method is also immune to users attempting to manipulate location information. Another advantage to this mechanism is that it does not add any extra load or network traffic to the PLMN infrastructure, and it also allows the location of road traffic to be determined in near real time.

The main drawback of this method is these control plane messages do not include the location of the UE beside the cell ID. Although the transmission power and propagation delays can be extracted from the signalling message to enhance the determination of UE position, it is still not practical for q-end type solutions. This is due to the fact that the system will have to constantly track the location of each UE, look for certain message types, and recalculate the location, which is very CPU intensive. This solution will not work well for LTE, because the NAS only receives the TA information which may consist of multiple cells. Also the NAS messages will not contain transmission power or propagation delay information, since the power controls for UEs are done by the eNodeBs. For this reason taps will have to be added to each cell sites and backhauled to

the correlation engine, which could be very expensive. Also, this resolution is dependent on the cellular network, such that if cellular network fails so will the solution.

Location determination Internally by Mobile Core Network equipment

Every Public Land Mobile Network (PLMN), be it CDMA, TDMA, GSM, UMTS or LTE, consists of Service Mobile Location Center (SMLC). SMLC is responsible for calculating the position of the UE. The SMLC can determine the approximate location of any type of phone (smart, non-smart, with or without GPS module) that is registered or served by the network, and even works for phones without a subscription or roaming agreement, and in the absence of any SIM or USIM card.

SMLC mandates the UE to provide certain information required to calculate the location of the UE. SMLC uses standard methods (mentioned in previous section i.e. triangulation, triliteration) to calculate the location. The method used by the SMLC to perform calculations or the information it requires from the UE is beyond the scope of this thesis. SMLC is used to determine the location when a mobile user calls the Public Safety Answering Point (PSAP) for emergencies (i.e. 911 call in North America, 211 in European Union, and 999 in United Kingdom). The PSAP uses the Gateway Mobile Location Centre (GMLC) interface to communicate with the SMLC in PLMN. The moment an emergency number is dialed the PSAP automatically triggers the SMLC, which then sends a message to the UE to provide the data required to calculate its position.

According to many government regulations for Enhanced-911 (E-911), the location calculation by SMLC has to be accurate within 50 meters for 67 percent of emergency

calls and 150 meters for 95 percent of the calls. There are several problems with using SMLC to find UE location for the q-end detection. First of all, 150 meters is not accurate enough for q-end purposes, especially considering that some GPS technology can provide up to 3 meters of accuracy. More importantly, SMLCs are designed and optimized primarily for emergency calls only and are not capable of handling a large volume of position requests. Additionally, SMLC also has a high dependency on its network elements which may malfunction during a catastrophic situation due to high usage volume. For these reasons, SMLC is not suitable for identifying traffic information for q-end detection.

2.2 Congestion and Speed Determination Methods

The first part of this sub-section will introduce traditional tools used to detect congestion and traffic status, followed by other recent developments to determine traffic situations using wireless technologies. Congestion can be detected or predicted in many ways. One of the oldest methods is human observation of live traffic from the road or from a traffic post. Congestion can also be revealed by counting the number of vehicles entering or exiting from two given points, the times of entry and exit, the vehicle length, and the distance between the two points. If the locations and corresponding times at the locations are known, then its speed can be calculated. Once the speed is calculated, congestion and its severity can be determined by comparing the calculated speed with the expected speed (i.e. posted speed). If there is no change in location, it will mean that the traffic has come to a standstill. Congestion can also be estimated or predicted using historic data combined with other factors like weather, events, time of day, day of the week, time of the year, and other such probabilities.

There are many tools available that collect information required to determine traffic status (i.e. vehicle presence, length or classification, count, speed or location). Traditional methods include manually counting and installing different probes, sensors, or cameras. More recent methods include use of mobile devices (i.e. cellular phones, data sticks, and other wireless enabled devices) as probes or tools to detect road congestion.

Determination of traffic condition using traditional method/tools

Human counting and observation is the oldest method of determining traffic problems. This method requires a trained observer to count and classify road users (i.e. pedestrian, different vehicle types - cars, trucks, bus, bicycles etc.) using tally sheets, mechanical count boards, electronic count boards, or handheld radars [8]. This method is limiting due to safety concerns, cost (i.e. need a lot of trained people to see the big picture), and unpredictable weather conditions.

The detector technologies can be grouped in three categories based on their implementation methods: intrusive detectors (in-roadway), non-intrusive detectors (above roadway or sidefire) and off-roadway technologies. “Intrusive detectors are installed within or across the pavement on roads and bridges” [11] and they become part of the road surface. This type of sensor requires a road and/or a lane to be closed for a long period of time during the installation and maintenance of the sensors, which causes congestion in itself and other indirect costs. This type includes: Inductive Loop, Magnetic Detector, Pneumatic Road Tube, Piezoelectric, and Weigh-in-Motion (WIM) sensors. Non-intrusive sensors do not become a part of the road and “can be installed above or on the sides of roads and bridges [11].” Installation and maintenance of this

type causes little or no disruption to traffic flow. Examples of non-intrusive detectors are: Active and Passive Infrared, Microwave Radar, Ultrasonic and Passive Acoustic, Video Image Processing (VIP), and Combined Detector Technologies. [8][11]

Intrusive or non-intrusive detectors typically provide secure, accurate, and unbiased traffic information, but they are very expensive to install and maintain, and limited only to the location they are installed in. According to [12] it costs about $15,000 USD to install a single traffic sensor and 500 - 600 sensors are required to cover 2500 miles of road.

Off-roadway technologies include Probe Vehicle (also known as Floating Car Data) and Remote Sensing. Probe vehicles are equipped with in-vehicle devices such as Global Positioning System (GPS), cellular phones, Automatic Vehicle Identification (AVI) or Automatic Vehicle Location (AVL). Remote Sensing technology uses arterial or satellite images to analyze and extract traffic information. More information including features, pros, cons and relative costs of these categories of sensors or detectors technologies and other sensor types (i.e. Pedestrian and Bicycle Detection) can be found in [8] and [11].

Determination of traffic condition using mobile and wireless technologies

There are many different ways a mobile or wireless network can help gather traffic information and alert people of traffic problems. Mobile networks can act as the carriers for traffic information and wireless devices can retrieve traffic information from a probe or another device. The device itself can also act as a probe and copies of regular control plane messaging between the mobile device and the network can be used to determine traffic information.

The technologies mentioned in this section also fall under off-road technologies or floating car data. The wireless technology used can be either satellite (i.e. GPS) or terrestrial-based (licensed or un-licensed) (i.e. TDMA, CDMA, GSM, UMTS, or LTE based cellular or IEEE 802.11 network) [13]. Contributing devices in this category include any cellular or mobile (CDMA, GSM, UMTS, LTE) phones or data sticks, any smart devices (laptop, tablets, portable media player (PMP) or digital audio player (DAP)) with any type of wireless communication methods (BlueTooth, Wi-Fi, ZigBee), any GPS based navigation system and carputers. The desired location or speed information can be retrieved from either the devices themselves (i.e. GPS receiver, GPS enabled phones, carputer etc.) or from the network infrastructure providing the wireless service using methods mentioned in previous section. By comparing the actual and expected speed congestion can be detected.

[13] provides a way to approximate speed and location of Bluetooth enabled mobile devices using wireless mesh networks (WMN). In order to calculate the speed the Media Access Control (MAC) address of the mobile device, the Received Signal Strength Indicator (RSSI) from the mobile device, the time of device detection, and the identity of the access point (AP) are all sent to a server using the WMN. The server receives this information from every AP that was visited by the mobile and contains the locations and IDs of all the APs that are configured to send this information. The server also correlates the information received from different APs and the IDs of mobile devices. By doing so, the server calculates the speed and recognizes the path traveled by the mobile. It can also determine the approximate location within a maximum error margin of 100 meters (maximum Bluetooth coverage) or less depending on the class of BlueTooth used.

Although [13] used open-source resources and heavily deployed BlueTooth technologies, it may not be practical to expect any open device to contribute to this method. Each contributing AP must be programmed to send specified information to the server, and the server must contain a list and the location of each of the APs, which may not be possible. Most BlueTooth APs are owned and managed privately and there is no assurance that the owners are willing to provide the information to be loaded in the server. However, this technique would be more feasible if used for Wi-Fi APs that are deployed on lampposts by a service provider. In that case the AP could already be provisioned at the server by the service provider.

[14] provides an algorithm for estimating traffic using GPS technology and short wave radio. In their research, they divided the areas into equal sized squares. The trial GPS receivers were installed in 4000 taxis, each of which sent their location, taxi ID, heading, speed, and timestamps to a server at regular intervals using shortwave radio. They calculated the traffic speed by dividing position difference with time difference. This solution require addition hardware and infrastructure and therefore additional cost. In [15], researchers used Bluetooth and GPS enabled phones to gather real-time vehicle information. The application on the phone extracted vehicle information (driving and engine performance) at regular intervals and used the phone's GPS receiver to add geocode to the data. It then sent the concatenated data to a server using either Wi-Fi or the cellular network, which then used this information to determine congestion and alert the users. There is no mention of q-end detection in their work. [16] proposes the use of GSM Location Services (LCS) and AGPS enabled mobile devices for collecting traffic information because GSMs LCS alone does not provide the position accuracy. Also,

regular GPS receivers do not provide accurate information on position in urban areas and do not synchronize quickly, but AGPS does. This paper also did not address q-end problem.

Pattara-atokom in [17] and [18] in their other works used cell dwell time and artificial intelligence to estimate road congestion. An application running on a GPS enabled phone collected time, LAC, CID, cell dwell time (CDT), and GPS (location, speed, and bearing) information and sent them to a server using the wireless network. The phones sent the cell data (LAC, CID, CDT) each time it moved from one cell to another, whereas, GPS locations were sent every second. The CDT was calculated by the application in the phone. This approach requires sending data (although a small amount) at least once a sec by a large number of phones, which could contribute to a significant amount of network traffic, and it was concluded that more work needs to be done for proper traffic or congestion estimation.

[19] offers a mechanism to correlate GPS location information from a mobile phone to spatial location on the road network and estimates traffic information using fuzzy logic, neural network, Kalman filter, and D-S reasoning. In addition, they use clustering to identify if more than one phone is in the same vehicle and provide a way to distinguish if a mobile device is in a vehicle, or not by using a speed threshold. Finally, they used an RBF network to adjust the traffic load for roads with vehicles that did not contain a cell phone. The author assumes that the network provider will provide the time and GPS coordinates, but it does not mention how. The solution offered in [19] is for gathering traffic information, but not for detecting congestion or q-end problems.

[20] exhibits a macroscopic estimation method of urban traffic network. It uses Cell Transmission Model (CTM) to model the traffic, which divides the roads into segments (i.e. cell, hence the name CTM). The author also makes a comparison of the macroscopic model with the microscopic model, proving that the macroscopic model provided better estimation of urban traffic than the microscopic model. [21] presented another CTM based model for estimating freeway traffic density. They used different filtering methods to tune the model. However, they did not address the determination of q-end problem.

Cell dwell time and RF signal strength can also be determined by capturing certain signalling link messages (as shown in section 2.1). Therefore, by applying the methods mentioned in [13], [17] and [18] it is possible to use the mobiles as probes and as congestion detectors.

Vehicle to Vehicle (V2V) and the Vehicular Ad-Hoc Network (VANET) are also getting a lot of attention in terms of vehicular safety and congestion detection. IEEE 802.11p is the standard behind V2V and VANET. As the name suggests the standard allows the vehicles to communicate with each other while moving at high speeds. The objective of the standard is to have a 1000 meter range. According to the study, a distance of 750 meter provides only 90% reliability when driving at a speed of 120km/h. [22] provides references to other research being done on traffic congestion detection using V2V and VANET. The main drawback of this technology is the fact that very few high end vehicles have adopted or implemented it, it will be a long time before the technology can be widely used to provide adequate road safety.

2.3 Existing Traffic Monitoring Systems/Traffic Information Provider

This section lists some of the live traffic service providers, the technology they use to gather traffic information, and the technology they use to distribute the information to the user. Please be advised that most providers do not share the detailed information or the method on how they gather or distribute the information.

Currently there are many providers that offer live traffic and congestion information. These services are provided using various methods. Some providers offer the service over the internet, some using Satellite signals, some using SMS, some using broadcast radio and television, and some using one or more of these methods. Many of these providers also offer applications for mobile devices through cellular or WiFi network. How these providers get the updated traffic information also varies. They use one or more sources and methods mentioned in the previous section to collect the information.

NAVTEQ (also known as www.traffic.com) uses the NAVTEC Traffic™ advanced data collection and processing tool. NAVTEQ combines and processes data from four different sources, including (a) proprietary digital traffic sensors deployed in major metropolitan areas, (b) GPS and probe devices, (c) commercial and government partners, and (d) their own local traffic operations centers' staffs. The staff listens to police and fire scanners, monitors traffic cameras, and gathers traffic information from driving cars and airborne helicopters and fixed wing aircrafts. [23] NAVTEQ provides services only in areas that they have sensors deployed or have operations centers. As of October 2011,

NAVTEQ has provided service to 130 markets in US and Canada, although only to a few Canadian cities.

NAVTEQ delivers their service over terrestrial and satellite radio, broadcast and cable TV, wireless applications and services, and over the Internet. They partner with different service providers, such as Sirius XM radio (NavTraffic), and Garmin. NAVTEQ also has a partnership with auto manufactures to provide navigation services. The services provided by NAVTEQ include continuously updated information on average traffic speed, incidents, construction, and road closings. They also provide delay estimates, rerouting information, and total travel time for a given route. [24]

TrafficLand (www.trafficland.com) traffic service is based on video cameras and provides a graphical image of the streets for the cities it serves. As of October 2011, it provided service to only parts of US, very few cities in Canada, including Halifax, Hamilton, Niagara, Ottawa, Sarnia and Toronto, as well as five other countries. However, there is no indication that the technology has the capability of showing the average speed of vehicles on the streets it serves and there is no indication of its ability to locate q-end. However, the company`s website provides congestion information and maps of the areas that they have cameras in using Google maps. The video images are available via XML feed or over the internet, and through personal computers and mobile devices. TrafficLand also provides traffic images to iPhone applications. [25]

A picture may be worth 1000 words, however they are not as useful to road safety. Instead, they make the road more dangerous, since pictures and video images draw people`s attention away when they are driving.

INRIX (www.inrix.com) collects traffic related information from millions of GPS-enabled vehicles and mobile devices as well as from traditional road sensors. They use their unique "Smart Driver Network" to aggregate this information, although the exact method used to collect this information and the nature of the information collected is confidential. They have relationships with auto manufacturers, mobile developers, and transportation agencies to obtain their traffic data. Some of these sources include consumer devices in taxis, airport shuttle services, cars, heavy goods vehicles and long haul delivery trucks. They also have partnership with Vizzion to extract traffic camera images for their users.

INRIX uses different tools to provide different services and claims to provide accurate real-time, historical and predictive traffic services on freeways, highways, and secondary roadways, including arterial and side streets wherever they have establishments. They also provide speed, congestion, and bottleneck information. In addition INRIX supplies accident, construction, road closure, and event information. They offer information of the fastest route as well as the estimated travel time which takes such factors as the current traffic conditions, the day of the week, the weather, holidays, accidents, special events into consideration. INRIX offers the driver with both audio and visual warnings stating "congestion ahead."

As of October 2011, INRIX served eight countries with the plan to deploy in over 20 countries by 2011. Their service is available through websites, mobile phones, and vehicle navigation systems. INRIX also partners with Dash Navigation, Inc, MapQuest, ARC Transistance (Europe), TCS Inc., TomTom and other companies to distribute and collect traffic information. Some of these providers use their web resources and

applications to distribute real time traffic information. For example, MapQuest offers the service over the internet as well as through mobile applications. TomTom partners with TrafficCast and owns TeleAtlas, and offers traffic services over GPS devices. [26]

Cellint (http://www.cellint.com) is another traffic information provider whose product name is TrafficSense. They use floating car data (FCD) to determine traffic information. TrafficSense performs pattern matching on data extracted from the signalling links for mobile networks using their proprietary patented technology. They collect the signalling information in real-time, enabling them to provide real-time traffic data without installing sensors. Beside real-time traffic information, they also provide "historical databases of speed and volume estimation, continuous over weeks, months and years, as well as and origin destination statistics" [27]. Their system is capable of detecting lapses in traffic flow within couple of minutes and has an error rate of less 10% in terms of calculating travel time. Information regarding their testing methods and results, system functionality and architecture can be found in [28] and [27] respectively.

Cellint provides their services to mobile phone subscribers, navigation providers, government agencies, road operators, mapping and media portals. They also provide data back to the cellular service provider in order to optimize RF performance, making it mutually beneficial situation for both Cellint and cellular service provider.

Intelligent Mechatronic Systems Inc. (IMS) Traffic (IntelliOne or http://www.intellimec.com/traffic/) also uses ordinary mobile phone network data for collecting live traffic and congestion information, similar to Cellint. The actual patented technology was created by IntelliOne and was acquired by IMS. IMS combines the

mobile signal data with road sensor data and other data feeds to offer more accurate traffic information to their users. The detailed information of the methods, sources, and nature of wireless signals used to calculate traffic information are not available.

IMS primarily provides congestion data and works with governments, transportation authorities, and public sector organizations to jointly develop Intelligent Transportation Systems. [29]

Google is also a map and real time traffic service provider that also uses data from mobile phones. Instead of tapping into mobile network signalling traffic or mobile signal data, Google actually uses the smart phones to do the speed calculation and to determine the locations before sending this information to Google. The information is sent only if a mobile user enables "My Location" feature of Google maps. A user can disable this feature and still use the map service. Google also gets traffic data from road sensors, and car and taxi fleets [31]. In order to ensure privacy, Google claims that they only use anonymous speed and location data, and that once they process start and end points of every trip, they delete the information [30].

Google provides traffic congestion information and direction services. Real time traffic information is available only in certain parts of Canada, US, Europe, and Asia. It provides these services over the internet. Google also supports map applications in different mobile phone platform including iPhone, BlackBerry and Android.

Although actual data sent by the Google application is unavailable, it seems the phones transmit regular speed and location information, therefore, constantly using network

resources. In contrast, in the proposed solution device does the congestion detection and sends messages only when necessary.

Concluding remarks on existing services

The purpose of traffic monitoring is to ensure road safety, congestion detection and avoidance, and route optimization. Although there are providers for congestion information, rarely anyone claimed to provide the actual location of the queue end or the location of standstill traffic, and most of these methods rely on the network being up. Many are also limited only to the locations where they have sensors or cameras deployed. As a result, in the case of an incident (i.e. an accident in the middle of highway where there is no camera or sensor), they will not be able to pin point the q-end location created due to the accident. At the instance of the accident, the q-end location will be the point where the accident took place, which will change as more vehicles stop behind the accident. Also in times of disasters when telecommunication networks are heavily congested, most of these traffic monitoring systems will fail.

2.4 Queue-end Detection and Notification Methods

This sub-section concentrates on existing research on q-end detection and notification systems. Finding q-end location is more challenging than just congestion detection. Congestion detection is the first step of q-end location detection, so they share some similar challenges and some common approaches in their detection. Q-end problems are the most dangerous on a freeway or highway where traffic is expected to flow at a high speeds, but due to unpredicted occurrences, traffic is at a standstill or near a standstill (i.e. due to accident). This type of variable is the toughest to detect due to its unpredictability and becomes even more challenging in poor weather conditions.

According to many studies, the abrupt terminations of highway traffic flow causes rear-end collisions, which is a major problem. On the highway, certain sections that are prone to accidents and/or congestion have some indication of hazard in the form of variable signage. However, without proper real time detection technology these signs cannot be very useful. The only way they can provide real time detection is when some sort of detection tool (i.e. sensor) is in place at each section of the road, which is expensive and impractical to deploy. Often emergency vehicles are placed at the q-end location. These vehicles are moved as q-end moves to warn upcoming traffic. In certain weather conditions flashing lights of emergency vehicle could be invisible. However, this is a reactive approach and is often too late to avoid initial multivehicle collisions. It only helps prevent further collisions after some of the damage has already taken place. [4], [32]

Typically, detecting q-end with sensors requires lot more sensors to be deployed compared to congestion detection. The more sensors deployed means higher cost of installation and maintenance. To solve this problem Khan proposed a solution that uses only two sensors, a computer, and artificial neural network (ANN) to detect q-end location in a temporary construction zone [32]. It proposed the use of a variable message sign or other media to warn drivers as well as where sensors are to be placed at the entry and exit points of potential q-end problem locations. The sensors would wirelessly provide the following data to the ANN: volumes of vehicle per lane, vehicle classification (i.e. heavy truck or small cars), and presence. The ANN would compute the q-end length and send updated information to variable message signs or other media. Khan was successful in simulating and calibrating q-end location calculation. However, the result

was based on predicted q-end models and not real-time sensors, and also required manual calibration. The fact that it requires placing the sensors and the computer means it entails preplanning, which makes this solution unsuitable for unpredicted events like an accident.

In [33], Birk et al have proposed another q-end detection and warning solution using a network of autonomous, wireless, on-road sensors and actuators. These sensors are low powered, solar re-changeable units that become part of the road. These sensors also act as road marking units (RMU) and are capable of providing traffic warnings using different light patterns. These RMUs communicate with each other as well as with road side units and open platform server (OPS) to determine and provide traffic information (i.e. q-end) to the road users. These sensors are also capable of communicating with Vehicular Ad-Hoc Network (VANET) via gateways. The main purpose of Birk`s solution is to fill in the gaps for vehicles that do not have a transponder or VANET capabilities. Although the sensors consumed low power, included built-in charging capabilities and were tested in presence of heavy vehicles (including snow ploughs), maintaining adequate power remained an obstacle. Snow and shortened winter days made it challenging to keep up with the power requirement. Although the cost of infrastructure is relatively low compared to other sensor based solutions, it still requires installation and maintenance of the road sensors. Their solution addresses the q-end issue. However, the solution is useless in the roads without the sensors or VANET enabled vehicles.

[22] presents a q-end detection and warning solution using v2v (IEEE 802.11p – the protocol used for v2v) technology. It also cites other research being done in q-end detection and notification using v2v technology. In [22], the author recommends the use

of Cooperative Awareness Messages (CAM) or beacon messages of IEEE802.11p protocol and their processing through fuzzy logic to detect congestion and q-end location. Their simulation shows promising results in traffic condition detection. Although IEEE802.11p is aimed for a range of 1000 meters it may be effective only up to 300 meters, which may not be adequate stopping distance for certain situations. The other challenge with this approach is that, it may be decades before v2v or VANET is widely available for all vehicles.

2.5 Current Use of Cell Broadcast in Public Safety and Warning System

Cell Broadcast System (CBS) consists of the Cell Broadcast Center (CBC) and the Cell Broadcast Entity (CBE). CBC is responsible for connecting to the radio network and controlling which cell or group of cells to deliver the broadcast messages to. Cell broadcast is standardised in 3GPP and is supported in GSM, UMTS, LTE and CDMA networks. It allows sending messages to all UE attached to a single cell, selected cells, a LA, RA, TA or to the entire PLMN which forms the cell broadcast domain. The CBS is not part of the cellular infrastructure, but provides instruction straight to the radio network using dedicated resources that are not used for processing calls or carrying user data. Even the radio channels for cell broadcast are separate. As a result, network congestion does not have an effect on SMS-CB message delivery.

Although the primary intent of CB deployment is for public safety, it is designed for other free or revenue generating services (i.e. advertisement, weather reports, traffic reports).

One of the main points of this thesis is to use Cell Broadcast (CB) or SMS-CB for public warning systems. The ability of these systems to sustain most network congestion, location specific messaging, multi language support, and to support different message priorities makes CB the prime candidate for public warnings. According to [34], a vendor installed cell broadcast services in more than 80 networks in 30 countries through 50 wireless service providers. The same vendor will also be deploying SMS-CB service in the USA to provide Commercial Mobile Alert System (CMAS) messages in order to fulfil the requirements of the Federal Communications Commission Warning, Alert and Response Network Act (WARN Act). Many governments around the world are investigating the use of CB as a public warning system. [35] and [36] provide a comparison of cell broadcasting technology to other public warning technologies.

One drawback of CBS is the lack of an authentication mechanism, which could prolong the message delivery since the process of authenticating a message is time consuming. However, [37] provides a solution that acts as a brokerage and authenticates the content provider over a secure link. The solution will also ensure that the message complies with national and local laws and jurisdiction and check the contracts and network policies.

[38] makes a recommendation on how to display a CB message on a mobile device, since the standard does not specify any rules for the display, beside stating that it is similar to SMS. [39] provides a solution that extends the CB capability from only showing a text screen on a phone to upgrading to an audio visual alarming device. In [40], the project team has developed an application for the Symbian series phones that logs CB messages, time received, the channel number, LAC, and cell ID. The purpose of this Symbian application is to ensure proper delivery of CB messages on a mobile phone.

In [41] Gundlegård suggests that CB be used to warn drivers of potential traffic hazards, including road conditions, accidents/traffic information, road constructions, lay-bys, and air and road surface temperatures. Gundlegård's also suggested the use of national traffic authorities' infrastructure as the primary source to provide traffic information to the CBE. CBE will then broadcast the message to the cell where the traffic complaint originated and to its adjacent cells. He also suggested the use of selective sensor information from a vehicle (i.e. air bag deployment, traction information) to be sent via SMS to CBE. The CBE will then be responsible for delivering the message to a vehicle's current coverage cell and adjacent cells to warn incoming traffic of the potential hazards. He also recommends that these messages be delivered using text-to-speech technology to ensure no distraction for the driver. Gundlegård recognizes the use of GPS in a vehicle, but did not suggest its use in traffic congestion detection.

Concluding remark

Many congestion and q-end detection mechanisms and their delivery mechanisms have been presented in this chapter. Some have provided feasible ways of detecting congestion, but impractical delivery mechanisms and others have presented functional delivery mechanisms, but lack detection mechanisms. At the time the research was conducted, only Trafficsense claimed to provide q-end length to its users. [42]

As mentioned above, there is still a large gap in proper real-time proper q-end detection and making its notification available on every road, especially on highways or freeways. This thesis provides a solution that addresses both issues using millions of smart phones to detect congestion and q-end problem and CBS for reliable delivery. Smart phones are relatively maintenance free and available in almost every street in the world. The use of

smart phones for q-end adds no additional cost to anyone, as people naturally maintain their phones. Also low or nearly zero usage of network bandwidth for congestion detection makes this solution attractive to users, service providers and law enforcement agents. In this thesis, it is assumed that the location and speed detection will be done by the smart phones' internal GPS system, or a GPS device attached to the smart UE.

CHAPTER 3 ARCHITECTURE AND DESIGN

This chapter presents the overall design and architecture of the proposed system including its physical and logical components. The chapter also includes solutions and implementation recommendations for different characteristic of traffic obstructions that are responsible for queue end situations.

3.1 Overview of System Architecture

This section presents a high level overview of the entire system. It introduces different components and their relationship to the system.

A device is the smallest physical entity in the entire system. A device can be, but is not limited to, singular user equipment (i.e. a mobile phone or data stick, a smart phone), a sensor, a GPS device, a computer, or a vehicle's onboard diagnostic system. Each device provides none, one, or more service(s). A device can also simply be a user of one or more service(s) provided by the grid.

A "micro grid" is the smallest logical entity of the overall architecture. In this thesis, it is also called a "node." It is a combination of one or more devices. That is, it inherits all the attributes of a device and introduces several others such as profiles and roles. The physical limit of a *micro grid* member is typically a vehicle or anything within the reach of Bluetooth, ZigBee, or a short cable. It is assumed most *micro grids are* capable of communicating through the cellular network, and are especially able to receive SMS-CB, or communicate through other wired or wireless technologies. There will rarely be a *micro grid* connected using non-cellular or wired connection. Unlike *micro grids*, a device may or may not be able to communicate directly through the cellular network.

A "macro grid" is another logical entity that hosts one or multiple *micro grids*. In this thesis, these *macro grids* are also referred to as "clusters." In the physical domain, a *macro grid* closely resembles the area covered by a cell broadcast zone/domain which consists of one or more cell sites. In most cases, a *macro grid* will not have any permanent *micro grid* members. The actual number of *micro grids* in a *macro grid* will vary based on time of day, events, incidents and many other factors.

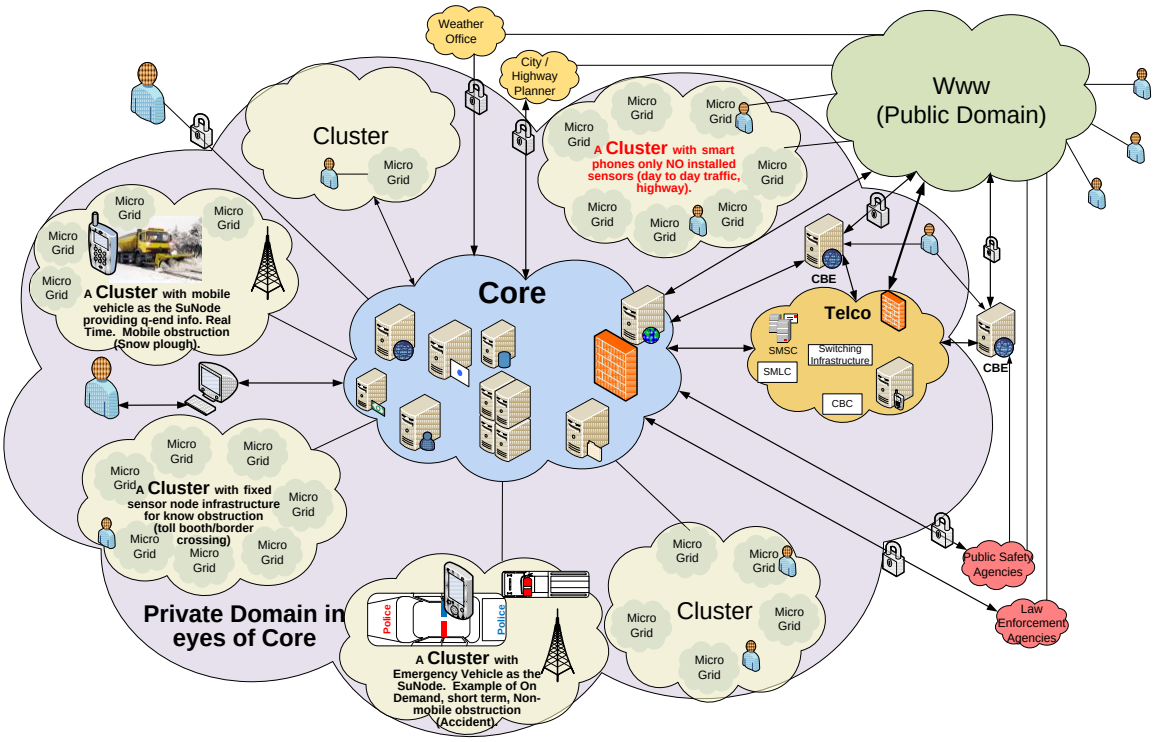

Figure 3.1 Overview of system architecture

Each *macro grid* will have a *macro grid manager*, which will either be a local trusted *micro grid* with elevated resources or the centralized control system. The *macro grid manager* is the liaison between the centralized control system and rest of the *micro grids* in the *macro grid*. Each *macro grid* can either operate as a standalone entity or be controlled centrally. Regardless of operating mode, the *micro grids* will perform their own congestion detection.

Logically the *core* is a single unit representing the centralized control system. It is also referred to as "server cloud". However, physically it is a combination of many servers in a cloud computing environment with geographically diverse locations. These servers host applications, provide different services and interfaces, and act as gateways between the cellular network provider and other organizations. The *server cloud* provides machine to machine, and machine to human interface to the entire system allowing administration work. It is also responsible for managing, provisioning, backing-ups, and maintaining the entire system. A graphical view of the system is present in Figure 3.1. More information on *micro grid*, *macro grid*, and the *core* is provided later in this chapter.

A normal condition is a condition when the traffic flows at an expected speed, and when conventionally functioning cellular network is present. An *abnormal condition* could be any of the following: congestion in road due to rush hour traffic, poor weather, accident, event, disaster, construction work, snow cleaning, transportation of an oversized load, trouble in a cellular network, or the absence of cellular coverage.

In *normal conditions* there will be no upstream or intra *micro grid* communication. The only exceptions to this rule are: a) when a trusted *micro grid* enters a *macro grid* and b) when there are several permanent non-mobile sensors installed with either wired or alternate dedicated wireless connections. However, each *micro grid* will be aware of the contact information of the *macro grid manager*. This will be achieved through broadcasting *heartbeat messages* at a regular interval. Each *micro grid* will constantly monitor traffic conditions, meaning that millions of *micro grids* will independently contribute to road condition detection. The *micro grids* will inform the *micro grid*

manager only when there is traffic interruption or detection of any abnormal condition. This will ensure efficient use of radio resources and reduce interference.

Depending on the type of abnormal condition the *macro grid* will operate either on the standalone mode or the centralized mode. In the case of standalone mode with no or limited cellular network resources the system will use ad-hoc wireless mesh network for intra *micro grid* communication within the *macro grid*. In this mode each *micro grid* will have to register with the *macro grid manager*. There will also be a *macro grid manager* selection process unless a local Super node (*Su)* was selected as *macro grid manager* prior to cellular network trouble.

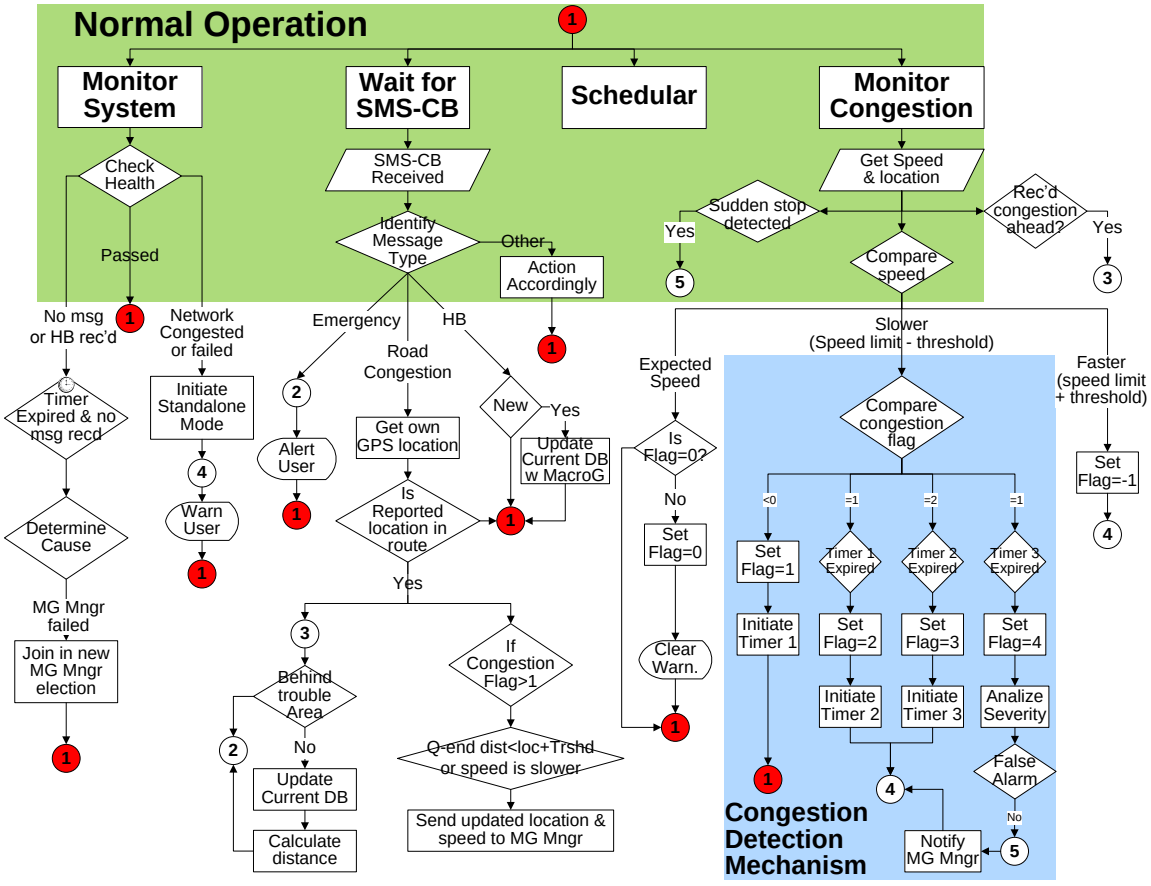

Figure 3.2 How it works - *Micro grid* perspective

An *abnormal condition* due to non-cellular network related issues will use SMS messages or IP protocol for upstream notifications to the *macro grid manager*. The downstream notification to the rest of the *micro grids* will be performed through SMS-CB.

An abnormal condition will be detected in one of following ways:

- Absence of expected *heartbeat* message
- Instruction provided in the content of a broadcast message
- Internal congestion detection mechanism
- Absence of cellular coverage
- Cellular network congestion detection

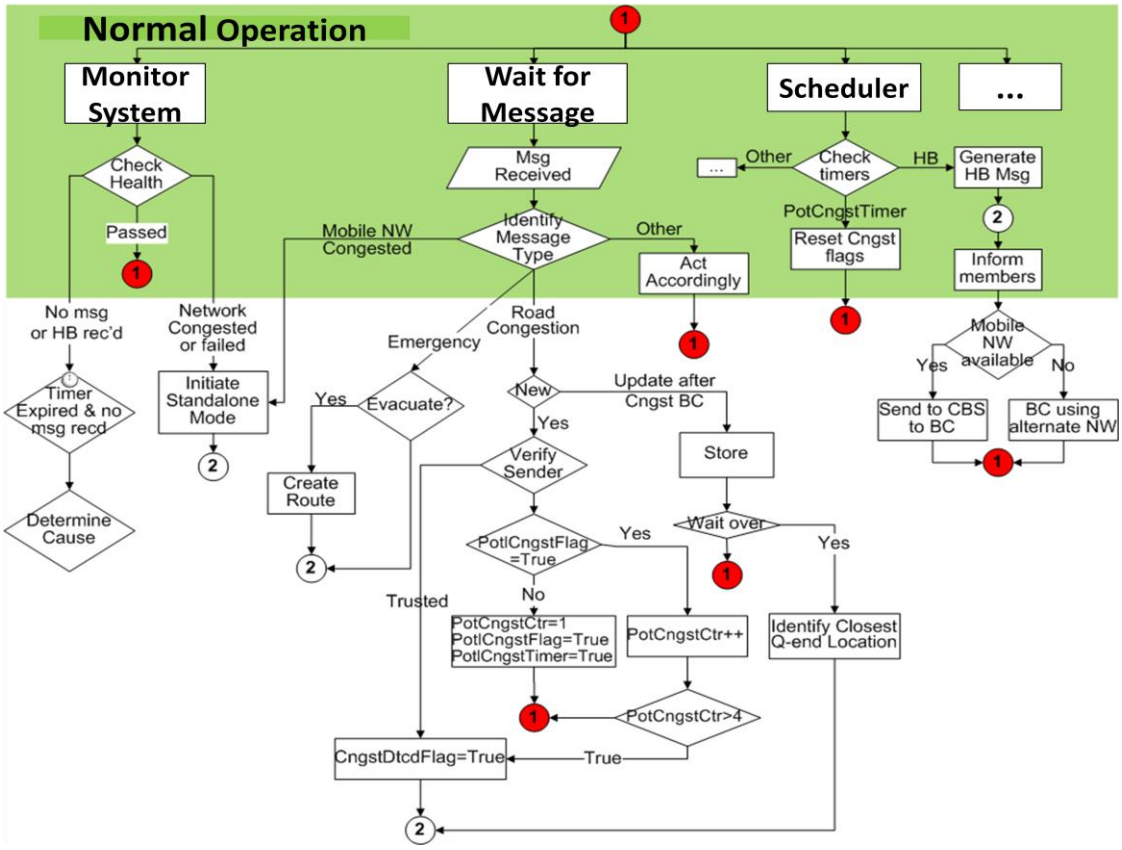

Figure 3.3 How it works - *Macro Grid Manager* perspective

All *heartbeat* messages, traffic and road condition messages, instruction messages, and messages containing alternate route information will comply with CAPv1.2 standards and will meet cell broadcast requirements. Figure 3.2 and Figure 3.3present a graphical view of how the system works.

3.2 Design Assumptions

This section lists the design assumptions made in this thesis. These assumptions are also reiterated at the appropriate sections.

- There will be adequate devices in the grid running the middleware
- The devices within the *micro grid* are assumed to be reasonably stationary relative to each other and to be available to one another while joined in the grid. It is assumed that intra *micro grid* communication is done mostly over Bluetooth, but it could also be accomplished through ZigBee or WiFi.
- Inter *micro grid* communication will take place using ad-hoc mesh network using v2v, Wi-Fi, Bluetooth, or ZigBee technologies when the cellular network fails (i.e. standalone operation).
- Most *micro grids are* capable of communicating through the cellular network, and are especially able to receive SMS-CB, or communicate through other wired or wireless technologies.
- Emergency vehicles will be present during emergencies and be equipped with multiple reliable sensors (speed, weather, road condition monitors etc.), multiple licensed (cellular and satellite) and unlicensed (i,e, Bluetooth, WiFi, ZigBee) wireless voice and data services. In addition they will be equipped with special

device(s) to form an elevated *micro grid* that will take the *Su* role / *macro grid manager* role, and be authorized to deliver traffic updates to the CBE for broadcast.

- In the case of fixed deployment, each sensor (*micro grids)* will have an alternate communication method (wired, Wi-Fi or ZigBee capability) and whenever possible, they will use these alternate direct connections or unlicensed wireless technology for intra *micro grid* communication to offload the cellular network and to increase reliability. The sensors will be place 250 meter apart.
- Many critical roads are already equipped with fixed sensors, warning systems (i.e. dynamic message signs), and additional police patrols.
- Service discovery protocols are already available to the middleware with known APIs.
- Codec used by the application supports minimum 160 characters for 82 byte messages.
- During standalone operation, at least one *micro grid* will be aware of the physical scope of the *macro grid*.
- Both the consumer and the resource provider are aware of any potential security and privacy risks involved when sharing resources (content download).
- People will drive more cautiously during poor weather conditions.
- Route plans, including hop attributes, will be either preloaded or downloaded to the *micro grids* from the *core* or the *macro grid manger*. Step by step escape routes will be broadcasted during emergencies.

3.3 Details of System Architecture

This section provides detailed description of the *core*, *macro grid*, *micro grid* and their components, characteristics and classification.

Description of the Core

The *core* is an important part of the system, comparable to an umbrella that holds all of the pieces together, connecting the fabric of the system. Although the system is designed so that each *macro grid* can operate individually, the *core* plays a significant part in the system and is responsible for several security measures. It authenticates each *micro grid* prior to it joining the grid to provide a service. It also segregates the grid from the external world by maintaining a virtually private network within the grid. Although many devices in the grid will have a direct connection to public internet, the internet traffic from the *micro grids* will not flow through the grid system, which is crucial for security reasons. Also, emergency warning messages will be validated by the proper authority prior to being broadcasted. Otherwise any malicious *micro grid* can take control of the *macro grid* and misinform the users.

The *core* is the interface between the grids and the operators, the telecommunication systems, different agencies (like law enforcement, Public Safety, Municipal, Provincial, or Federal planners etc.), and the weather systems. The machine to human interface of the *core* allows operators to provision new sensors, maintain *micro grid* profiles, and perform system upgrades and maintenance. It provides machine to machine interface between the sensors (be it the grid or an external weather office), speed violator information from the grid to law enforcement agencies system and between different databases and different systems within the *core*. It also provides interface between

different warning systems such as digital billboards (also known as variable message sign), siren generation systems, and other broadcast media.

The *core* notifies *macro grids* about their neighbours, so the adjacent *macro grids* can pass important traffic information among themselves. It is also the gateway between the grid and the telecommunication system providers. For example, the *core* will communicate with the CBS, it will also provide the public a means to view traffic status via the internet by posting the information in web servers.

The *core* is responsible for user authentication and management, *macro grid* management, and service subscription and billing information handling. It will store maps, points of interest information and different contents (i.e. music, firmware, application etc.). The *core* will deliver the above resources to the subscribed end users in case where they are not locally available. It will also provide optimal travel routes, information regarding traffic violations to law enforcement agencies and supply traffic volume information to different government agencies that are crucial for road and highway planning.

Typically the *macro grids* will communicate with the *core* using cellular networks (i.e. SMS or data network). If the *macro grid* loses cellular communication, it may still be able to communicate with neighbouring *macro grids* using other unlicensed wireless networks, which may provide a connection back to the *core*.

The *core* consists of following servers: Application Server, Database Server, Web server, Certification server, E-Commerce Server, Content Server, File Server, Real Time Communication server, and Firewall.

Components of the Core

Although detailed information of different components of the core is beyond the scope of this thesis, a brief description of each component is provided below:

The **application server** will execute all requests from different grids and user interfaces. These requests may come through another server, machine to machine communication, or through man-to-machine communication. It will be responsible for communicating with all the servers depending on the request received. For example, it will contact the database server for looking up subscriber profiles, and lists of services. It will use the *certification server* for grid authentication and *web server* to post traffic updates to the World Wide Web (internet).

The **Database Server** will contain different databases required for the grid to function. It will contain different *micro grid* profiles and other grid information. The **Web server** will provide user interface to and from the system through the world wide web/internet. The **Content server** is responsible for storing map graphics, firmware information, and downloadable content. The **File Server** will provide secure and regular file transfer protocol (ftp) service for smooth download experience. **Real Time Communication server** will be responsible for the auto generation of traffic information.

The **Certification server** will be responsible for *micro grid* certification, authentication, and key exchange whenever applicable. The **E-commerce server** will be responsible for billing, purchase of new service(s), or modification of existing service(s) or subscription. The **firewall** will secure and protect the system from unauthorized access.

Macro Grid / Cluster Definition

In order to understand the architecture of the system, it is important to first understand what is meant by a *macro gird* and its relationship with geographic maps and the locations of mobile towers. Ideally, the coverage or scope of each *macro grid* will be a single cell. In terms of practicality, there will be cases where two or more cells will provide the coverage for a single *macro grid*. There will also be cases where more than one *macro grids* will be served by a single cell.

From the CBS perspective, each *macro grid* will be part of a single cell broadcast domain to which it will have to send messages. However, it is inaccurate to define the scope of a *macro grid* based purely on cell broadcast zones or domains. There will be cases where multiple *macro grids* will belong to a single broadcast domain and there is always some overlap in cell coverage over a given area. The planned and actual shapes and sizes of the cells do not match. For example, the shapes of cell coverage areas are never a perfect circle, semi-circle, triangle, or hexagon as shown in the cell plans. The number of cells or the size of a cell in a tower, are not unique either and a cell can be split to multiple smaller cells.

The implementation of pico cells, femto cells, and dual carriers in a single cell site introduces a complete overlay of a cell on top of another. This also presents a new set of challenges. One solution is to not include one of the overlapping cells as part of the broadcast domain. However, the problem of doing so is that the devices that will be connected to these cells may miss emergency messages. Regardless of overlay a UE will always be connected to a single cell with the exception of dual cells, where a UE will connect to two cells at once. Although traffic information is not needed inside a building

(i.e. femto or pico cell), it is important that everyone receives the emergency messages. An alternate approach is to have different broadcast domains for traffic service and emergency services, but there are some limitations that will prevent use of this approach. Additionally, having multiple domains for different applications will add significant administrative overhead.

To mitigate these problems, each *macro grids* will be designated a pre-assigned boundary defined by a polygon, or one or more triangles of different sizes to fit the broadcast zone as closely as possible. Circular shaped *macro grid* will be rarely used. In border areas, a device may be connected to a cell that is part of a different broadcast domain than its current location. Another issue that often occurs in a border area is that the UE can register and deregister back and forth between same cells. To resolve this, a timer will be set before a *micro grid* needs to register to a different *macro grid*. This will not have much impact, since the *micro grids* will not require registration to a *macro grid* most of the time. In addition, traffic or emergency issues will be broadcasted to adjacent *macro grids* to warn about incoming traffic and to ensure a *micro grid* registered in an unexpected cell is warned.

From the system's point of view, a *macro grid* is a logical unit that facilitates none, one or more *micro grids*. Each *macro grid* will have a *macro grid manager*. The *macro grid manager* will receive local traffic trouble reports from the *micro grids* within the same *macro grid* and will make decision to broadcast. It will also be responsible for generating heartbeat messages at a regular interval. Whenever possible, there will be one or two *micro grids* with elevated responsibilities, called super node (*Su*) and backup super node (*BuSu*) that will take the *macro grid manager* responsibilities, otherwise, the *core* will

take the management responsibilities. *Su* and *BuSu* will not only offload the *core*, but also give local control to the *macro grids*. This way, if there is any communication failure, the *macro grid* can remain informed of local situations via alternate methods.

Each *macro grid* will be responsible for storing the map of its own area. Copies of local maps will be stored either by the *Su* node and/or *BuSu* node, or distributed across other *micro grids* in the *macro grid*. Regardless of the shape of the *macro grid* coverage area, the map stored in each *macro grid* will be the closest rectangle that will cover the entire *macro grid*. Copies of these maps will also be stored in the *core*. The *core* will keep track of the area covered by each of the *macro grids*.

Micro Grid / Node Definition

A *micro grid* will be the smallest logical entity of the entire system. Each *micro grid* can be a single device or a combination of devices. In this thesis, a device can be a sensor, a phone, a smart phone, a laptop, a Global Positioning System (GPS), a car's on board computer or diagnostic (OBD) system, or any other module capable of providing a service and communicating with other devices. Some examples of combined devices in a *micro grid* are: a phone, or a smart phone, or a combination of a smart phone and a GPS,, or a GPS and a computer, a traffic, road or weather sensor, or many other combinations. Not all *micro grids* are mobile (i.e. any permanent sensor mounted on the road, a traffic post, or on a support structure will not be mobile). Figure 3.4 shows examples of a variety of *micro grids* residing in a *macro grid*.

All the devices or modules in a *micro grid* will collaborate with each other and act as a single unit in the eyes of the *macro grid*. The scope of a device in a *micro grid* will be

within a vehicle or anything that can be confined as a single entity (i.e. a box, a sensor mount, laptop with Data Stick, or a lamppost). The devices within the *micro grid* are assumed to be reasonably stationary relative to each other and to be available to one another while joined in the grid. It is assumed that intra *micro grid* communication is done mostly over Bluetooth, but it could also be done through ZigBee or WiFi. It can even be a direct or wired connection such as a USB data stick connected to a laptop, or a phone connected to a laptop using tethered cable.

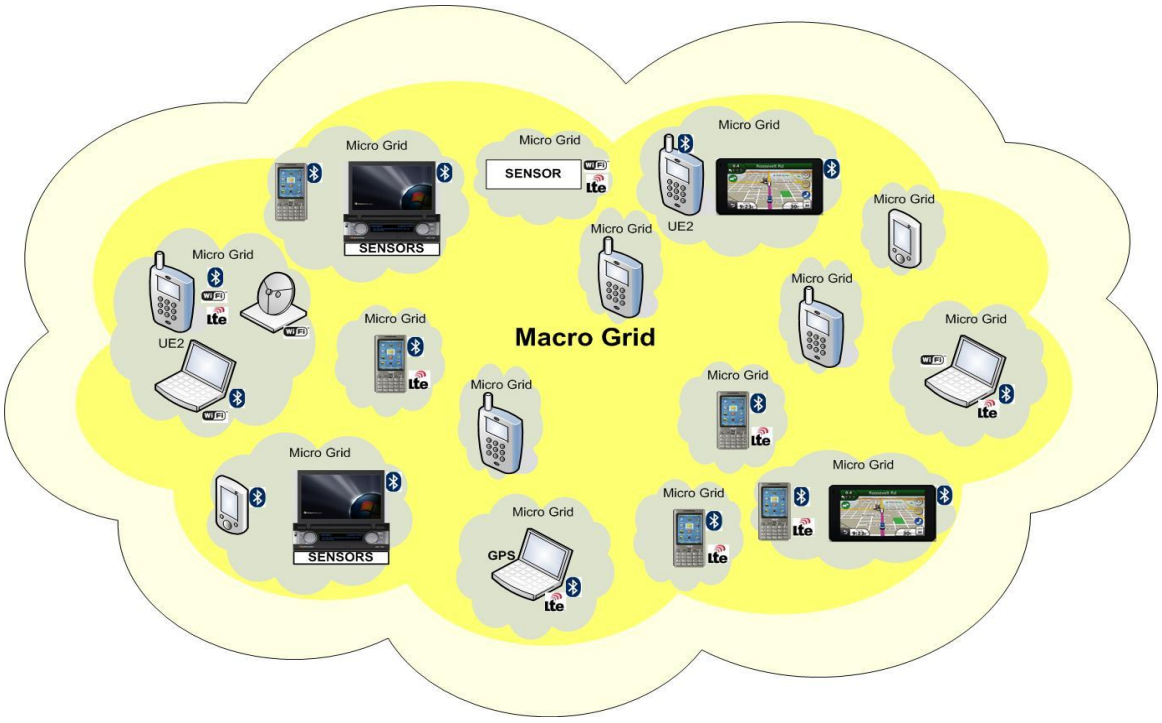

Figure 3.4 Typical Macro gird with Micro grid members

One of the most desired attributes of each *micro grid* is the capability to receive cell broadcast messages and be registered in a cellular network. Although cell broadcast capability is required and desired, a device may still join or participate in the grid via other communication methods. Therefore, these *micro grids* with no cellular module will have to depend on other *micro grids* in the *macro grid* to relay these messages.

For certain inter *micro grid* connections (i.e. permanent sensor deployment), resource sharing, and standalone operations the *micro grids* will use ad-hoc mesh networking using unlicensed wireless technologies, including vehicle-to-vehicle communication techniques. It will use the model described in [43], [44]. Detailed information on these is beyond the scope of this thesis.

Each *micro grid* in a *macro grid* will either provide one or more services or just be a user of services. The main services provided for q-end problems are routing, managing communication, and sensing. Routing service include passing information to a different node. This will be applicable when forming an ad-hoc mesh network. Managing communication include maintaining addresses of neighbors, knowledge of available resources etc. In normal conditions it mainly applies to *macro grid managers*.

Sensing will include speed and congestion detection, and/or the identification of vehicle presence, the quantity of vehicles, road condition detection (i.e. slippery, wet, slushy, icy etc.), weather condition detection (e.g. rain, snow, visibility) and so on. The sensing capability of a *micro grid* will not be restricted to road mounted sensors (which could be a *micro grid* as well), but also through carputers or OBD, smart phones and other handheld devices. However, smart phones will be the primary devices considered in this thesis. Many cars are equipped with traction sensors, snow and ice sensors, temperature sensors as well as speed, sudden stop sensor, location (GPS) and direction sensors. The carputer or OBD can both provide traffic information (via cable, or Bluetooth, WiFi or Zigbee modules) to a smart phone (as described in [15]) or to another smart device that is linked to the *macro grid*, perhaps through vehicle's own communication system.

Micro Grid Classification

Depending on resource availability, reliability, and subscription *micro grids* are divided into three classes: (a) User Only (UO), (b) Service Only (Sr), and (c) Service and User (SrU). *Micro grids* with *UO* class are the users of warning related services, *Sr* class is the provider of one or more services, and *SrU* class are both users and providers of services to/from the grid. There are three types of profiles: "Un-Trusted", "Trusted", and "Subscribed". There are also three types of roles: "UserNode" (Ur), "SuperNode" (Su), and "Backup SuperNode" (BuSu). Some of these classes can only have one permanent profile and role and some can change their profile and role by changing their subscription. In most cases, the *Su* will take the *macro grid manager* role.

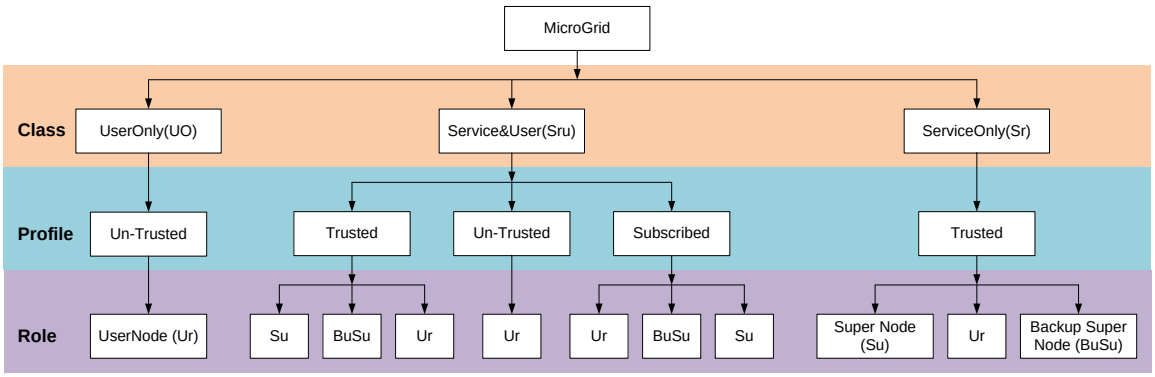

Figure 3.5 Micro grid classification, profile, and roles

A graphical view of *micro grid* classification, profile and role is provided in Figure 3.5. Description of these classifications, their potential profiles and roles are provided in details in following few paragraphs:

Any *micro grid* can take the *Ur* role, but participation in congestion detection depends on available resources. Typically *micro grids* assuming *Su* or *BuSu* role will have additional resources. For example, these elevated *micro grids* may have a higher CPU power,

capability to serve as WiFi/Zigbee access point, and/or have connections to alternate communication systems (i.e. satellite, wired internet connection/digital subscriber lines) in addition to its cellular module that is capable of data and SMS communication.

User Only (UO) *micro grids* are the users of the services and they do not contribute to the grid at all. Typically this would be UEs with limited resources. They will only be able to receive cell broadcast messages from the grid. These *micro grids* can only have the profile of an *un-trusted* user and role of *Ur*. An example of these devices would be a basic phone or data stick with no smart built into them.

Service only (Sr) *micro grids* will only provide services and will not use any information for their own benefit. These *micro grids* will always be considered trusted and reliable (i.e. will have a *trusted* profile). To ensure security, these *micro grids* will have to be provisioned manually into the *core.* These *micro grids* will be able to take the role of *Ur*, *Su*, or *BuSu* while providing the service they are assigned for. These *micro grids* will have a higher priority to become a *Su* or *BuSu*, provided they have the required resources. Typically, these *micro grids* would be fixed mounted sensors, a wireless hub or an access point, an internet gateway or any other manageable *micro grids* that has something to offer. Information coming from these *micro grids* will have the highest weights in speed calculation or any other warning message generation.

Service and User (SrU) *micro grids* will have some or all the characteristics of *User Only* and *Sensor Only micro grids*. The *SrU* can have a profile of *trusted*, *un-trusted* or *subscribed* and a role of *Ur*, *Su*, or *BuSu*. Their contribution to the grid will depend on their profile and the services they have to provide.

A *SrU* with a *trusted* profile will have the most to offer to the grid and will be considered most reliable. It will be considered a good candidate for *Su* or *BuSu* role. An example of a *trusted SrU* will be the *micro grids* residing in emergency vehicles (i.e. police, ambulance or fire). It is assumed that emergency vehicles will be equipped with multiple reliable sensors (speed, weather, road condition monitors etc.), multiple licensed (cellular and satellite) and unlicensed (i,e, Bluetooth, WiFi, ZigBee) wireless voice and data services. Their alternate communication methods will allow them to communicate with the CBS and others in the *macro grid* in case of the cellular network failure. Information coming from these *micro grids* will also have the highest weights in speed calculation or any other warning message generation. These *micro grids* can also be an intermediate device (like routers in a ZigBee) to pass messages to the other *micro grids*. Note, the *micro grids* inside the emergency vehicles will typically be launched with *user* role and remain as such, unless there is an emergency or the *Su* is absent in a *macro grid*.

A *SrU* with a *subscribed* profile will also be able contribute to the grid. However, the information coming from these *micro grids* will have a lower weight in speed calculation or other warning message generation than the *trusted micro grids*. An example of these would be any UE that has bought a subscription to services other than basic network access. Often, their service will be verified by another means (i.e. against stored track record) by the *core* or *macro grid manager*. At the time of *Su* or *BuSu* selection, these could be considered only if there is no other *trusted micro grid* present in the *macro grid*. The *core* may flag a *subscribed micro grid* as unreliable based on its past performance, in which case it will be considered as *un-trusted*. The primary member of this type of *micro grids* will be UE (a smart/non-smart phone or data stick) with internal or external (i.e.

GPS, a carputer, laptop, OBD, or other devices) resources that enable it to provide services. These types of *SrU*s will be eligible to get other services as well (i.e. point of interest, more customized information etc). These *micro grids* may also take the task of routing.

SrU with an *un-trusted* profile will also be able to contribute to the grid, but their contributions will have the lowest weight. They will be the last choice to be elected or selected as a *Su* or *BuSu*. Information from these *micro grids* will be rejected if there is enough *trusted* or *subscribed micro grids* providing a similar service. However, if they provide a unique service, it may (if they are necessary) not be rejected. The device type for this profile will be same as a *subscribed SrU*. The main difference between the *subscribed* and *un-trusted* profile is: an *un-trusted* profile holder will not have a subscription to any special services (i.e. point-of-interest, content downloads etc.) provided by the system. Any *micro grid* with data access can still get different services from the internet, but not through the grid.

Regardless of classification, profile, or role all *micro grids* qualify to receive traffic information and emergency warnings. The user class and profile will be set at the time of software installation. Part of the installation process will require an authentication that will set the class and profile, which will be stored in the non-volatile memory of the primary device of the *micro grid*. If the authentication fails, the default class of *Ur* and *un-trusted* profile will be assigned. The profile may still be authenticated during *macro grid* registration and service requests. If there is a change in the profile (i.e. result of subscription change) the highest potential profile will be changed via an Over the Air (OTA) message (i.e. a special SMS message).

When the application is launched each *micro grid* will have the default role of *Ur*. This role will change if there is no *Su* or *BuSu micro grid* in its current *macro grid*. The role of *micro grid* could change on demand depending on the situation and resources available in the *macro grid.* Typically only a *trusted micro grid* can take the role of a *Su* or *BuSu*.

3.4 Queue-end Scenarios and Corresponding Sensor Implementation

Based on predictability, risk factors, and frequency of occurrence, there are multiple types of obstructions that can cause q-end problems. In this thesis the q-end problems are categorized into following five scenarios: (a) predictable non-mobile permanent obstruction, (b) predictable non-mobile temporary obstruction, (c) predictable time sensitive obstruction, (d) somewhat predictable mobile obstruction, and (e) unpredictable unknown obstruction.

The proposed middleware will work in all of these scenarios. Although the deployment of traffic sensors is recommended for certain scenarios in this thesis, the use of smart phones will still work in all of these cases. It is more justifiable to have pre-deployed sensors in some situations. The only difference would be the type of detection device, presence and volume of trusted *micro grids*, and the number of communication choices. Regardless of the sensor choice, they will all serve the same purpose, which is congestion detection and notification. Diverse sensors will have different methods of identifying congestion, but they will all use the same application programming interface (API) for inter and intra grid communication. The downlink communication to all *micro grids* in a *macro grid* is primarily taken care of by the SMS-CB.

Predictable Non-mobile Permanent Obstruction

Toll booths, border crossings, and ferry crossings are considered to be predictable non-mobile permanent obstructions. In this thesis, the risk of q-end problems due to this type of obstructions is considered low because predictability is very high. As a result, there are many precautions already taken for this type of scenario. For example, there are many road signs (fixed as well as variable) that warn people of upcoming toll booths. Also there are speed limit changes and extra lanes added close to these obstructions. In this thesis, these types of scenarios are considered the prime candidates for deploying fixed and permanent sensors.

If permanent sensors are deployed, they all will have a *Sr* classification and *trusted* profile. In every *macro grid* containing permanent sensors, at least two of the sensors will have additional resources. These elevated *micro grids* will take the *Su* or *BuSu* role and will have the proper authorization at the CBE to issue cell broadcast messages. If any of the elevated *micro grids* fail, the other permanent sensor in the same *macro grid* will have the potential to take the *Su* or *BuSu* role. An emergency vehicle equipped with similar resources will have the power to override and become a *Su micro grid* in case of an emergency.

This thesis recommends that sensors should be deployed every 250 meters for this type of scenarios. It is assumed each of these *micro grids* will have Wi-Fi or ZigBee capability. Whenever possible, these sensors will use either direct connection or unlicensed wireless technology for intra *micro grid* communication to offload the cellular network and to increase reliability. The 250 meter distance will allow intra *micro grid* communication via Wi-Fi or Zigbee even if one of the *micro grids* is out of commission. This distance

also ensures adequate stop time is provided to upcoming traffic travelling at 120 kilometers per hour in any weather condition. Figure 3.6 shows a layout of how the sensors can be deployed.

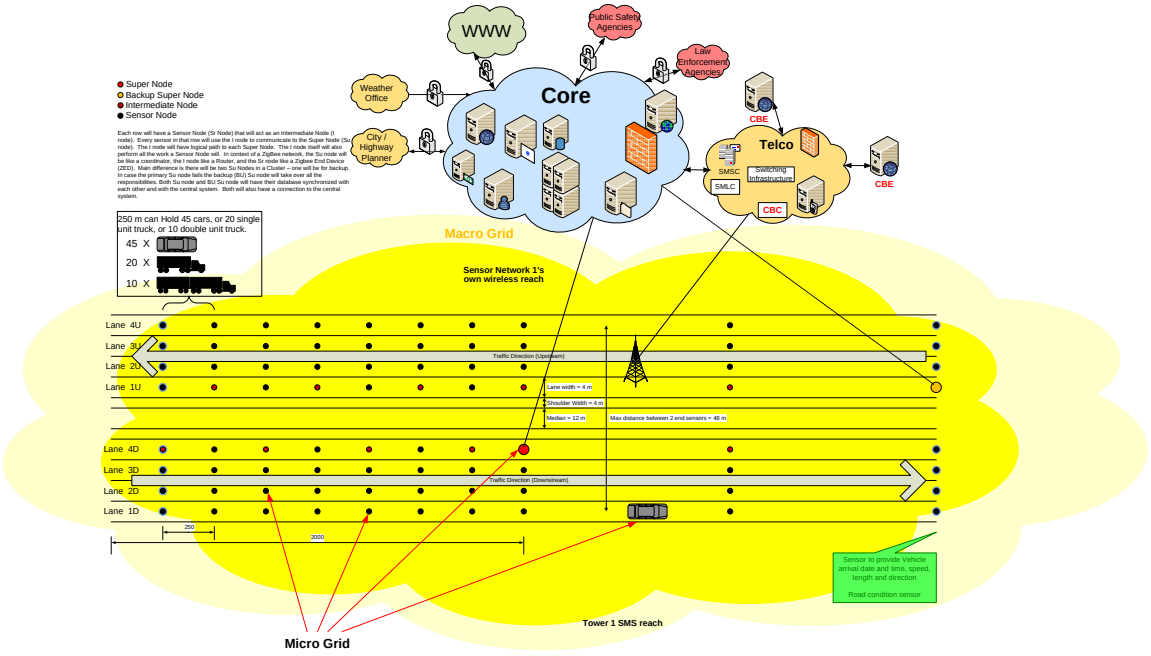

Figure 3.6 Recommended layout for fixed sensor deployment

The congestion detection algorithm used by a fixed installed *micro grid* (i.e. a traffic sensor) could be different than the ones used by a smart phone. A smart phone will detect its own speed and location, whereas a traffic sensor may use its counting, classification (i.e. size), speed and presence detecting ability along with artificial intelligence to find q-end location and the speed at the q-end. There will also be more interaction among the fixed *micro grids* than the ones in a *macro grid* consisting of mobile devices. However, these *micro grids* will still use same API to communicate to the *macro grid* and rest of the *micro grids*. Therefore, any mobile *micro grid* roaming in the roads in a *macro grid* will still contribute to the grid and will extend the reach beyond the location of the last fixed sensor deployed.

Predictable Non-mobile Temporary Obstruction

Construction zones or temporary street events are considered to be a predictable non-mobile temporary obstruction. Most construction work or street events are scheduled and planned ahead of time. Therefore, there are ample opportunities to put temporary warning signs as well as special sensors to warn and predict q-end locations. Hence, this situation is regarded as low risk. This thesis recommends that there be at least two temporary sensors deployed with elevated resources and authorization. These temporary sensors will be classified as *Sr micro grids* with *trusted* profile and will take the role of *Su* or *BuSu*. They will provide updated traffic information to the CBE.

The algorithm used by these two sensors will be different than the one mentioned above or the ones used by mobile *micro grids*. The algorithm used by these two sensors would be the one recommended by [32]. Although these two sensors will take the *Su* and *BuSu* roles they will still collaborate with other mobile *micro grids* that will use the section of the roads where they are deployed, meaning that these sensors will broadcast the q-end location based on their visibility and assessment, and the receiving *micro grids* will correct them, if necessary, in the same way as any other implementation of this middleware. This will fill in some of the gaps observed by the work done in [32], as the two sensors will assure that warning is provided about upcoming construction. They will also facilitate alternate communication (i.e. act as Wi-Fi access point) methods in case the cellular network fails.

Predictable Time Sensitive Obstruction

High volume traffic due to the time of day (i.e. rush hour), seasonal (i.e. holiday season), and special events (concerts, game, exhibitions etc.) are considered predictable time

sensitive obstructions and are somewhat predictable for most common users and traffic control agencies. Although some uncertainties exists (especially for visitors or residents that are unaware of the event), these cases are thought of as moderate risk.

Many downtown areas have high rise buildings that face high traffic during rush hours. The buildings may interfere with certain GPS and other wireless based location determination techniques. For these scenarios this thesis recommends that few fixed sensors be deployed in parts of downtown and high traffic areas. However, the density of permanent sensors could be further than 250 meters apart. It is assumed that many of these critical roads are already equipped with fixed sensors, warning systems (i.e. dynamic message signs), and additional police patrols. Other mobile *micro grids* will still enhance and extend the reach of these fixed sensor deployment by participating in the grid. Functionality of this type of deployment will be similar to first scenario mentioned in this section.

Somewhat Predictable Mobile Obstruction

Snow ploughs and extra wide or large loads are deemed as a somewhat predictable mobile obstruction. The reason these are categorized as "somewhat predictable" is these vehicles provide many warnings, in the form of special signal lights, escort vehicles, and road signs. Some even make their location information available to the drivers in real time. In this thesis, it is assumed that these vehicles will be equipped with special device(s) to form an elevated *micro grid* that will take *Su* role, provide reliable sensing capabilities, have more than one communication methods, and be able to deliver traffic updates to the CBE for broadcast. This will be useful when the visibility is poor, as these vehicles often exacerbate the problem. In addition, the *micro grids* in the surrounding

area will collaborate in relaying and determining the actual q-end length. The algorithm used here will be similar to the unpredicted unknown obstruction scenario and/or full mobile scenario, the only exception being that the presence of an elevated *micro grid* is guaranteed. Therefore, in the case of standalone mode of operation the *micro grids* in the area will still be able to receive and participate in traffic and other warnings.

Unpredictable Unknown Obstruction

Any type of sudden accident (i.e. vehicular, industrial etc.), disaster (i.e. fire, earthquake, tsunami, earth slide, avalanche, poor weather condition, terrorist attack etc), or other emergency that significantly interrupts traffic falls under the "unpredictable unknown obstruction" category. This type of scenario is completely unpredictable and posses the maximum risk. There is no way to predict where and when an accident or disaster will take place, thus this type of scenario cannot be pre-warned unless all roads are equipped with sensors, cameras, and/or monitored by trained human observers, which is impossible. These scenarios can cause congestion and standstill traffic, where traffic is expected to run at a high speed. This will, in turn, cause severe q-end problems. It can happen in a location invisible to upcoming traffic and be aggravated by poor weather conditions. This scenario is the prime candidate for rear end collisions. This type of scenario typically lasts for a relatively short time (few minutes to few hours), but requires on-demand, real-time, and immediate notification. Providing warning about this problem is the main focus of this thesis. This thesis recommends the use of smart phones or other mobile devices with connectivity to the cellular network and preferably with alternate wireless modules for the detection of unpredicted q-end in order to notify upcoming vehicles.

Depending on the severity of the incident, this type of situation clogs up both the transportation infrastructure, and the telecommunication network, especially if it is caused by a terrorist attack or a major disaster. The grid will detect q-end problems and warn people before complete telecommunication breakdown occurs due to people overreacting and seeking information. In cases like this, it is assumed that emergency vehicles such as police, fire trucks, and ambulances will be present. These vehicles will be equipped with *trusted* pre authorized *micro grids* that have alternate communication methods (i.e. satellite) to maintain a constant communication with the *core*, which then initiates the SMS-CB message based on location.

In case of major accident or severe congestion without any other catastrophe, there will not be any significant impact on the cellular network. All mobile *micro grids* will collaborate to allocate the q-end using the mobile network.

In cases when the tower or its backhaul network is damaged, the *macro grid* will operate in standalone mode using alternate wireless methods. The *micro grids* in emergency vehicles may also access the *core* through their private or alternate networks. The *micro grid* in the emergency vehicle will collaborate with other emergency vehicles or other *micro grids* to update traffic and queue end information via ad-hoc mesh network, which could be facilitated by these emergency vehicles.

3.5 Services Provided by the Solution

The system is designed to provide different services to different types of users. The primary service provided by the system is congestion and q-end location detection, and notification of any emergency messages leading to people's safety mostly using SMS-CB.

Based on sensor availability, it can also detect other threats. The warnings include information on emergency situations, q-end location, or step by step travel instructions to a safe location.

A *micro grid* with *UO* or *SrU* classification (i.e. a driver's phone) will get notifications of road conditions (i.e. queue end, congestion, traffic information), emergency information issued by government agencies, speed warnings, and routes to safe location on their device screen or in text to voice format. In case of road obstructions or congestion on the driver's route, it will inform the calculated distance of how far the obstruction is. In addition, subscribed users will be able to get map services (i.e. directions, points of interest, wait times) or download contents (i.e. firmware, videos, or songs).

Any time there is a road condition or an emergency, the *macro grid manager* will send the message to be broadcasted to the effected *micro grids* as well as to the *core* for storage and report generation. In cases where the connection to the *core* is unavailable, the *macro grid manager* will store the information locally until the connection is backed up. The *core* will run reports at regular intervals and dispatch them to different agencies. All reports will be dispatched automatically via machine-to-machine interface.

Depending on the algorithm used, and the number and type of *trusted micro girds* (i.e. speed sensors, counters, cameras) deployed the *core* will give different statistics. For example, it will send road statistics to federal, provincial, municipal or independent transportation planners and to law enforcement agencies. If a road constantly becomes congested or faces lots of accidents, it will notify the planner to react on them (i.e. change traffic signal parameters, add road capacity etc.). It will send sensor health information to

operation group, so they can send the proper maintenance crew. Law enforcement agencies will receive traffic violators' and stolen vehicle information from the system. A speed sensor will trigger a camera (if available) or just send traffic violation information to a law enforcement agency, which can be used to plan for permanent or temporary traffic enforcement tools or dispatch of police officers.

CHAPTER 4 SOFTWARE (MIDDLEWARE) ARCHITECTURE

This section introduces the middleware design, followed by a brief description of different components of the middleware. It contains XML definitions to present attributes of different components. The chapter concludes with the class definitions, UML drawings of different processes and multiple congestion detection algorithms. All the grid members (*micro grids* with different roles and the core) will have the same modules installed in them. However, the modules will inherit different attributes depending on where they are installed.

4.1 Overview of Software Architecture

The control and intelligence of the system is provided by the software component, which is divided into three layers: Physical layer, Middleware, and Application layer. A pictorial view of the software architecture is provided in Figure 4.1 Overview of software architecture and middleware. The physical (lower) layer is for communicating with the hardware. It consists of firmware and plug and play technologies. Details of these are beyond the scope of this thesis.

The middleware is the middle layer of the software component. It is the bridge between the application layer and the physical layer, and hides all the lower layer complexities from the application layer. The application layer communicates with the user. It processes user requests and provides the feedback.

The middleware is responsible for all inter and intra *micro grid* communication. It handles all *macro grid* level assignments. It deals with different modes of operation (i.e. standalone mode and centralized mode). It sets up and maintains all available resources

and services at the time of initialization and during the uptime of the application. It is also responsible for data collection from different internal and external modules. For example, it reads the sensor outputs (i.e. GPS data from GPS sensor, temperature from a temperature sensor etc.) and stores them in the current database, which is accessed by the application layer. The GPS information may come from a device's internal module, an external GPS device or a carputer. However, the actual source will be transparent to the application layer.

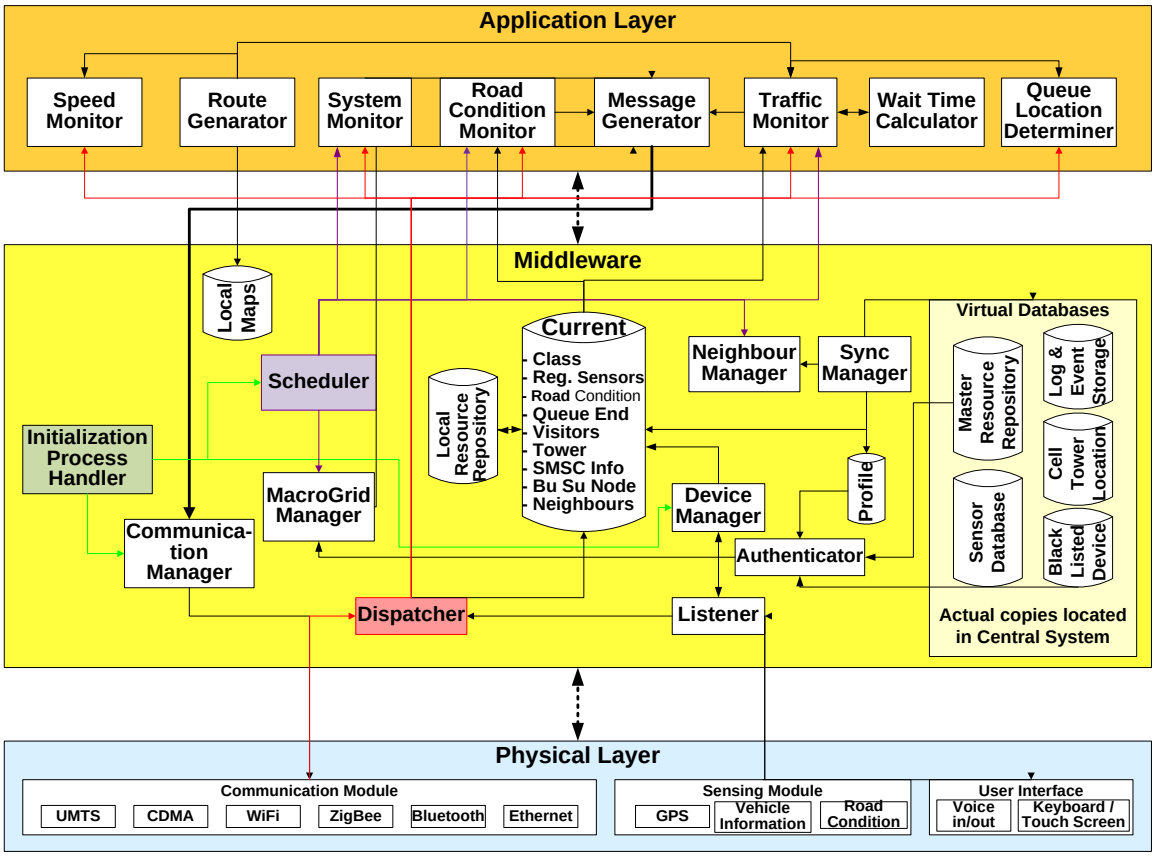

Figure 4.1 Overview of software architecture and middleware

The middleware is also responsible for the path or method to be used for sending and receiving any data. For example, if there is a secure Wi-Fi connection available for data transmission it will use Wi-Fi instead of cellular network. It will hide the *macro grid*

manager information from the application layer, meaning that the application layer will not know if it is sending the data to a *Su micro grid* or the *core*. Similarly it will not know from whom the data is coming from. The middleware will perform all authentications, as well as registration and deregistration to a *macro grid* as necessary.

Not all components of the middleware or application layer modules will be active for all *micro grids* at all times. The middleware will activate or deactivate software and hardware resources in and out as necessary. For example, a *micro grid* with *UO* profile will not need to have its *neighbor manager* active when operating in normal conditions. It will only be active when the *macro grid* is in standalone operational mode. On the other hand, a *trusted micro gird* on a fixed permanent deployment will communicate with other *trusted micro grid* on the same *macro grid* using *neighbor manager* module.

The middleware will also emulate as if it has all the resources as the *core*. If an application requires any information that is not in its local database it will obtain the information from the *core* and deliver it to the application layer while hiding the complexity of acquiring the information. Authentication is an example of this type of scenario. The local *macro gird managers* do not store profiles of all *micro grids*, but when a *trusted micro grid* attempts to join, if it does not have the profile, it contacts the *core* to get the authentication information.

4.2 Details of Software Components

This section provides a description of different components of the middleware.

Middleware Components

The **Device Manager** is responsible for discovering and managing all internal and external resources. These resources can be both the hardware and the software. Internal hardware resources are referred to as the device's own built-in modules (e.g. GPS, Bluetooth, Wi-Fi, keypad etc.). A few example of internal software detection includes, but is not limited to, presence of voice recognition and playback software, SMS and/or cell broadcast send or receive APIs, device's location based services APIs, and speed detection APIs.

External resources are the devices or modules within the *micro grid*. A vehicle's OBD or carputer, a personal computer or PDA, individual GPS system, a temperature or humidity sensor, and wireless access point are all examples of external resources. These devices can be connected wired or wirelessly. The method used to discover and manage these devices is beyond the scope of this thesis. It is assumed that other applications in the middleware will have APIs to access services discovered by the *device manager*.

The **Listener** works with the *device manager*. It constantly reads the output of the services discovered by the *device manager*. It then stores the results in the *current database*, making the information available to all of the other applications in the middleware. For example, the GPS sensor determines the current location and stores the information in its own buffer. The *listener* extracts the location information from the GPS module and saves it in the *current database* every second. It extracts any new cell broadcast message available to the cell broadcast application as soon as it is available. Some of the other items it monitors includes: key press, voice command detection, and

movement from one cell to another (that is change of cell ID), sudden stop sensor (if available through OBD). These items are sent to the *dispatcher* for immediate action.

In case of the *core* the *listener* will listen to upcoming traffic updates/enquiries coming from different *macro grids*, other service providers (i.e. weather stations), requests coming from different agencies, provisioning system, and web servers.

As the name suggests, the **Sync Manager** is responsible for synchronizing different information and time. An instance of *sync manager* will run in most *micro grids* as well as in the *core*. Depending on the role of the *micro grid*, the *sync manager* inherits different attributes. In the case of *Su* and *BuSu micro grid*, it will synchronize *macro grid* information with each other and with the *core*. For example, *sync manager* will update *macro grid* member information, speed at different points in the *macro grid* coverage area, and list of available services. The *sync manager* will store *micro grid* member information only if permanent sensors are deployed or when in standalone mode. However, each *Su* and *BuSu* will share and sync the list of *trusted micro grids* in the *macro grid*.

Any time there is a change in a *micro grid's* profile (i.e. subscription/un-subscription to a service), or role, the *core* is responsible for relaying that information to the *micro grid* itself. If the *micro grid* changed its role to or from a *Su* or *BuSu*, then the *core* will update the information to the *micro grid's* neighbors as well. The *sync manager* will also be accountable for syncing the event logs (i.e. failure, intrusion reports) with the core or backup system and harmonizing the traffic violation and security logs with law enforcement agencies. It will also send updated traffic information to a web server.

The **Communication Manager** is responsible for hiding the communication complexity from the application layer. It manages different communication methods and routes between different *micro grids* in the *macro grid*. For example, two *micro grids* may communicate via SMS, the internet protocol, or through the *core*. It may also be able to communicate via a local Wi-Fi access point. It records all methods of communication in its routing table. However, it puts different costs for different routes. A route via direct hardware connection (i.e. cable) is given the least cost, followed by using Bluetooth, ZigBee, Wi-Fi, and mobile networks. A route using the cellular network is considered the most expensive. The actual implementation of routing handover is beyond the scope of this thesis. In cases where there will be intra *micro grid* communication using a non-cellular wireless media, the *communication manager* will still maintain the responsibility.

The system is designed so that there will rarely be any network traffic in normal conditions. Even at the time of regular road traffic congestion detection, the telecommunication network traffic volume will be small (i.e. 82 bytes or 150 characters maximum payload). Therefore, unless there is cellular network congestion the traffic information will be sent using cellular network.

The **dispatcher** analyzes the incoming data and processes them according to the message header. For example, if it receives a traffic related message, it sends it to the traffic monitor. If it receives new *heartbeat* messages, the *dispatcher* will update the c*urrent database* with the current contact information. If it receives a join (register) or leave request it will initiate the registration/deregistration process. Note join and register does not apply to every operating mode.

The **authenticator** will perform all authentications in the system including query to the blacklisted database for stolen equipment. It will first look at its local database, and if the authentication information is locally unavailable it will then attempt a query to the *core*.

The **sensor database** will consist of list of devices and *micro grids* available in the network. It will contain the sensor name, location, IP Address, MAC Address, communication method, type, model, serial numbers, features, installation date, street name installed, and permanent/temporary notes. This only applies to the *core* or *Su* of a fixed sensor deployment.

The **Current Database** stores the information required for day-to-day operation that has a limited shelf life. For example, it stores current speed, location, hop, and *macro grid* information, which changes either every second or when a *micro grid* moves from one hop to another hop or from one *macro grid* to another. The **Neighbour Database** is a subset of *current database*. It stores different contact information of neighbours.

The **Local Resource Repository** is used by each *micro grid* to store local resources like communication methods, GPS sensors, maps of the *macro grid*, and route information. It also stores any shareable resources, such as songs, firmware, and other content. A *Su* will also store a list of services other *micro grids* in the *macro grid* have to offer (i.e. temp, humidity, and rain sensors etc.).

The *core* uses the **Master Resource Repository** to track each *macro grids'* physical scope. This is used when generating the *heartbeat* messages. Information in this database includes: *macro grid* ID, location, number of cells, cell IDs, and map overage

area. For each active *macro grid* the core creates an instance, so it can track different attributes.

The **Neighbour Manager** is responsible for neighbour management. The *Su* and *BuSu* maintain neighbour relationship each other as well as with the *core*. The *Ur* initiates the neighbour relationship with the *macro grid manager* only when there is congestion or failure in the telecommunication network. The *neighbour manager* is responsible for sending and receiving *heartbeat* messages at regular intervals in order to maintain neighbour relationships. If a *heartbeat* is not received three times in a row, then it will consider the neighbour to be unavailable.

The **Point-of-Interest Database** stores a list of the points-of-interest locations and their attributes. The *core* keeps a list of all points-of-interests, whereas the *Su* only stores the portion of this database that is relevant to its location. This database stores lists of hospitals, police stations, services (restaurant, gas station), tourist attractions and their corresponding location, contact information and hours of operation.

The **Map Database** stores the map segments of the area covered by a *macro grid* and its surrounding area. Each road or road section is stored as a *hop*. Therefore, this database inherits all the attributes of a *hop*. In case of *core* the map database contains maps for each *macro grids*.

Application Layer

The **Traffic Monitor** performs several important functions and calls several other modules (i.e. **Wait Time Calculator** and **Queue Location Determiner**) to complete its tasks. The main tasks include: speed and direction calculation or retrieval, congestion

detection, traffic violation detection and warning, and estimation of travel time. All of these functions are triggered events. They are either triggered by the s*cheduler* or by other events (i.e. entry/exit to/from a cell or *macro grid,* or receiving of a *congestion detection* message). *Traffic monitor* reads the required information (i.e. route plans) from the *current database*. For the case of fixed sensor it is only triggered by the *scheduler*.

Regardless of network conditions or of *macro grid* operating mode, each capable *micro grid* performs speed and direction calculations, or retrieval, every second. For *micro grids* that already have a built in speed and direction calculation application, this module reads the speed and direction from that application and stores it in the c*urrent database*. For *micro grids* that have access to only the GPS locations, but not the speed or direction, the application will still retrieve the location information every second. The *traffic monitor* will then calculate the speed and direction of the *micro grid* using the time interval and two points, one point retrieved in current sampling time and the one stored in a previous sampling time. It will then store all three values in the *current database*. The *current database* will store the traffic information in a rolling buffer (i.e. delete oldest data, when buffer is full).

In addition to speed and direction retrieval or calculation, the *traffic monitor* will compare the current speed with the speed limit for a particular section of the road. If the current speed of the mobile *micro grid* is less than a threshold value it will initiate a congestion detection process. The congestion detection process is explained in a later part of this chapter. Once the congestion detection process is complete and confirmed, it will flag the route as congested and send a *congestion detected* message to the *macro grid manager*. Upon receiving the *congestion detection* message, the *macro grid manager*

will verify the sender's profile to properly weigh the information it just received. Based on the profile of the *congestion detected* message sender, the *macro grid manager* will either raise a *potential congestion* flag or a c*ongestion alert* flag. If the *congestion detected* message comes from a *subscribed* or *Un-Trusted SrU* a *potential congestion* flag will be raised and it will start a timer (equivalent to 2 complete *hop* delay time). If *macro grid manager* receives a similar *congestion detected* message from at least five other *subscribed* or *un-trusted SrU* or at least one *trusted micro grid*, all located in the same hop before the expiration of this timer, it will change the flag from *potential congestion* to *congestion alert*. Otherwise it will reset the flag to normal. This will reduce false alarms (i.e. malfunctioning *micro grid*, someone stopping for a non-traffic related reasons, or intruders to the system). If the congestion message comes from *trusted micro grid* then *congestion alert* flag will be raised right away. Once the *congestion alert* flag is raised, the location will be marked as congested. This will trigger *congestion alert* messages to be broadcasted via cell broadcast.

If a mobile *micro grid* travels at a speed greater than a predefined percentage (say 10%) of the speed limit for 5 consecutive sample times, it will generate a warning to the driver via screen display or voice signal if the *micro grid* class is *UO* or *SrU*. If the *micro grid* is of *Sr* class, then it will first trigger a traffic camera if available then send the picture and speed information to a law enforcement agency. If there is no traffic camera in the proximity it will just send the traffic information to a law enforcement agency.

Each time a *trusted micro grid* enters or exits a *macro grid* it will send its speed and location to both the old and new *macro grid managers*, which give the *macro grid managers* a sense of current running traffic conditions even when there is no reported

issue. The fact that only *trusted micro grids* perform this task will reduce the cellular traffic load and the computing load of the *macro grid managers*, reassuring that there is no unreported issue due to lack of participating *micro grids*.

The **Wait Time Calculator** calculates how much time a *micro grid* will require to remain in a hop, a *macro grid* coverage area, or a route. These times are used by certain congestion detection algorithms. The **Queue End Determiner** is responsible for comparing the current location with reported troubled locations to ensure q-end location is not actually further than what is reported.

The **Speed Monitor** is used by the *macro grid manager* to process traffic speed reported by *trusted micro grids* at the entry and exit points of the *macro grid* coverage area. The *macro grid managers* use this to determine the average running speed at different entry and exit points of the *macro grid*. This information is used by the *route generator* when creating route plans. The **Road Condition Monitor** interprets sensor outputs of different road condition sensors (i.e. slippery, wet, rainy, snowy etc.) or carputers and initiates proper warning message generation for the given road condition.

The **System Monitor** is responsible checking the health of different components of the *micro grid* (i.e. sensors, devices, communication links) and the *core* (i.e. different servers and communication links). It logs and reports any failures or problems to the appropriate maintenance authority or simply displays them on the *micro grid's* display.

The **Route Generator** is responsible for creating optimal routing information for users. The **Message Generator** dynamically generates different messages in CAPv1.2 format.

4.3 Description of Objects

This section provides description of different objects, their relationships with other objects, and corresponding XML diagrams describing these relationships.

Relationship between *Micro Grid* and *Macro Grid*

The *macro gird* inherits all of the attributes of *micro grids* and are made up of one or more *micro grids*. This is demonstrated on Figure 4.2. Note, since *Su* and *BuSu* are also *micro grids* they are listed under members, but have special flag to identify them.

```
<MacroGrid>
      <MacroGridId> <MacroGridId>
      <PrivateAuthKey> <PrivateAuthKey>
      <PublicAuthKey> <PublicAuthKey>
      <AreaCovered>
                  <GPSPoint1> <GPSPoint1>
                  <GPSPoint2> <GPSPoint2>
                  <GPSPoint3> <GPSPoint3>
      <AreaCovered>
      <Members>
            <MicroGridSu> <MicroGridSu>
            <MicroGridBuSu> <MicroGridBuSu>
            <MicroGrid1> <MicroGrid1>
            <MicroGrid2> <MicroGrid2>
            ...
            <MicroGridN> <MicroGridN>
      <Members>
      <ServiceList>
            <Service1> <Service1>
            <Service2> <Service2>
            ...
            <ServiceN> <ServiceN>
      <ServiceList>
<MacroGrid>
```

Figure 4.2 XML for macro grid attributes

Relationship between a Device, its Profile, and *Micro grid*

The attributes of a device includes its name, address, communication methods (i.e. USB, Bluetooth, Wi-Fi etc.), type, model, serial number, manufacturer, and features. A *micro grid* inherits all the attributes of a device and then adds additional attributes such as a profile, service(s), location, and contact information. A *micro grid* consists of one or

more device(s). The attribute named “Profile” assigns the role and classification of the *micro grid*. The services that a *micro grid* lists are provided by one or more devices. Figure 4.3 shows the *micro grid’s* attributes and its inheritance in an XML format.

```
<MicroGrid>
      <Contact>
            <Ip Address> <Ip Address>
            <Phone#>     <Phone#>
            <Email Address>    <Email Address>
      <Contact>
      <Profile>
            <Classification> <Classification>
            <ProfileType> <ProfileType>
            <Role> <Role>
            <Mobility>
                  <Mobile> <Mobile>
                  <DateInstalled> <DateInstalled>
            <Mobility>
      <Profile>
      <PrivateAuthKey> <PrivateAuthKey>
      <PublicAuthKey> <PublicAuthKey>
      <Services>
            <ServiceName1> <ServiceName1>
            <ServiceName2> <ServiceName2>
            …
            <ServiceNameN> <ServiceNameN>
      <Services>
      <Location>
                  <GPSLat> <GPSLat>
                  <GPSLon> <GPSLon>
                  <StreetInstalled> <StreetInstalled>
                  <IP Address> <IP Address>
      <Location>
      <Members>
            <Device1>
                  <Name> <Name>
                  <MAC Address> <MAC Address>
                  <CommMethod> <CommMethod>
                  <Type> <Type>
                  <ProdName> <ProdName>
                  <ProdModel> <ProdModel>
                  <Serial numbers> <Serial numbers>
                  <Manufacturer> <Manufacturer>
                  <features> <features>
            <Device1>
            <Device2> <Device2>
            …
            <DeviceN> <DeviceN>
<MicroGrid>
```

Figure 4.3 XML for micro grid attributes

Attributes of Hop and Route Map

A ***hop*** represents a segment of a road. It is typically, from one intersection of a road to another. A *hop* cannot be longer than 10 km in length, because it is a reasonably short distance to ignore the effect of the Earth's curvature. To keep the complexity and resource requirements to a minimum, only simple algorithms are used for distance calculations. If the speed limit of a road changes, or if there is a pedestrian crossing in the road, it will also break the road into multiple *hops*, even if there are no intersecting roads. In some cases a road will be further divided into multiple *hops* provided it meets certain criteria (i.e. the name of the road changes without any direction or speed change). One of the objectives is to keep the size of the hop information message to less than 82 bytes long so it can fit in the payload of any cell broadcast message. Table 2-1 shows the attributes and designated number of bytes for each hop and Figure 4.4 shows the attributes in a XML format.

Table 4-1 Attributes of a hop

Attribute	**Description / Example**	**Number of Bytes**
Header	Rte02Hop06	10
Beginning GPS coordination Latitude		4
Beginning GPS coordination Longitude		4
End GPS coordination Latitude		4
End GPS coordination Longitude		4
Hop length	Integer (maximum 10 Km)	2
Speed limit of the street	Integer	2
Attribute of the road (pavement type)		2
Road Curvature	- Straight, back, curved (S, B, C)	
Type of road entering	Highway, freeway, school zone, residential area, bike path	2
Traffic direction	one way, two way, divided/undivided	2
Number of lanes		1
Street name		20
Time Zone		1
Direction of the next hop	- for example left or right (L, R) - North, East, West, South (N,E,W,S)	1

A **route map** consists of one or more *hops*. It inherits all the attributes of a *hop* and adds few attributes to represent the routes current status. Figure 4.5 shows the relationship between hop and a *route map.*

```
<Hop>
      <Sequence> <Sequence>
      <Header> <Header>
      <StartLat> <StartLat>
      <StartLon> <StartLon>
      <EndLat> <EndLat>
      <EndLon> <eNDLon>
      <Length> <Length>
      <SpeedLimit> <SpeedLimit>
      <RoadPvType> <RoadPvType>
      <RoadCurvature> <RoadCurvature>
      <RoadType> <RoadType>
      <TrafficDir> <TrafficDir>
      <NumLane> <NumLane>
      <StreetName> <StreetName>
      <TimeZone> <TimeZone>
      <NextDirChg> <NextDirChg>
<Hop>
```

Figure 4.4 XML for hop attributes

```
<RouteMap>
      <CurrentHopNo> <CurrentHopNo>
      <CurSpeed> <CurSpeed>
      <ExpDuration> <ExpDuration>
      <TimeEntered> <TimeEntered>
      <Members>
            <Hop1> <Hop1>
            <Hop2> <Hop2>
            ...
            <HopN> <HopN>
      <Members>
<RouteMap>
```

Figure 4.5 XML for route attributes

Message Structure

All messages sent and received from any node will have a minimum of two criteria: it will have to fit the cell broadcast message structure and meet the OASIS Common Alert Protocol v1.2 (CAPv1.2). Although CBS supports up to 15 pages, only the first page will be used for most operational and traffic related messages. That is maximum message length is 82 bytes (or up to 192 characters depending on the codec). It is assumed that the codec used will fit minimum160 characters, so it can fit in a single SMS message. Some of the messages that will be sent are: heartbeat messages, registration and deregistration messages, traffic updates, q-end messages, hazardous road and weather conditions, cellular network conditions, disaster information, routes to safe locations, and other advisories. Non traffic related messages, messages with additional information, or route plans for emergency evacuations will contain multi page cell broadcast messages.

CAPv1.2 messages require several mandatory fields to be sent with each message. **In order to meet the two criteria, a mapping scheme has been created in this thesis**. This scheme ensures that all mandatory fields are included in every message sent, and that they also fit within a single CBS page. Table 4-2 shows the scheme and its relation to CAPv1.2 protocol (mandatory fields are shown in bold). If any non-mandatory fields including the ones in category "Resource" are required, they will be sent over subsequence SMS-CB page(s). The first page will have adequate information for the middleware to be aware of any upcoming pages.

Table 4-2 Mapping between CAPv1.2 standard and messages used in this thesis

Category	Element	Element name	CAP v1.2 Recommended Values	Thesis		
				Recommended Values	Number of Bytes	Number of Characters
Alert	**Message ID**	**Identifier**	Number or STRING - no spaces, commas, restricted character (< and &)	An integer between 1 and 16777216	3	3
	Sender ID	**Sender**	Globally unique — no spaces, commas or restricted character (< and &)	An integer between 1 and 16777216	3	3
	Sent Date/Time	**Sent**	dateTime format	UNIX Time	4	10
	Message Status	**Status**	"Actual", "Exercise" "System", "Test", "Draft"	1, 2, 3, 4,5	1	1
	Message Type	**msgType**	"Alert", "Update", "Cancel" , "Ack", "Error"	1, 2, 3, 4, 5, 6 ("Information")	1	1
	Scope	**Scope**	"Public", "Restricted", "Private"	1, 2, 3	1	1
Info	**Event Category**	**category***	"Geo", "Met", "Safety", "Security", "Rescue", "Fire", "Health", "Env", "Transport", "Infra", "CBRNE"	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, A	1	1
	Event Type	**Event**			5	5
	Response Type	response Type*	"Shelter", "Evacuate", "Prepare", "Execute", "Avoid", "Monitor", "Assess", "AllClear", "None"	1, 2, 3, 4, 5, 6, 7, 8, 9	1	1
	Urgency	**Urgency**	"Immediate", "past", "unknown"	1, 2, 3	1	1
	Severity	**Severity**	"Extreme", "Severe", "Moderate", "Minor", "Unknown"	1, 2, 3, 4, 5	1	1
	Certainty	**Certainty**			1	1
	Audience	Audience		All (1), Law enforcement (2), Emergency (3), Service(4), Resident(5), Parents(6), Teachers(7), Drivers(8), Biker (9), Walker (10)	1	1
	Effective Date/Time	Effective	dateTime format (2011-11-20T15:55:00-06:00)	UNIX Time	4	10
	Expiration Date/Time	Expires	dateTime format	UNIX Time	4	10
	Headline	Headline	Max 160 character		9	10
	Event Description	Description		"Congestion" "Construction" "Standstill Traffic"	14 to 60 (Default **21**)	15 to 80 (Default 24)
	Contact Info	Contact			20 (Default **5**)	20 (Default **12**)

Category	Element	Element name	CAP v1.2 Recommended Values	Thesis		
				Recommended Values	Number of Bytes	Number of Characters
Area	**Area Description**	**areaDesc**		1 for a point, 2 for two points, 3 for a triangle, 4 for Square, Rectangle or parallelogram, 5 for half circle, 6 for polygon (max 6 points), 7 for a circle	1	1
	Area Polygon	polygon*		Triangle will be used to represent most shapes. Latitude and longitude values for each point (4 + 4 bytes) will be sent.	8 – 48 (Default 24 for triangle) (0 if circle)	18-108 (default 54 for a triangle) (0 if circle)
	Area Circle	circle*	The circular area is represented by a central point given as a [WGS 84] coordinate pair followed by a space character and a radius value in kilometers.	A circular shape will be represented by the center point and the radius in meters (4+4+4 bytes).	12 (0 if not a circle)	22 (0 if not a circle)
	Altitude	Altitude			4	10

Alert Message Structure (normative) Document Object Model [7]

It is assumed that since these messages will be delivered over a specific channel in reality, all received messages will be CAPv1_2 compatible. According to CAPv1_2 the fields with a "*" are permitted to have multiple instances. More information on CAPv1.2 requirement can be found in [7]. This thesis recommends multiple instances be sent over different pages or messages, making it unnecessary to include the version number.

4.4 Process Description and UML Representations

This section describes important processes and corresponding UML drawings used by the software.

Initialization / Device Turn-up - At application launch, a *micro grid* will always be assigned the default role of user (*Ur*), but its profile will be loaded in the *current*

database. Firstly, the *device manager* will search for internal resources and services it can offer, and store them in the *current database*. It will then enable the *listener* so that it

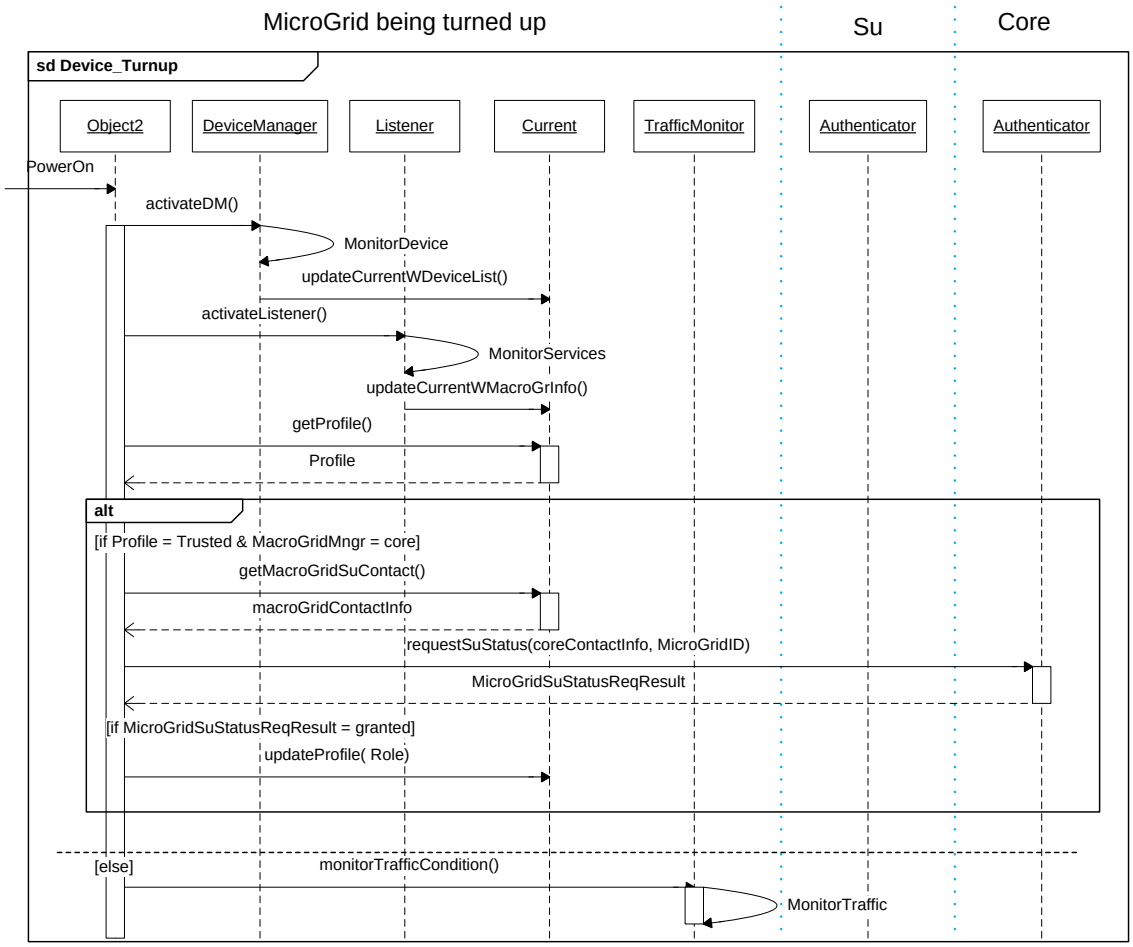

Figure 4.6 Micro grid initialization and turnup process

can receive SMS-CB or other messages. Then it will wait for *heartbeat* messages, which will enable it to find the contact information and status of the *macro grid manager*. If the *micro gird* detects the absence of local *Su* or *BuSu* it will try to register itself as the *Su* or *BuSu* provided it has the appropriate profile and authority. Upon receiving a *register message*, the *core* will authenticate the *micro grid* and assign or deny it the role. If a *Su* and *BuSu* are present, the *micro grid* will continue with the *Ur* role and start the traffic

monitoring process while waiting for CB messages. It will expect *heartbeat* message at regular intervals. This process is described in UML format in Figure 4.6

If the *micro grid* does not receive any CB message or if registration to the cellular network fails, it will start using alternate wireless communication methods. If it gets a *heartbeat* message via an alternate method, it will try to register with the *Su micro grid*. Details of alternate wireless communication methods are beyond the scope of this thesis.

Device Registration/De-registration to/from a *Macro Grid* - During normal operation (i.e. the telecommunication network is functional) there will be no need for a *micro grid* to register. Only *trusted micro grids* will perform registration when they enter a *macro grid,* so that in the case of an emergency or a *Su* failure, an alternate *micro grid* can take over the *macro grid manager's* responsibilities. This also ensures lower network traffic. Each *micro grid* will report its current location and speed at the time of registration and deregistration. The scheduler of the *macro grid manager* will initiate a timer for each of the registered *micro grids*. If no *heartbeat* is sent before the expiry of the timer, then the *micro grid* will be implicitly deregistered from the *macro grid*. The registration and deregistration process is shown in UML format in Figure 4.7 and Figure 4.8respectively.

In the case of a telecommunication network failure or congestion, all *micro grids* will have an option to register to the local *micro grid manager* using an alternate wireless method. This will ensure that any emergency messages pass to all the *micro grids* in the *macro grid*, in case cell broadcast fails. Another instance when the registration process will be active is when a *macro grid* consists of permanent sensor deployments. It will be among the *trusted micro grids* (i.e. fixed sensors).

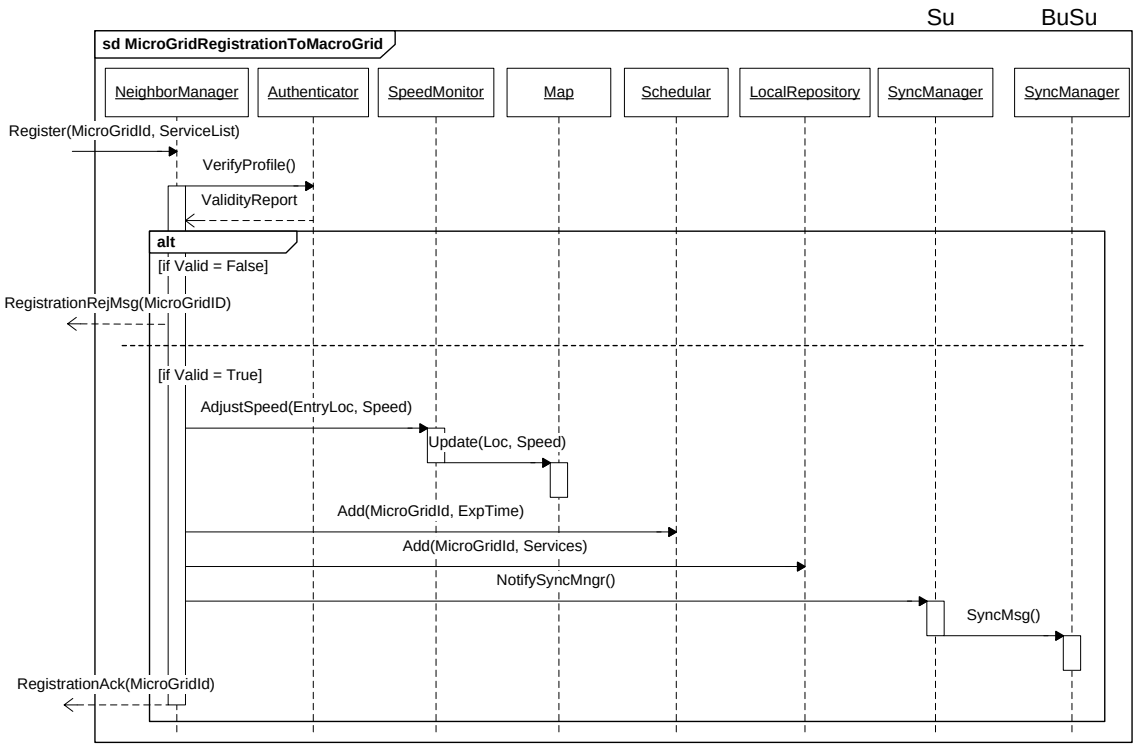

Figure 4.7 Micro gird registration process

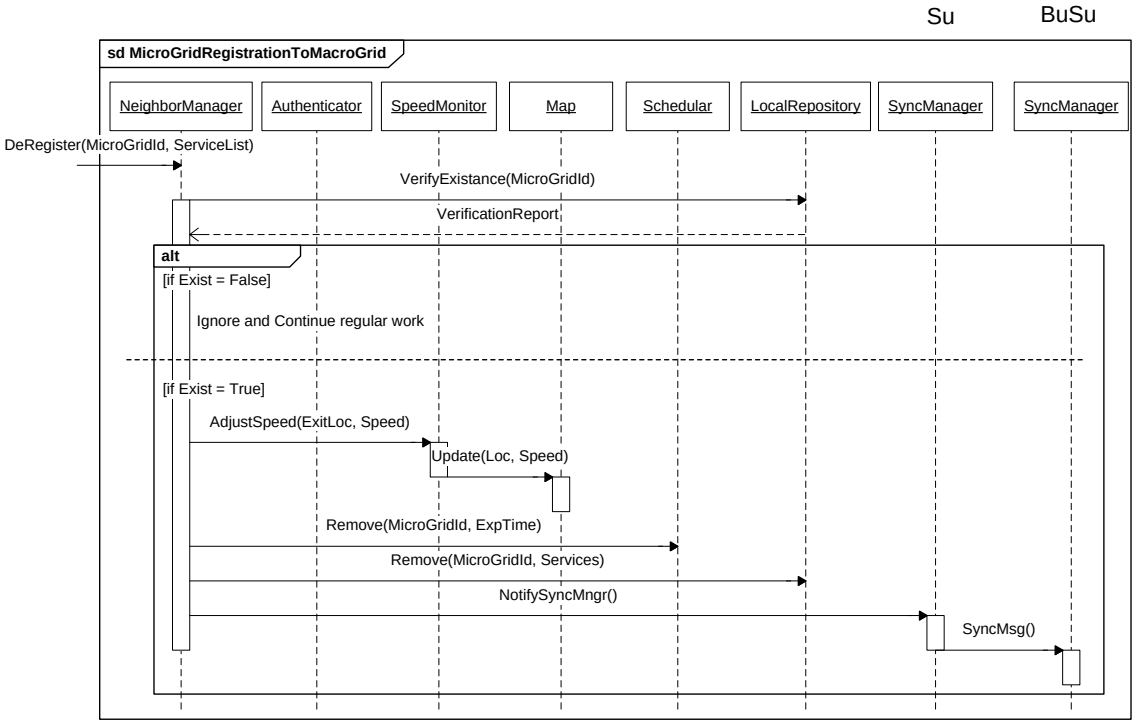

Figure 4.8 Micro grid de-registration process

Handover Process between *Macro Grids*

The handover process is described here. It is also presented in UML format in Figure 4.9. Handover is used when the congestion detection algorithm used is based on *macro grid* dwell time. It is also used when a *macro grid* runs in standalone mode, or when a *trusted micro grid* enters/exits a *macro grid*.

***Micro Grid* (node)**

Detects new cell ID
- Records last sampled location and speed
- Records current sampled location and speed
- Waits for cell broadcast message
- Retrieves *macro grid* ID from the broadcast Message
- If different *macro grid* ID then
  - Send "Leave message" to old *macro grid*
  - Calculate approximate time it will be present in the *macro grid*
  - Updates "Current" database
  - Sends "Join Message" to the new *macro grid*
    - waits for Acknowledgement message
      - If no Acknowledge message received then sends it to *BuSu*
- Continues to check speed every second
- Continues to check sensor readings
- Continues to listen to CB for updates

New *Macro Grid Manager*

- Authenticates the *micro grid*
- If *micro grid* is invalid
  - then ignores the message
- If valid, updates its local repositories
  - Updates neighbour database
  - Adjusts speed database for that location (i.e. entry point)
  - Adds to scheduler so it verifies it presence by sending a message
  - Sends sync message to *BuSu*

New *Macro Grid Manager*

- Verifies the *micro grid* from which the leave message came from exits or not
  - If *micro grid* does not exist in the neighbour Database then
    - Ignore
  - If it exists then:
    - Removes the services provided by it from it repository
    - Removes from scheduler

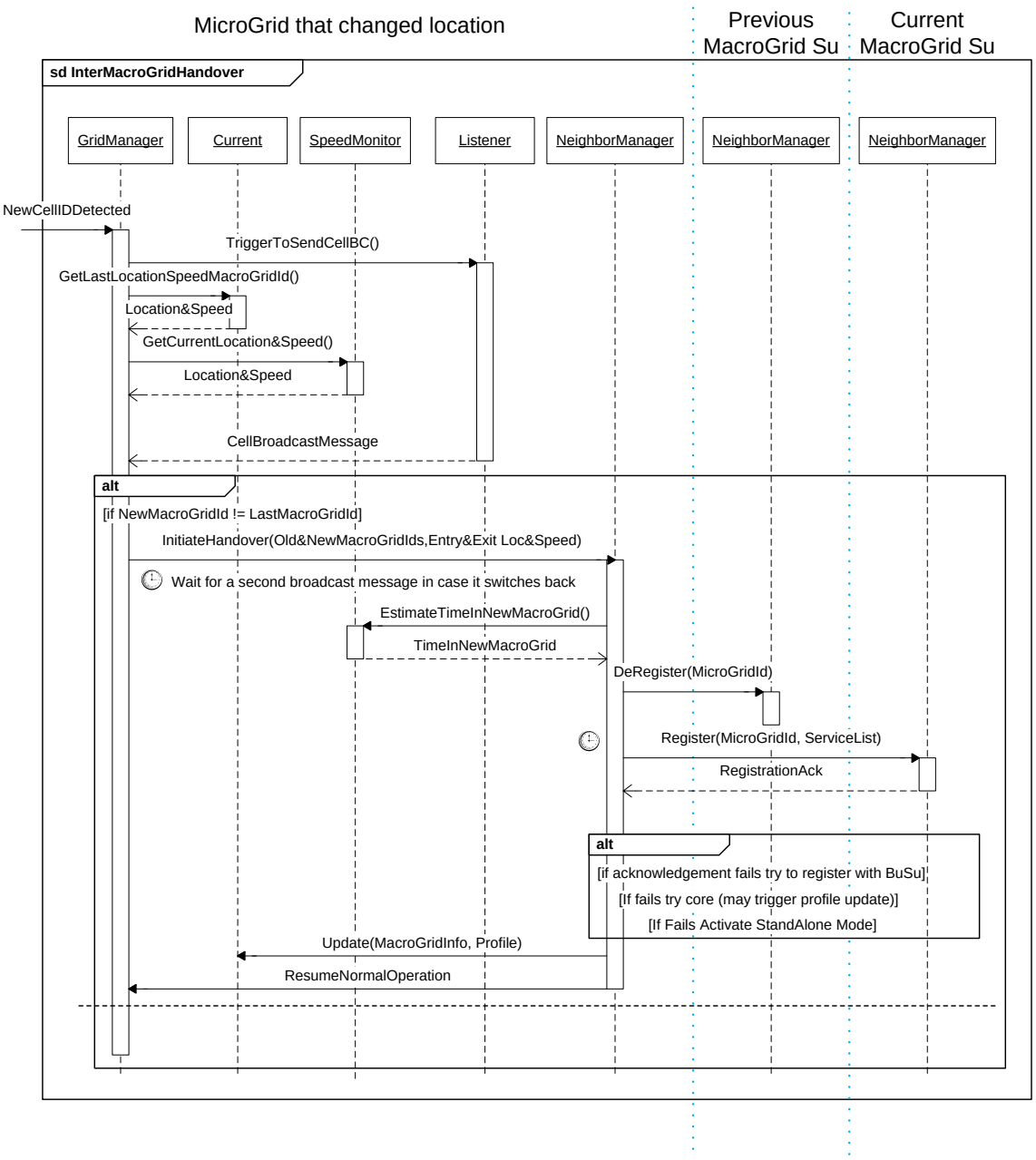

Figure 4.9 Micro grid handover process between two macro grids

Profile Change Trigger Process - Profile changes will be triggered via OTA or software push and will be primarily controlled by the *core* unless there is a total communication loss with the *core*.

4.5 Congestion Detection Algorithms

This section provides congestion detection algorithms used in the demonstration and ends with additional algorithms that were considered for the thesis.

Every second, the application monitors its current location and speed. It also verifies which hop it is in and compares its speed with the expected speed for that hop. If the speed is within acceptable range, the congestion detection flag is set to zero or left unchanged if already set to zero. If the speed is below a threshold (say 10% below speed limit) of the actual posted speed limit, it initiates the congestion detection process. Prior to declaring a hop or location to be congested, it waits for three timers, called *congestion detection timers*, to expire. All of these timers are dynamically calculated for the hop they are in. These timers prevent the false declaration of congestion, since there are many valid reasons to slow down beside congestion (i.e. traffic signal, pedestrian crossings, stop signs etc.).

The first step of the congestion detection process is to raise the *congestion detection* flag from a value of zero to one, and to start the first *congestion detection* timer. The first timer is equal to the total expected hop delay, which is the summation of deceleration time (time it takes to stop a vehicle from speed limit of current *hop*), the reaction time, the known end of hop delay (i.e. duration of traffic light or stop sign), plus the acceleration time (time it takes to get a speed equal to speed limit of current *hop* from speed of zero).

If the speed remains below the speed limit after the expiry of the first timer, the second timer is started which is equivalent to acceleration time. If it is still below the speed limit

after the expiry of the second timer, the third timer is triggered. The third timer is also the summation of deceleration time, reaction time, known end of hop delay, plus acceleration time. The third timer could be the same as the first timer if it is still in the same *hop*. This is to give adequate time in case a vehicle needs to be stopped at the same traffic light twice, which is often the case even when there is no congestion (i.e. some people naturally drives slower). Typically, people will be able to reach speed limit (which will reset the congestion flag to 0) in normal traffic conditions before having to stop at the same light twice.

In order to prevent false declaration of congestion, prior to sending a congestion message, the *micro grid* compares if it has reached its final destination or not. If it is at its final destination, then it will shut down the application. If it is a smart phone and the final destination has not been reached, it will also try to retrieve information from the vehicles OBD system to check it the parking break or gear is engaged, or if the door open indicator is on. If they are, it will conclude that there is no congestion and no message will be sent. If the above is unavailable then it will prompt a message on the user screen to confirm congestion. If the user does not respond to the prompt within five seconds it will send the congestion message. Note, the system expects a user will not respond to the prompt while driving, therefore, it will send the message.

After all three timers expiry and verification of true congestion, the *micro grid* sends a *congestion detect* message to the *macro grid manager* with its current location, speed, and heading. The reason for sending most recent location, speed, and heading instead of the one recorded at the beginning of congestion timer 1 is to avoid misinformation in case the congestion has cleared. It also accounts for the time it takes a traffic light to be green.

However, historical traffic information is buffered which is typically adequate for three timers.

Upon receiving the *congestion detect* message, the *macro grid manager* verifies the source. If the message is from a *trusted micro grid* it then immediately forwards the message to CBS for broadcasting to the entire *macro grid*. If the message is not from a *trusted* source, then the *macro grid manager* waits for five other similar *micro grids* or a *trusted micro grid* (whichever comes first) to report the similar congestion for the same hop prior to forwarding it to the CBS. This is to ensure integrity of the system and avoid spoofing. The CBS then broadcasts the message to the rest of *micro grids* in the *macro grid*. The actual validation process is beyond the scope of this thesis. This is another precautionary measure by the *macro grid manager* to prevent false declaration of congestion.

The *micro grid* also verifies if the received congestion location is part of its current route. If the location is not part of its current route, the application briefly displays a message stating that it received a *congestion detected* message but then assures that it does not have an effect on the driver's current route. If the congestion is in its route, it then verifies if the congested location is ahead or behind its current location. If it is behind, displays a message stating it received a congestion message for a location behind it.

If the congestion is in the *micro grid's* path and is ahead of its current location, it will store the message in its *current database*. The application will then display the distance between itself and the troubled location. This distance will be updated every second until the vehicle passes the reported trouble location. In addition, the application will look at

its own traffic history (speed, distance travelled, timestamp etc.) and determine the maximum speed it drove in the last "total expected hop delay" period (equivalent to its first timer). If the speed reported in the received congestion detect message is faster than its maximum speed in the recent history and if it is further than a threshold distance (say 50 meters) from the reported location, and already has a congestion detection flag set to a value greater than 1, it will immediately send a message to the *macro grid manager* with its own current location and the maximum speed found in that period. In this case, it does not wait for all three timers to expire. Any *micro grid* in the grid meeting these criteria will do the same. The *macro grid manager* will compare all the responses as a result of this broadcast and rebroadcast the message containing the furthest location after its waiting timer expires. This is how actual q-end location is detected. The messaging should stop once the actual q-end location is found. After receiving this message, no node will send any more messages, unless it is behind the threshold distance which will then automatically update the q-end location. Also, the final message will be rebroadcasted multiple times so upcoming traffic that missed the original message will be aware of the situation. Repeated messages will be filtered by the middleware running in the *micro grid*. As new vehicles arrive and the q-end gets further away, the newer *micro grid* will correct the q-end location. If the rate of new vehicle arrival is slower than the running speed of traffic, the messaging will stop. Eventually once the congestion clears, the *micro grids* will start sending clear congestion message.

The congestion message will also be broadcasted once to the adjacent *macro grids*, thus warning any nearby *micro grids* of the unexpected delays in the upcoming *macro grid*. This will also take care of any *micro grid*, in the border area, (that is registered to a cell

belonging to a different *macro grid*, but located in the area of the congested *macro grid).* Figure 4.10, Figure 4.11, and Figure 4.12 shows the congestion detection algorithms used by mobile *micro grids* and *macro grid manager.*

```
Retrieve CurrentSpeedLimit
Retrieve CurrentSpeed
   Compare CurrentSpeedLimit with CurrentSpeed
      if CurrentSpeed > CurrentSpeedLimit times AcceptableHighPercentage
         Set CongestionFlag to -1
      else
          if CurrentSpeed < CurrentSpeedLimit times AcceptableLowPercentage
              Retrieve CongestionFlag
              if ((CongestionFlag equals 0) or (CongestionFlag equals -1))
                  Set CongestionFlag to 1
                  Get Calculated CongestionTimer1 for currentHop
                  Set CongestionTimer1Expiry time
              else if CongestionFlag equals 1
                  Check if CongestionTimer1Expiry time has expired
                  If it has expired
                      Set CongestionFlag to 2
                      Get Calculated CongestionTimer2 for currentHop
                      Warn User about potential congestion
              else if CongestionFlag equals 2
                  Check if CongestionTimer1Expiry time has expired
                  If it has expired
                      Set CongestionFlag to 3
                      Get Calculated CongestionTimer3 for currentHop
                      Set CongestionTimer3Expiry time
                      Warn User about potential congestion
              else if CongestionFlag equals 3
                  Check if CongestionTimer1Expiry time has expired
                  If it has expired
                      Set CongestionFlag to 4
                      Verify it is not due non-congestion related
                          If Current location is not equal final destnation
                          Or Parking brake is not engaged
                          Or Door opened is not equal true
                             Initiate Send message to SuNode (MacroGridManager)
                                Determine Severity of Congestion
                                Generate CongestionDetect message with proper severity & format
                                Send message to MacroGridManager
                                Warn User about congestion
```

Figure 4.10 Congestion detection algorithm#1 of a mobile *Micro Grid*

```
Extract congestion location and other parameters.
   Determine if the congested location is part of its current route or not.
      If not part of its route
         Rejects the message
         Displays a message stating that it received a non effecting congestion message
      If part of the route, determines if congested location is ahead or behind
         If behind
            Briefly display a message (for a second) stating congestion is behind
         If ahead
            Calculate the distance from itself and the congested location
            Display the calculated distance and congestion speed on the Alert screen
            Look up own travel location and speed history, and congestion flag
               If congestion flag is greater than 1 (i.e. suffering from congestion)
                  Send a message to the macro grid manager
                     Inform congested location is actually further behind than location broadcasted
                  Until the congested location is past it does the following:
                     Calculate the distance between itself & congested location every second
                     Warn about the speed and distance every second.
```

Figure 4.11 Action taken by a Micro Grid after receiving congestion detection broadcast

```
Upon receiving a CongestionDetectMessage
Set PotentialCongestion equal True
Check the profile of the sender micro grid
If Trusted
    Initiate congestion detection broadcast
    If CongestionWaitTimer not equal zero
       Reset CongestionWaitTimer to zero
       Reset CongestionMsgRecdCounter = 0
Else
   If CongestionWaitTimer equal zero
      Activate CongestionWaitTimer
      CongestionMsgRecdCounter = 1
   Else If currentTime equal CongestionWaitTimer
      Reset CongestionWaitTimer to zero
      Reset CongestionMsgRecdCounter = 0
   Else
      CongestionMsgRecdCounter = CongestionMsgRecdCounter + 1
      If CongestionMsgRecdCounter > 5
         Initiate congestion detection broadcast
         Reset CongestionWaitTimer to zero
         Reset CongestionMsgRecdCounter = 0
      Activate CollaborationTimer
      Wait for Micro grid feedbCK

Wait for CollaborationTimer to expire
   If CollaborationTimer expire is true
      Compare which micro grid reported furthest q-end location
      Broadcast furthest q-end location and corresponding speed
      Reset CollaborationTimer
Until CongestionClearMessage is received
   Wait for updated q-end location information
      if new location is received
         Broadcast new q-end location
When CongestionClearMessage is received
   Reset PotentialCongestion equal False
```

Figure 4.12 Congestion message processing by a *Macro Grid Manager*

The same principal will be used if there is a fixed sensor deployment. The only difference is all fixed deployed sensors are considered *trusted*. Each sensor responsible for congestion detection will report the speed and location it detects, but it will also consider the expected delays for the location.

Other Algorithm Considered

Most of these algorithms use the same basic principle as the first algorithm. Basis difference is the main criteria used to determine congestion i.e. speed based or distance based. Due to the similarities, the pseudo codes of these algorithms are not provided.

- Determine congestion based on distance travelled during a time interval and compare it with a predicted distance during the same time interval, instead of comparing the speed. For example, if the traveled distance is 0 or less than a threshold length (say 10 meter) after waiting for expected total hop delay time (acceleration time, deceleration time, reaction time, plus end hop delay) then consider the hop to be congested.
- Use comparisons for both distance and speed to conclude that a hop is congested.
- Compare calculated time in a hop and / or in a route with the actual time spent in a hop or in a route. If it has not traveled a threshold distance in a threshold time (say half the distance, at half the calculated time) then declare the hop congested. This method is not very practical, because depending on the route length, it may be too late to declare the congestion.
- Compare the calculated time that a *micro grid* should be in a *macro grid* based on its current route plan, with the actual time spent in that *macro grid*. This will be done by first intersecting the (i.e. *macro grid*) perimeter with the route map to find the entry and exit points. Then, it will calculate the expected time in the *macro grid* while taking into account the speed limit, as well as presence and number of traffic lights, stop signs and other obstructions. If the *micro grid* remains in a grid longer than the estimated time, it will initiate the congestion detection process. Again this

algorithm will not be efficient due to variation in *macro grid* size, number of turn in the route, and variation of speed limits it contains. The efficiency of this algorithm could be improved slightly by comparing a percentage of distance traveled to the same percentage of expected time, but still it will not be very accurate depending on several factors such as the location of the traffic light or other obstructions. Research with similar concept was done in [15] and [16], but they used cell dwell time instead of *macro grid* dwell time and concluded that more work is required to be able to successfully determine congestion.

- Use of actual sensors and artificial intelligence. Actual intelligence used will depend on the sensor capabilities. This will apply more towards fixed sensor deployments. For example, when using two sensors (one at entry point and one at exit point) of a section of a road. The sensors will count, classify (i.e. based on size), and read the vehicle speed to determine length, location, and speed at the end of the queue.
- The final algorithm considered is a combination one, two or all of the above methods

4.6 Operational Modes

Normal Operation (mobile network is available) - In normal conditions, each *micro grid* will receive a broadcasted *heartbeat* message every four seconds via SMS-CB. A *macro grid* may or may not have a local *macro grid manager* present, in which case the *core* will take the *macro grid manager's* responsibilities. The *heartbeat* messages will contain *macro grid* specific information, including: message ID, sender ID, time sent, *macro grid* coverage area, *macro grid manager* contact information and the type of node (i.e. local *Su* or the *core*) responsible for the grid management. It will also be able to tell whether or not the *macro grid manager* is mobile and its status. In normal conditions no

upstream messages will be sent to the *macro grid manager*, however, each *micro grid* will continue to monitor traffic conditions. Also in normal operating state the *macro grid* will act as a centralized system unless a local *Su micro grid* is available to take the *macro grid manager* role. If a *micro grid* finds any traffic anomalies it will send it using the cellular network.

Standalone Operation - In the case of a disaster during which the *core* is unreachable or the cellular network is unavailable, the *macro grid* will operate in standalone mode. In this mode, the *micro grids* will use the ad-hoc mesh network. The *macro grid* will be self sufficient and will have the ability to operate independently in terms of detecting the queue end and sending other relevant information to its *micro grid* members. In this mode, many more software modules will be active. For example, the *neighbor manager*, and *macro grid* registration/deregistration process will be initiated by all *micro grids*. This will ensure that a new *Su* or *BuSu micro grid* can be selected quickly if such a *micro grid* leaves the *macro grid*. It is assumed that at least one *micro grid* will be aware of the physical scope of the *macro grid*.

Standalone switch over process will be triggered by either the absence of an HB message, the detection of a cellular network issue, or based on information received in a SMS-CB message. The *core* can be unreachable due to hardware or software problems or network issues (non-cellular network related on the core side). The cellular network can have different types of problems as well. For example, there could be no coverage (i.e.: no tower installed or transport/backhaul network to the tower is damaged), congestion in AS (RF level or radio controller level), or congestion in NAS (call processing abilities impaired).

In the case of a *core* problem (but not a cellular network problem), the *macro grid* operation will be similar to normal operations in the *micro grid* point of view, provided the *macro grid manager* is a local *Su micro grid*. If the *macro grid manager* was the *core,* then the *macro grid manager* selection process will begin. The *core,* being the *macro grid manager*, indicates absence of trusted elevated *(Su) micro grid* in the *macro grid*. However, there could still be *micro grids* with a *trusted* profile in the *macro grid*. The *trusted micro grid*(s) will first send a HB message to the entire *macro grid*. If there is more than one trusted *micro grid* they will all send a HB message. The *micro girds* will store the information of the last received HB message. The *trusted micro grids* will then communicate among themselves to select the *macro gird manager* that will become the new *Su micro grid*. There will also be a *BuSu micro grid* selected. The selected *macro grid manager* will then continue to send the HB message. The *Su* and *BuSu micro grids* will send HB messages between each other and synchronize other relevant information.

If there is no *trusted micro grid* in the *macro grid*, the *micro grids* with *subscribed* profiles will attempt to send broadcast messages using an alternate communication (i.e. ad-hoc wireless mesh) method, after waiting to hear from a *trusted micro grid* for a predefined time. Note: these *micro grids* will not have authorization in the CBS, therefore, they will not attempt to use SMS-CB. They will also go through a selection process to choose the *macro grid manager*. If there is no *micro grid* with *trusted* or *subscribed* profile, then *micro grids* with *un-trusted* profile will send messages using ad-hoc mesh network to select a *macro grid manager*. *Un-trusted micro grids* will wait even longer before they begin to send messages. Once a *macro grid manager* is selected

all *micro grids* will have to register to it to ensure message delivery. However, a *micro grid* can deny registration due to security concerns.

Cellular Network Congestion - In case of cellular network congestion, be it in AS or NAS, the SMS-CB should still work. If the *macro grid manager* can reach the *core* via an alternate communication method, it will continue to send HB and other instruction (i.e. to use alternate communication methods and contact information, to activate other processes, etc...) via SMS-CB. Otherwise, it will use the ad-hoc mesh network. However, the *micro grids* will use the ad-hoc mesh network to communicate back to the *macro grid manager*. It could also be triggered by the *micro grid* itself, if the *micro grid* loses cellular connectivity.

Request for a Service – When a *micro grid* requests for a service, it will send its request to the *macro grid manager*. If the requested service was, for example, to download a song, and the *macro grid manager* is a local *Su*, upon receiving the request the *Su* will search its local resource repository. If the information is available within the *macro grid*, the *Su* will provide the *micro grid`s* contact information (the provider could be the *Su*, *BuSu*, or a different *micro grid*). If the information is unavailable in the *macro grid*, the *Su* will send the request to the *core*. The *core* will respond by sending its ftp server location back to the *Su.* The *Su* will forward the information to the requester. Upon receiving the response the requesting *micro grid* will make direct connection with the provider over the cellular network, a Wi-Fi network or another network. Note that it is assumed that both the consumer and the resource provider are aware of any potential security and privacy risks involved in the transaction. There is typically security built into this type of service, but information on the subject is beyond the scope of this thesis.

CHAPTER 5 IMPLEMENTATION, RESULTS, AND ANALYSIS

This chapter provides information on the software development and its implementation to provide the proof of concept presented in this thesis. This section describes how different services mentioned in the above sections are implemented.

5.1 Development and Test Environment

This section starts with implementation assumptions then introduces different application development and simulation environment. It also lists the classes and methods used in the application development. Formulas used for acceleration/deceleration and distance calculation can be found in [47] and [48] respectively.

Implementation Assumptions

- If a *micro grid* travels in a location for which there is no loaded hop information (i.e. speed limit, start and end location of the hop etc. are unknown) it either uses 50 km/h or the last known speed limit as the speed limit in the *current database*
- There will be less than 100 hops in a route
- Reaction time of the average driver is 1.5 seconds
- The average acceleration is constant for all vehicles and drivers, and is equal to 5.3108352 km/h/s (i.e. 1.475232 $m/s^{2)}$, which is derived from 3.3 Mph/s [45]
- The deceleration rate used is 5.6 m/s^2, which is equal 20.16 km/h/s, and is the worst case scenario found in a study done in [46].
- While calculating the expected duration in each hop, it is assumed that a vehicle will start from 0 km/h at the start point of each hop, that it will have to stop at the end of the hop, and that it will have to wait for any expected delay (i.e. for the

traffic light or stop signs). The overlap of distance travelled during acceleration and deceleration is ignored to keep the calculation simple and to add buffer for error. Centripetal acceleration is also ignored in this calculation. Although this approach prolongs the congestion detection by few seconds, it also helps to prevent false declarations of congestion.

- Sensor limitations such as delays (i.e. GPS synchronization time), sampling time, accuracy, and line of sight are assumed to be negligible.
- GPS signals will always be available regardless of high rise buildings, weather, and forests.
- There will always be smart phones present with the application running in them.

Code Assumption - Several assumptions were made while writing the code. The application will be turned on only when it is travelling on the road and need to participate on the grid to contribute to traffic condition detection. The congestion happens only in one spot of a route and in one direction. All roads are 50 meters-wide and consist of single lanes. The roads are straight (curvatures are not accounted for in this version of the software). The use of SMS instead of SMS-CB (cell broadcast) is adequate for the proof of concept.

Application Development Environment

The software has been developed in the BlackBerry® Java® Development Environment (JDE) version: 6.0.0.37, a fully integrated Java Micro Edition (Java® ME) based platform. Although BlackBerry JDE is based on Java technologies, certain packages and classes have slightly different implementations than what is defined in the Java ME SDK

3.0 packages. Microsoft Notepad and Java SDK 3.0 were used to code and compile the server application that represents the SMSC or the CBS.

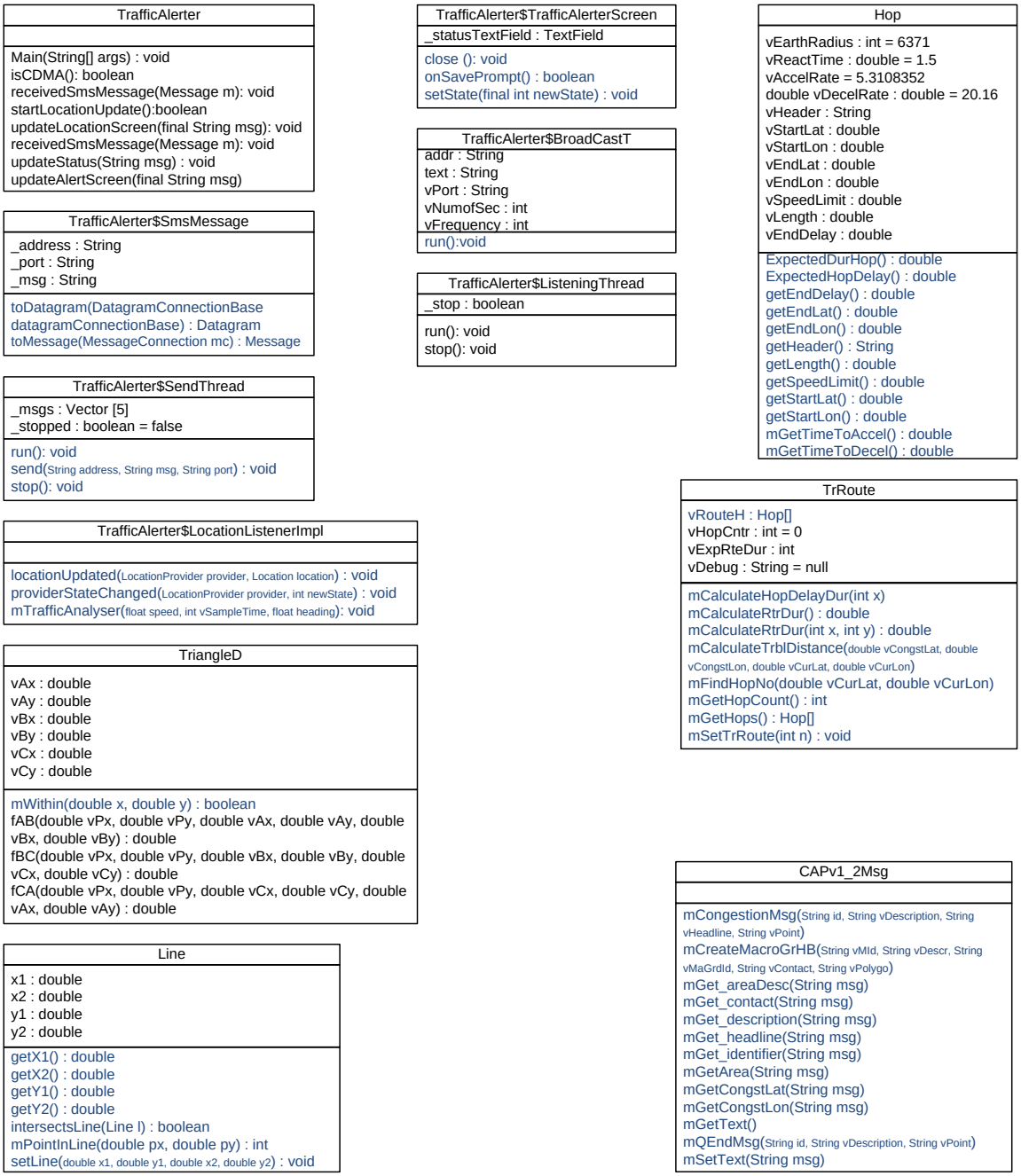

Figure 5.1 Classes used in the developed application

The software development started using two sample applications and a sample server that came with the BlackBerry JDE. The name of the sample applications are "gpsdemo" and

"smsdemo". These two applications were combined, and unnecessary classes and methods were removed. Many new classes and methods were added to build the new application. Figure 5.1 shows all the classes used in the developed software. Note, that the class and method names are slightly different. For example, *Traffic Monitor* is called mTrafficAnalyser, *Wait Time Calculator* is implemented as mCalculateHopDelay() and mCalculateRteDur(), and *Message Generator* as CAPv1_2Msg.

The sample server was originally designed to listen to a specific TCP/IP port for incoming SMS and to return a copy back to the senders listening port as a new SMS. The server was first modified to learn and store the phone number and the corresponding port number of any simulated phone that sent the SMS message. Next, CBS characters were developed such that if a SMS message was destined towards the CBS phone number, the server would forward the message to all of the phones that it had learned about after verifying sender's authority. This server was re-modified, to only act as a SMSC. The phone with the *Su micro grid* role was given broadcast responsibilities. The *Su micro grid* accomplished this responsibilities by sending individual SMS via the server to every phone that was programmed to receive the broadcast message.

The reasons behind using SMS are: a) the BlackBerry JDE simulator does not support sending CBS using a program [49] and b) the real life network where the application was tested did not have CBS. However, it was decided that the use of SMS was adequate in order to prove the concept presented in this thesis because the API responsible for listening to incoming messages on a BlackBerry is the same one as defined in "JSR 205: Wireless Messaging API 2.0", which extends and enhances "JSR 120: Wireless Messaging API." JSR 120 is designed to receive both SMS and cell broadcast message

[50]. Therefore, it was assumed that if the concept worked for SMS, it would work for CBS as well. Also, the CBS message length aligns nicely with SMS message length.

Simulator Environment

Although BlackBerry JDK came with a device simulator, two different simulators representing different BlackBerry phone models and operating system were used to ensure operability in different environments. The simulators also simulated their movement. The simulators used were: BlackBerry Smartphone Simulators (6.0.0.534 (9800) and BlackBerry Smartphone Simulator 7.0.0.261(9850). Both programs offered similar features and allowed GPS data to be imported from a file that follows the format defined by the National Marine Electronics Association (NMEA). It also allowed individual locations to be added in order to create a route.

Hardware platform used for the simulation - All of the development, tests, and simulations were done on a computer with Intel(R) Core(TM)2 Duo CPU P8600 @ 2.40 GHz processor with 4.00 GB of random access memory and a 32-bit Windows Vista operating system.

Hardware used in real life test - Although the simulated environment was used for initial testing, the application was road tested using two different models of BlackBerry phones designed for the UMTS network. The models and their corresponding operating systems were: BlackBerry 9800, 6.0 Bundle 2647 (v6.0.0.600, platform 6.6.0.223) also known as the BlackBerry Torch 9800. The second phone was a BlackBerry 9860, 7.0 Bundle 1355 (v7.0.0.261, Platform 5.0.0.469).

The application was tested on a UMTS (specifically HSPA+) network.

5.2 Experimental Setup, Data Collection, and Application Capabilities

Global Positioning System (GPS) coordinates were used as the unit for location. Although GPS coordinates do not fit well with Cartesian coordinates and are represented using degrees, minutes, and seconds it has many advantages over other units. It can accurately locate any point on the surface of the earth and can be cross-referenced with geographic maps. GPS coordinates are available in many affordable handheld devices (i.e. mobile phones or GPS navigation systems) and built-in vehicle navigation systems, and coordinates can be easily converted into decimal points. Five decimal points have been used for location calculation, since it provides accuracy of close to 1 meter (0.00001° equals 1.1 meter) [51]. This way, the 32 Bits (4 bytes) Java floating number can be used without compromising any precision.

Capabilities of the software

This section lists all the capabilities of the software that were developed and tested:

- The application uses less than 1% of the CPU resource and 1.1MB of memory.
- Can send or receive SMS messages.
- Retrieves current GPS location, speed, heading, altitude from the internal GPS API.
- Can load defined routes in its memory and extract hop attributes.
- Attributes of the hop included this version are: Hop ID, start and end GPS points, length, posted speed limit, and known end of *hop* delay (length of red light).
- Identifies which hop of the route it is currently situated in.
- Dynamically calculates time it takes to reach the speed limit from rest.
- Dynamically calculates time it takes to stop from speed limit to speed of zero.
- Can monitor and compare current speed with hop's speed limit.

- Performs congestion detection as described in first part of section 0.
- Partial implementation for other congestion algorithm (i.e. estimate duration in a hop or route, ability to perform line intersection etc.)
- Dynamically generates congestion or other messages in CAPv1.2 format.
- **Identifies Congestion Severity**
  - ❖ If current speed equals less than 10% of speed limit after expiry of congestion timers, it then checks for severity.
    - If speed equals 10% of speed limit, then sends message with standstill traffic flag
    - If speed equals 50% of speed limit, sends message with major congestion flag
    - Otherwise sends congestion message with no flag

Process used when new SMS message is received

- Verifies if received message is warning related or not
  - ❖ If message is warning related then check heading
    - If header equals *heartbeat* message check if it is new
      - ➢ If *heartbea*t is for the same *macro grid* with no change
        - ○ Ignore the message.
      - ➢ Otherwise if *heartbeat* is for new *macro grid* then
        - ○ Extract own current location from the *currentDB*
        - ○ Extract *macro grid* coverage area from the *heartbeat* message
        - ○ Verify if inside the *macro grid*
          - If inside, update *currentDB* with new *macro grid* information
          - Otherwise ignore the message and remove old *macro grid* information
    - If heading equals a congestion message then verify own role (i.e. *Su* or other)

- If role equals *Su*, verify if it is the authorized *Su*
  - If authorized *Su* then
    - Broadcast the message to all the other test phones that are hard coded
    - Initiate “Congestion message received process”
  - If not authorized *Su* then
    - Initiate “Congestion message received process”
- If role equals anything but *Su*
  - Initiate “Congestion message received process”

- If heading equals a registration message then
  - Initiate registration process (not implemented)
- If heading equals a de-registration message then
  - Initiate de-registration process (not implemented)

- If message is not warning related then
  - Display the message as a received SMS message

Congestion message received process

- Extract congestion location and other parameters.
  - Determine if the congested location is part of its current route or not.
    - If not part of its route
      - Rejects the message (for the sake of debugging it displays a message for a second or less stating that it received a non effecting congestion message)
    - If part of the route, determines if congested location is ahead or behind
      - If behind
        - Briefly displays a message (for a second) stating congestion is behind

- If ahead
  - Calculates the distance from itself and the congested location
  - Displays the calculated distance and congestion speed on the Alert screen
  - Looks up own travel location and speed history, and congestion flag
    - If congestion flag is greater than 1 (i.e. suffering from congestion)
      - Sends a message to the *macro grid manager* informing the congested location is actually further behind than the location broadcasted.
  - Until the congested location is past it does the following:
    - Updates the distance between itself the congested location every second and warn about the speed and distance every second.

Screenshots of the Application

This section shows different screenshots of the application. The name of the application is "TrafficAlerter." The first screen after launching the application is a **disclaimer,** warning people with privacy concern that their location information will be sent. Next, it allows the user to choose the **route** they wish to travel. Then the application prompts the user to choose a **role** (Su role[1] or User). It then asks if a copy of incoming SMS messages should be sent to the inbox or not. Finally, the application launches and displays relevant information. The application screen is divided into four sections. The first section is to allow for SMS messages to be sent. The second section is for displaying internal status messages, including current time, role, location, heading, speed, hop and route information (ID, speed limit), expected (dynamically calculated for each

[1] Only one phone number is hardcoded to act as a full *Su micro grid* to prevent accidental misuse.

hop and route) time in a hop and in the current route, maximum and minimum speed in the past 200 seconds, and speed status flag. The third section is for alert which shows messages based on external sources. The last section displays non-warning SMS messages. Figure 5.2 shows screenshots of different steps of the application launch.

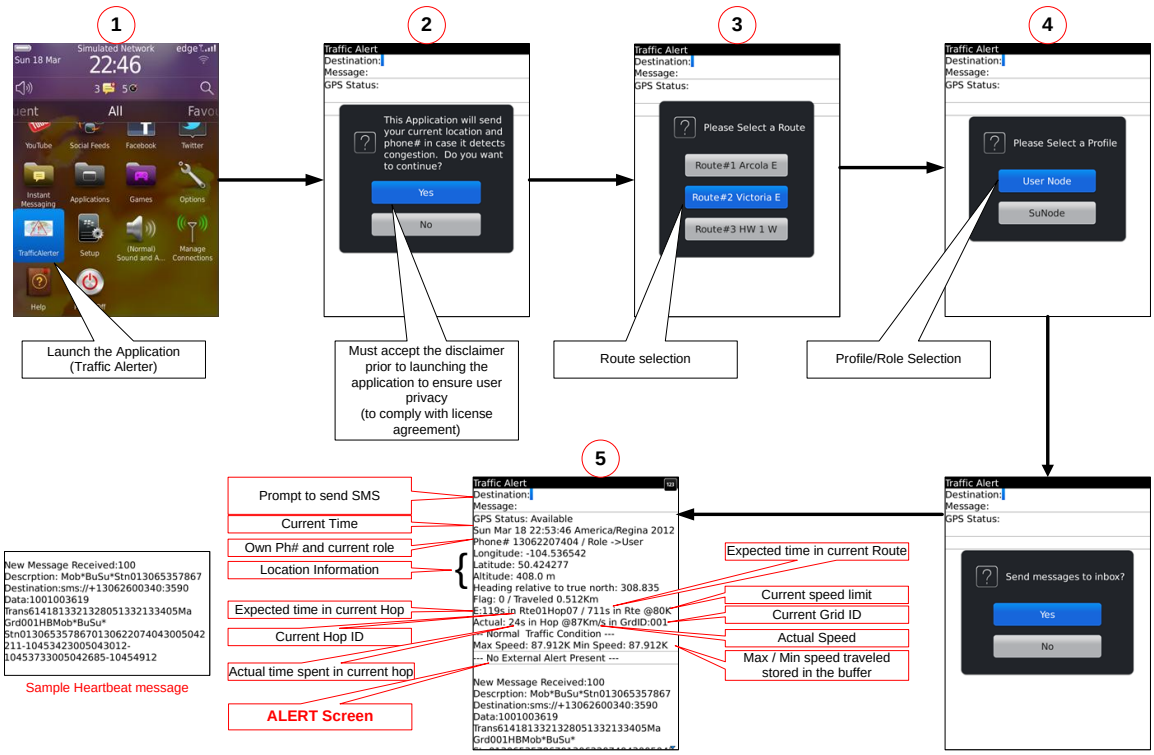

Figure 5.2 Screenshots of steps of application lanch

Congestion timer 1 is represented by flag 1, congestion timer 2 by flag 2, and congestion timer 3 by flag 3. Flag 4 signifies all 3 congestion timers have expired. Flag 0 indicates normal traffic and flag -1 denotes speeding. The corresponding status messages are: "*** Speeding!!! Please SLOW DOWN ***" for flag -1; "--- Normal Traffic Condition ---" for Flag 0 and 1; and "*** CONGESTION DETECTED ***" for Flag 3 and 4.

Test Performed on Simulated Environment

The simulation was setup using three simulated phones. The phones communicated with each other via the SMS server that was developed. The logical view of the setup is shown in Figure 5.3. Each device was preregistered with the server (each sent a dummy message so that the server could learn their phone and port numbers). Each phone travelled the same route as shown in Figure 5.4. *Micro grid* 1 started at hop Rte02Hop06 with a speed of 20 km/h, and *micro grid* 2 started at hop Rte02Hop05 slightly later than *micro grid* 1 with a speed of 30km/h. Finally, *Su micro grid* started at hop Rte02Hop01 with a speed of 50 km/h when *micro grid* 1 was waiting for congestion timer 3 to expire and *micro grid* 2 was waiting for congestion timer 1 to expire.

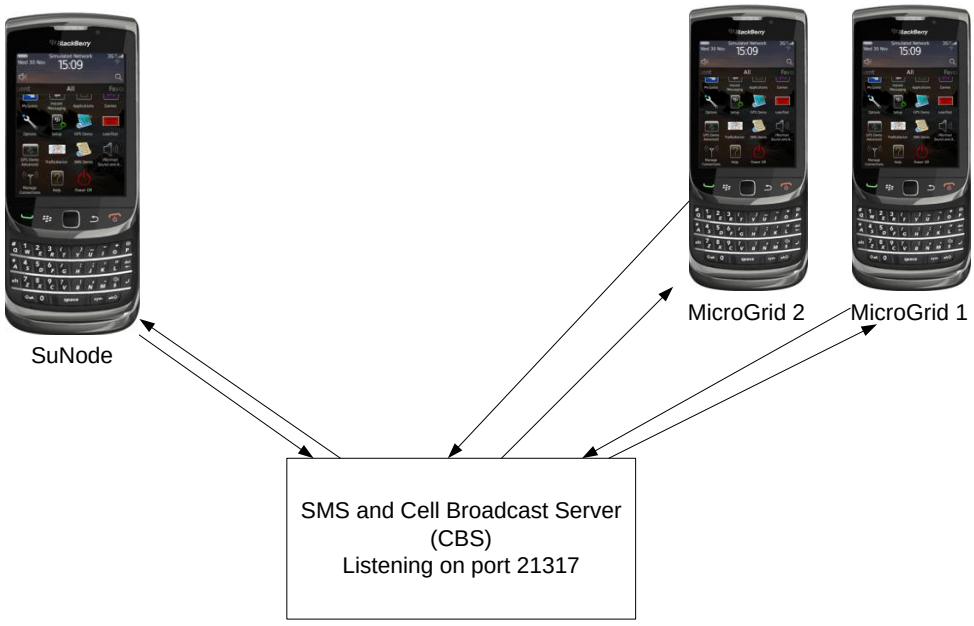

Figure 5.3 Simulation environment

Shortly after *Su micro grid* started, the congestion timer 3 of *micro gird* 1 expired and sent a "congestion detect" message to the *Su* node. Upon receiving this "congestion detect" message the *Su* node processed it and then broadcasted it. When this message

was received by the *micro grid* 2, it recognized that it was in a congestion state as well and that its location was further than *micro gird* 1. As a result, *micro grid* 2 sent another *congestion detect* message to the *Su micro grid* warning that the congested location was actually closer than previously sent. The *micro grid* 2 also updated its alert stating 'congestion is ahead' and the distance between itself and the troubled location reported by *micro grid* 1. This distance was updated every second to warn the driver to slow down to 20km/h, until *micro grid* 2 pasted the first reported congestion location. After it passed the congested location the screen displayed a message stating that it had passed the congestion and removed the alert from the display.

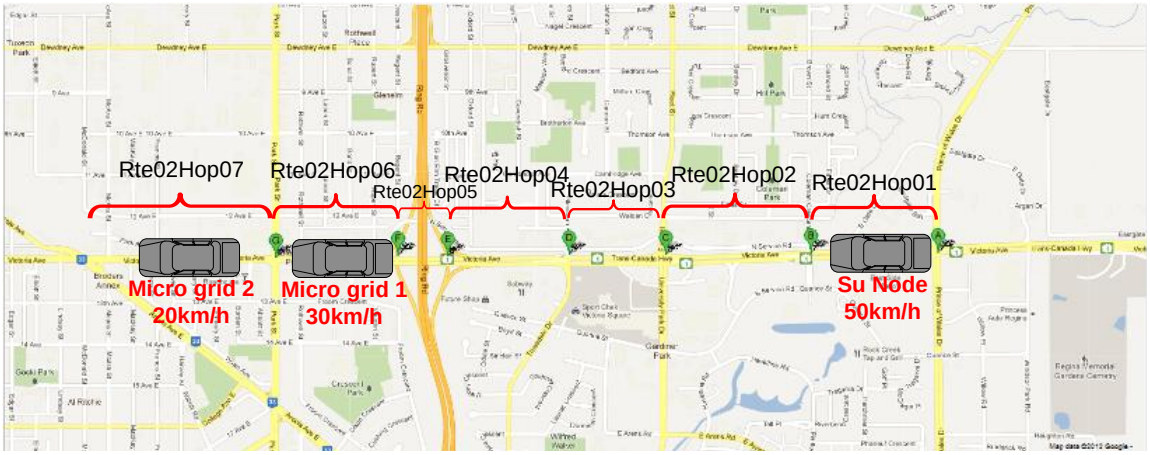

Figure 5.4 Map for simulation test scenario

Upon receiving this second congestion message, the *Su micro grid* broadcasted it to the other *micro grids*. The *Su* also updated its own warning screen. The previous warning was to slow down to approximately 20 km/h at a further distance, whereas, the latest message was for approximate speed of 30km/h at a closer distance. The, first warning for the *Su micro grid* was replaced so quickly that it was not noticeable to the driver, but

message traces proved that the proper messages were sent and they were visible when the captured video was played in slow motion.

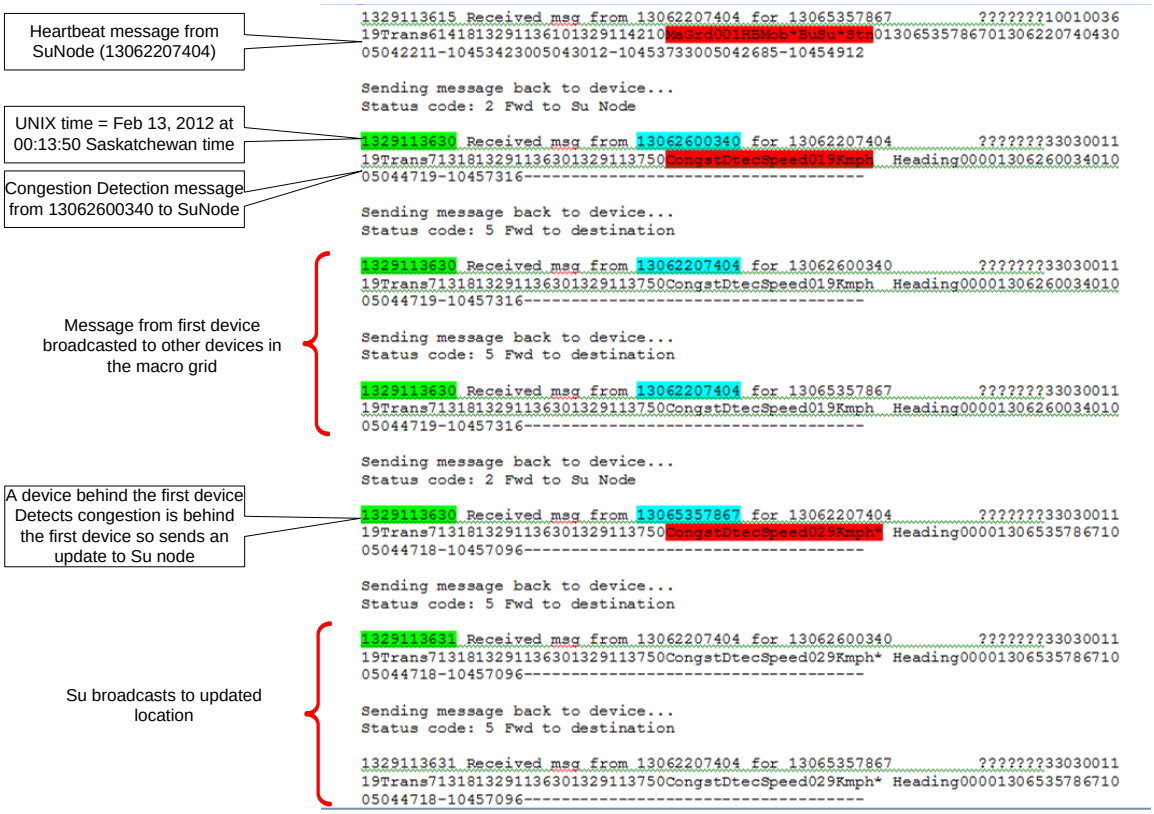

Figure 5.5 Congestion messaages during simulation

When the second *congestion detect* message reached *micro grid* 1 it displayed a message stating that it had received the *congestion detect* message behind it. The display then cleared the alert message.

The entire simulation was captured on video using "Microsoft Expression Encoder 4 Screen Capture software" for later analysis and proof. A summary of the test result is provided in Table 2-1. Some of congestion related messages are shown in Figure 5.5.

Table 5-1 Congestion timer trigger times and corresponding location (simulation)

Phone#	Flag #	Hop#	Time	Latitude	Longitude	Speed	Distancce travelled
1	Start	Rte02Hop01	00:13:31	50.447029	-104.534993	49.9	0
1	0	Rte02Hop01	00:13:31	50.447029	-104.534993	49.9	0
1	1	Rte02Hop01	00:13:32	50.447025	-104.535189	49.9	0.027
1	2	Rte02Hop02	00:14:36	50.447067	-104.547741	49.9	0.735
1	3	Rte02Hop	N/A	N/A	N/A	49.9	N/A
1	4	Rte02Hop03	00:14:48	50.447079	-104.550095	49.9	0.874
2	Start	Rte02Hop05	00:11:30	50.44714	-104.562185	19.98	0
2	0	Rte02Hop05	00:11:30	50.44714	-104.562185	19.98	0
2	1	Rte02Hop05	00:12:30	50.447164	-104.566892	19.98	0.227
2	2	Rte02Hop05	00:12:35	50.447166	-104.567284	19.98	0.299
2	3	Rte02Hop06	00:12:47	50.44717	-104.568226	19.98	0.355
2	4	Rte02Hop07	00:13:51	50.447196	-104.573247	19.98	0.643
3	Start	Rte02Hop04	00:11:47	50.447111	-104.556494	29.98	0
3	0	Rte02Hop04	00:11:47	50.447111	-104.556494	29.98	0
3	1	Rte02Hop	00:11:48	50.44712	-104.556612	29.98	0.016
3	2	Rte02Hop05	00:12:51	50.447149	-104.564026	29.98	0.449
3	3	Rte02Hop06	00:13:04	50.447157	-104.565556	29.98	0.541
3	4	Rte02Hop	00:14:07	50.447195	-104.572969	29.98	0.944

Test Performed Using Real Life Network

Site selection - The location selected for this test was close to the outskirts of the city of Regina. It was selected because of its relatively high speed limit, the presence of a traffic light and a shoulder lane. The test was performed on a Sunday afternoon when traffic volume was relatively low. Most of the driving took place on the shoulder to ensure public safety. The screens of both phones were recorded on video for future analysis and proof. Due to safety concerns only two vehicles were used for the drive tests. The map of the drive test is shown in Figure 5.6.

Micro grid 1 (vehicle 1) started at the beginning of hop Rte01Hop07 and *Su micro grid* (vehicle 2) started about 0.5 km behind it. The start times of the two vehicles were approximately 1.5 minutes apart. Both vehicles started by driving at the speed limit or

higher to ensure all the flags and timers were reset prior to simulating the congestion. Shortly after the vehicles reached the speed limit and recognized the hop they were in, both were slowed down to a speed between 20 and 40 km/h. *Micro grid* 1 detected the congestion first, and sent a congestion detection message to *Su micro grid*. The *Su micro grid* then broadcasted the message to three phones (one of them was *micro grid* 1 and one was itself).

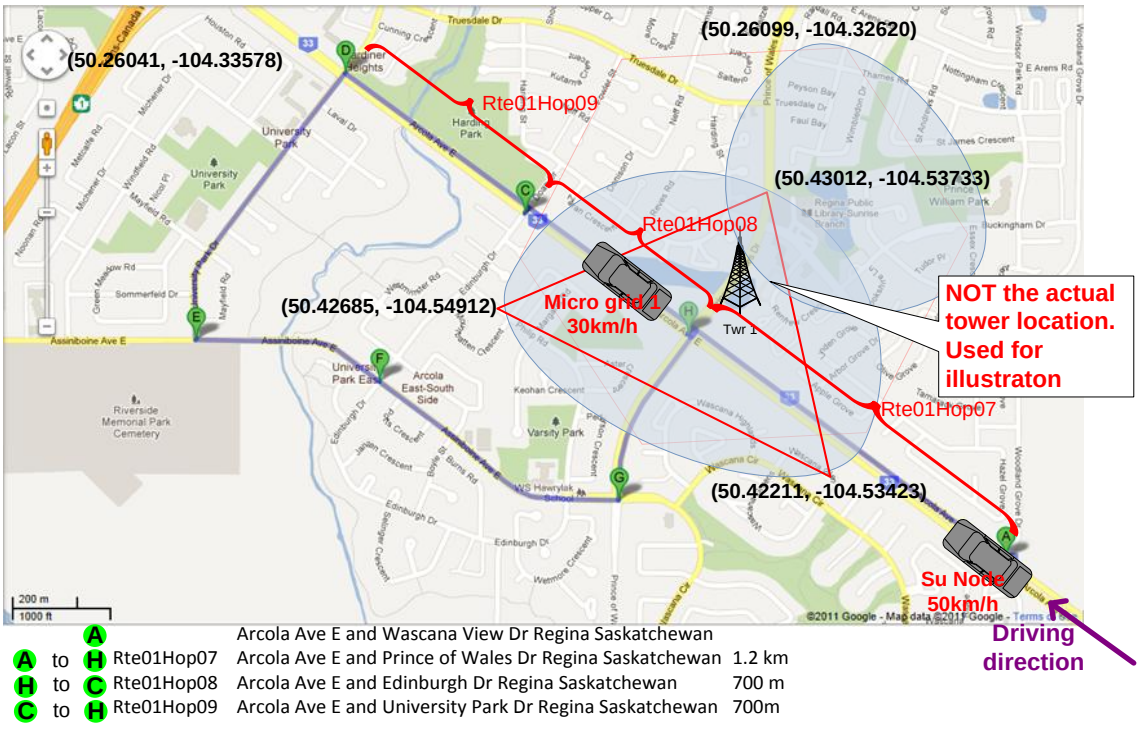

Figure 5.6 Map of drive test scenario

In addition to broadcasting, the *Su micro grid* also alerted the user stating that congestion was ahead. It showed the distance from its current location and the reported congestion location, updating it every second until the reported location was passed. It also confirmed (with a message on the screen) that the congested area was passed, and cleared the alert screen.

The same test was then repeated with slightly different scenario with the same setup. The only difference was that the *Su micro grid* was actually ahead of the slower vehicle when it received the congestion detection message. The *Su micro grid* showed that it had received the *congestion detect* message, but also confirmed that is was behind it so it cleared the alert.

It was observed that the application took 12 seconds longer to initialize and to display the first location information in the real phone compared to simulated phone. The delay in the display was expected and was not significant enough to cause real concern towards the usefulness of this application. According to the network capture tool there was a delay of about 2 seconds for the SMS to get delivered. Otherwise, the application was able to perform all the expected functions. The test results are summarized in Table 5-2.

Table 5-2 Congestion timer trigger times and corresponding location (live traffic)

Phone#	Flag #	Hop#	Time	Latitude	Longitude	Speed (Km/h)	Distance travelled
1 (Su)	0	Rte02Hop07	15:39:25	50.422893	-104.533327	80	1.015
1 (Su)	1	Rte02Hop07	15:39:37	50.424331	-104.536309	56	1.259
1 (Su)		Rte02Hop08	15:40:39	Congestion message received			1.784
1 (Su)	2	Rte02Hop08	15:40:42	50.427447	-104.542754	21	1.802
1 (Su)	3	Not Applicable	This timer was bypassed as part of the algorithm, since it already received *congestion detect* message				
1 (Su)	4	Rte02Hop08	15:40:55	50.428015	-104.543947	39	1.901
2	0	Rte02Hop07	15:38:05	50.42113	-104.52976	75	0.658
2	1	Rte02Hop07	15:38:11	50.42171	-104.53091	68	0.758
2	2	Rte02Hop07	15:39:17	50.42543	-104.53855	28	1.372
2	3	Rte02Hop07	15:39:31	50.42599	-104.53974	19	1.463
2	4	Rte02Hop08	15:40:36	50.42806	-104.54394	29	1.569

Several other tests were performed with only one phone during different traffic conditions. The application was successful in detecting congestion during rush hour.

The application did not generate any incorrect congestion detection message during non-peak hour traffic. It proves the congestion detection algorithm is effective in detecting congestion.

Performance Test of the Capabilities of a Real Phones to Calculate Speed

Multiple other drive tests were done to determine the phones' ability to calculate speed as well as to find out the necessary amount of time for the device to synchronize with GPS satellites. These tests were performed using the AGPS enabled BlackBerry BB Torch phone and was also done with an autonomous GPS enabled CDMA phone (BlackBerry 9630 aka Tour). The AGPS enabled phone required shorter synchronization times compared to the autonomous GPS enabled phone, as expected. The speed accuracy tests were performed using a MAP application that came with the BlackBerry Torch phone. This test was performed in open spaces (i.e. highway), rocky mountains (near Banff), as well as in the Calgary downtown area where tall buildings are present.

Some of the tests were done on the same route with the phone detached from the cellular service and some with cellular service on. The tests determined relatively short synchronization times for both cases and the speed readings were also consistent. However, the speed was always offset by approximately 7 km/h. Some minor inconsistencies were found near certain road curvatures. These errors are negligible for the q-end and congestion detection problem.

5.3 Results and Data Analysis

This section describes different issues that were identified during the application development and testing as well as their related workarounds. It also discusses how the

required broadcast frequencies were derived. This section concludes with a comparison of how the proposed solution gains network resource usage efficiencies both in terms of the number of signalling messages used in a single message and the number of messages needed to detect q-end location.

Issues and Mitigation Recommendation

The first two issues mentioned here are related to BlackBerry simulator. The third and fourth issues are related to the simulated SMSC and the last issue is related to errors due to decimal precision of double numbers.

1 When a route included multiple sections, the phone simulators did not provide a smooth transition when the phone moved from one section to another. It generated spikes of extremely high speeds, which caused many problems when testing the application. However, when a phone was tested in real traffic using the same route, this was not an issue. To work around this simulator problem, a route was created with only two end points that was one long stretch instead of multiple shorter stretches.

2 It was discovered that if the GPS route was not started prior to the launch of the application in the phone simulator, the application would not get location information from the GPS route simulator. To work around this known problem [52], the GPS route was always started prior to running the application in the simulated environment. This was not an issue in the real life trial.

3 The simulated SMS server trimmed two characters in positions 141 and 142 if the message length was greater than 140 characters. It still sent the remaining characters to the phone and the phone displayed remaining characters (total of

148) flawlessly. In order to bypass this issue, two extra characters were initially added in positions 141 and 142. Later, when it was decided to test the application in real phones, the messages were shortened to 140 characters (so the same message could be used in both simulated and production environment). This was done by dropping a mandatory field of the CAPv1.2 message. However, this did not jeopardise the application validation, since all other required fields were present and the real phone did not have any issue with sending all 150 characters.

4 The simulated server showed a few extra characters in its own display that were not part of the original payload. However, it did not send those extra characters to the phone, so it did not have any effect on the validation tests.

5 Many of the numbers used in the calculations were of the type Double. This resulted in "Double precision errors" and was a significant problem when testing if a device was inside a hop or not. As a result, the two points used to determine the presence in hop, were converted to a rectangle by adding a number equivalent to 50 meter, hence the assumption of 50 meter wide road.

Analysis of Broadcast Frequency Calculations

An analysis has been made with the assumption that there will be a *micro grid* or a sensor about 250 meters apart. If traffic is back to back, a 250 meter span will hold approximately 45 average size sedans, 20 single Unit Trucks, or 10 multi unit trucks [53]. On a good, dry, sealed road surface with a 75% efficient brake, a vehicle requires about 90 meters to stop when driving at a speed of 110km/h, making 250 meters sufficient stopping distance during both good and bad weather conditions [54]. It is also assumed that during poor weather conditions, people will drive more cautiously.

If traffic issues are broadcasted every 15 seconds and a vehicle is driving at 120 Km/h, the vehicle will travel about 500 meters in 15 seconds. The worst case error due to sensor position will be 249 meters (assuming the sensors are functioning properly) which leaves 251 meters to stop a vehicle. 250 meter will be a reasonable distance when accounting for poor weather and vehicle conditions and minor sensor errors. However, it does not leave any margin for complete failures or absence of a sensor. Changing the report interval to 8 seconds provides 733 meters to react, which is more than adequate to account for poor weather conditions, poor vehicle conditions, minor sensor errors and at least one sensor failure. These intervals can be changed if the speed limit is different or if the system is implemented at a location where poor weather is not an issue. However, to be safe and to account for network delays, every traffic congestion message should be broadcasted every 8 seconds to account for delay in message delivery.

Bandwidth Efficiency Gain Comparison

This section compares proposed solution with other similar solutions. Google [30] has developed a map application for smart phones that collects traffic information and notifies the user of traffic conditions. Their application makes the phone determine its current speed and location, before sending the information to Google. Google's map application does not have the necessary intelligence to detect congestion, so Google's off-site equipment detects the congestion. This implies that the phone has to send regular traffic updates to Google. The frequency at which the application sends the location and speed information to Google is unknown. The fact that Google does the congestion detection in their off-site location means that if network fails, their solution will not work.

In [17], [18] and other publications Pattara-atokom developed a smart phone application that calculates cell dwell time and sends the dwell time along with GPS location, LAC, and cell identity to a off-site server, which determines congestion. Thus, this solution will not work in case of wireless network trouble. Additionally, their application sends GPS data every second, and sends cell dwell times every time the phone changes cell tower, which requires frequent network resource usage.

Both of the above solutions use large amount of network resources and signaling to send data to off-site equipment. Figure 5.7 shows the amount of signaling required to send a single simple data packet in a UMTS network. Similar messaging will be required to receive a data packet. Figure 5.8 shows the signaling requirement for sending a cell broadcast. UMTS packet core requires 53 signaling messages to send or receive a data packet to a single UE whereas, in the case of cell-broadcast the number of signalling messages required is only 8 regardless of the number of recipients in a nodeB [55]-[56].

A study was been done for this thesis to demonstrate the improvement in using the proposed solution in terms of signalling requirement. The study compares the applications made by Google and Pattara-atokom to the solution proposed by TrafficAlerter, the application developed in this thesis. In this study all the solutions are put in the same scenario to solve a problem. In this scenario, there is an accident on a busy highway, where the speed limit is 110 km/h. The highway road surface and the weather allows for good driving conditions. It is assumed that there is a constant flow of incoming traffic and that each vehicle is driving approximately 90 meters apart (in order to maintain 3 second stop rule) from the vehicle directly in front of it. The study was done based on the number of vehicles entering the highway on a five kilometer stretch for

the duration of time necessary for TrafficAlerter to detect congestion. It is assumed that the highway is twinned (i.e. 2 lanes per direction) but that all vehicles analyzed are in the same lane and travelling in the same direction.

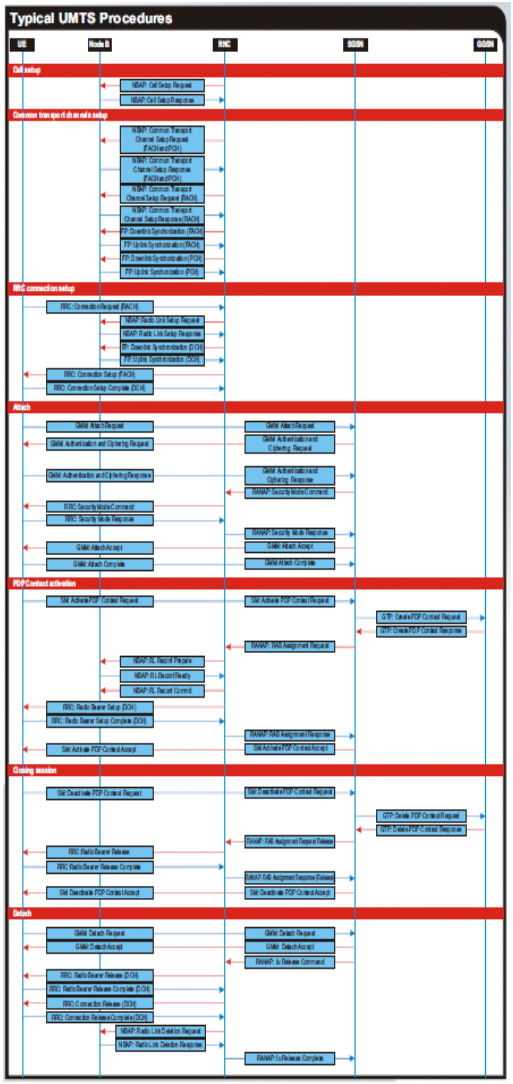

Figure 5.7 Typical signalling flow for packet data in the UTS network [55]

Since the frequency of sending location and speed information is unknown, in the Google map application, it is assumed that each phone will send one update and receive one update during this time of assessment. For Pattara-atokom's application, it is assumed each phone will receive a congestion notification via the UMTS data network. For the

TrafficAlerter application, it is assumed that the threshold distance for congestion detection is 50 meters, meaning that if a phone receives a cell broadcast with a troubled location, it will not send a response to the *macro grid manager* unless it is 50 meters away from the reported location. The *hop* and *macro grid* diameter are 10 kilometers long, the entire hop fits in the *macro grid*, and there is no end of hop delay (i.e. no traffic signal or stop sign at the end of hop). It is also assumed that the accident happened in the middle of the hop, which would be approximately five kilometers from the end. Based on the length of the test portion of the highway, road and weather conditions, speed limits, 3 second stopping rule and a constant flow of traffic there will be approximately 50 vehicles (mostly sedans) in the *macro grid* at the time of the accident. New vehicles will enter the *macro grid* every 3 seconds. The maximum network delay to send or receive any type of message (IP packet or SMS) is assumed to be 250 milliseconds both ways. All vehicles are equipped with smart phones running the application, all devices will participate in the test, and all have trusted profiles. It is also assumed that all three applications will use the IP network for sending traffic data. Both Google's and Pattara-atokom's applications will use the UMTS IP network for downlink data, whereas TrafficAlerter will use SMS-CB for downlink communication as per design.

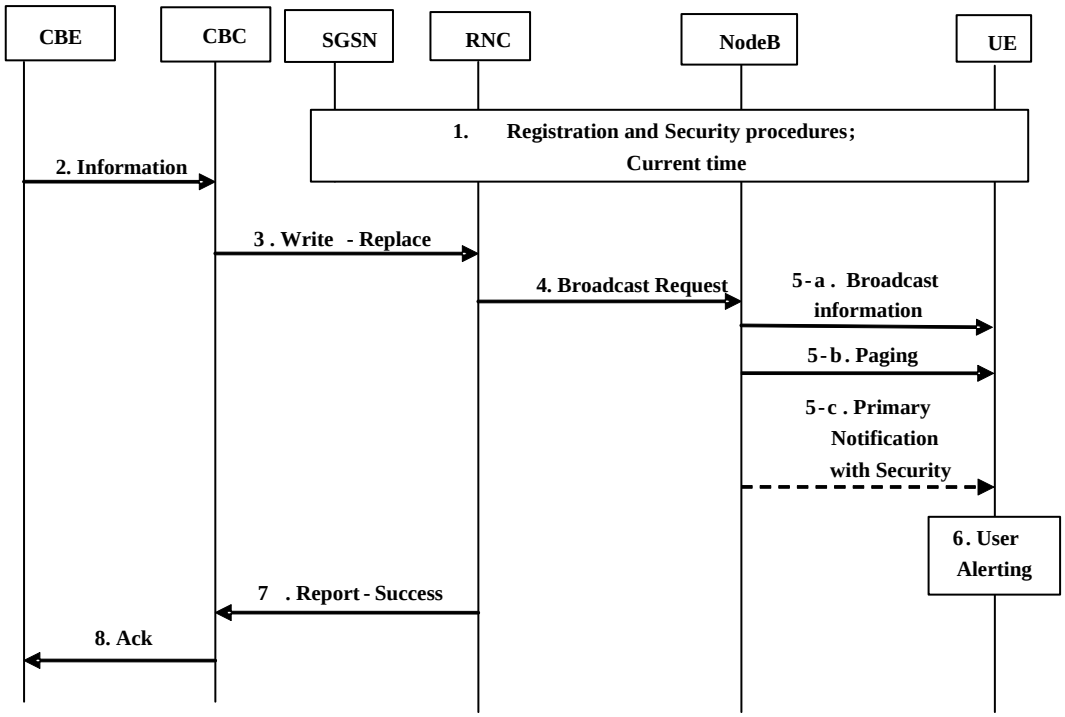

Figure 5.8 Signalling message flow for cell broadcast message in UMTS network [56][2]

Based on the above assumption, the value of first and third timers of the TrafficAlerter will be 27.7 seconds. That is 20.7 seconds to accelerate (5.31 km/h/s) from 0 to 110 km/h, plus 5.5 seconds to stop (decelerate) (20.16km/h/s) from 110 to 0 km/p, plus 1.5 seconds of reaction time, plus a 0 second end of hop delay. The total time between the time of accident and expiry of all 3 timers is about 77 seconds, which means 25 new vehicles will enter the *macro grid* before the timer expires. At the end of 77 seconds, there will be a total of 75 vehicles in the test *macro grid*. Figure 5.9 presents a graphical view of the test scenario.

[2] "Network registration and security process" (step 1) is performed each time a UE is attached to a network (e.g. after each power-on). Therefore this step will not take place during each SMS-CB.

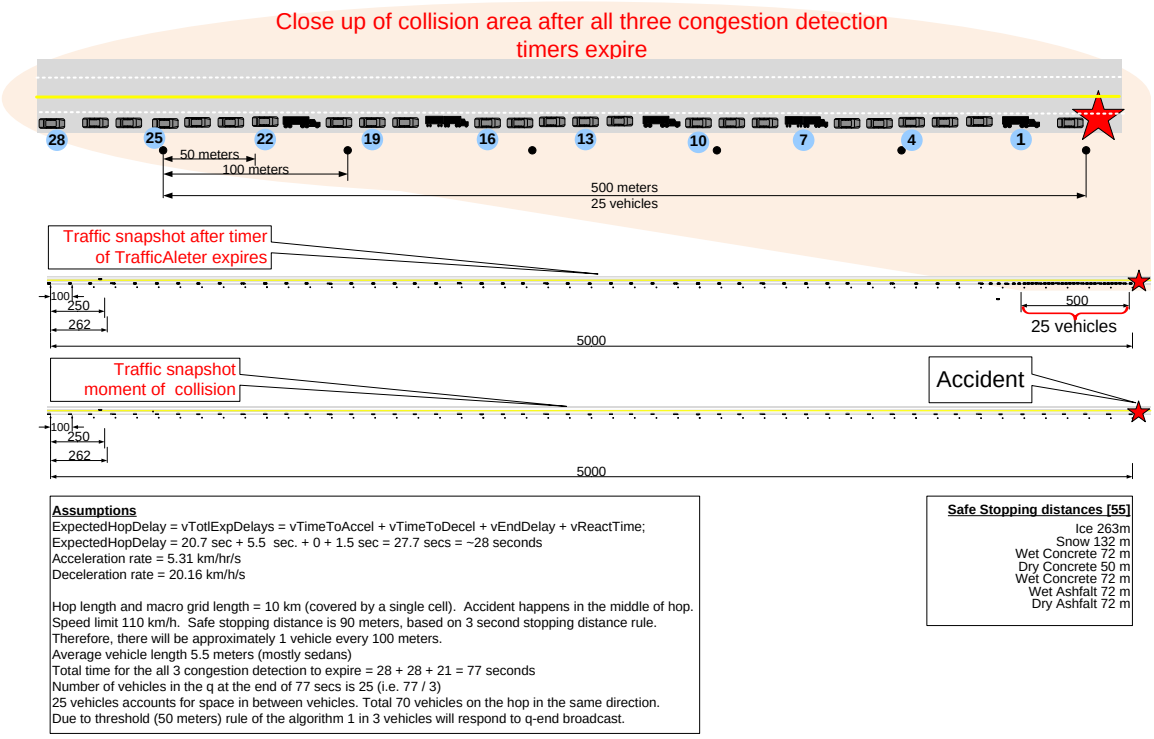

Figure 5.9 Network signalling traffic demonstratrion test scenario

At the end of 77 seconds, the vehicle 1 will send a message to the *macro grid manager*, which will broadcast the message to all the 75 vehicles in the *macro grid*. At the 78th second the vehicles numbered 4, 7, 10, 13, 16, 19, 22, 25 and 28 will send a message back to the *macro grid manager*. The *macro grid manager* will evaluate all the incoming messages and send an updated message containing the location and speed of vehicle number 28, since it represents the real q-end at the 78th second. Following that every 9 seconds or more (incoming vehicles will receive the warning and slow down so it will take longer than 9 seconds), there will be a message sent to the *macro grid* with the updated location until the traffic resumes movement. This will be due to the fact that q-end location will grow as new vehicles come closer to the q-end. Eventually vehicles may take alternate routes or part of the road will open for traffic.

Table 5-3 shows a comparison of how much signalling will take place for each of the applications during this 80 second period. As shown, the proposed solution uses significantly less network resources, both in terms of the number of messages sent and received as well as the signalling resources used.

Table 5-3 Comparison of message usage by different applications to solve the same issue

Application	Number of devices	Number of senders	Number of receivers	Maximum Number of Messages		Total Messages		Number of Signaling Messages for 1 message		Total Signalling Messages used		
				Sent by 1	Received by 1	Sent	Received	Uplink	Down link	To send	To receive	Total
Google	75	75	75	1	1	75	75	53	53	3975	3975	7950
TrafficAlerter	75	10	75	2	1	10	(2)150[3]	53	8	530	**16**	**546**
Pattara-atokom	75	75	75	77	1	5850	75	53	53	306075	3975	310050

In the case of Google`s application, 75 phones will send 1 message each and receive 1 message each. In the case of Pattara-atokom's application 75 phones will send 77 messages each and will each receive 1 message. Both Google`s and Pattara-atokom's applications will require 53 signalling messages to send or receive a message. For TrafficAlerter, only ten phones will send a single message and all phones will receive 2 messages requiring a total of 546 signalling messages. All the received messages by TrafficAlerter will only require eight signalling messages per message sent. Therefore, the proposed solution will provide a significant improvement over existing solutions.

[3] In reality only 2 messages are sent, but received by 70 phones

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

The main objective of this thesis is to prevent accidents by notifying drivers of vehicular queue ends. The second objective is to ensure the the service is available when it is most needed, regardless of cellular network conditions.

Existing technologies and research was analysed and found to be insufficient in providing uninterrupted warning messages. Currently, there are many solutions to provide live traffic information over the internet as well as over the mobile network. There are many ways to find location, detect congestion or q-ends, and warn people of these findings. The traffic information collection, problem detection methods, and delivery of the detected traffic issues vary from one proposed solution to another. Many of these solutions rely heavily on installed infrastructure, such as sensors, cameras, or the telecommunication network. The two main problems with these dependencies are a) during a disaster, both the transportation and the telecommunication infrastructure can get congested to a point where they are no longer functional; and b) road hazard detection infrastructure is available in only limited locations.

The solution provided in this thesis addresses both of these issues. The system utilizes millions of smart phones to work in detecting road hazards and to retrieve/generate warning messages using limited radio resources. It will use Cell Broadcast technology (also known as SMS-CB), a one-to-many geo-specific technology for most of the downstream communication and SMS or internet protocol for upstream communication. SMS-CB uses dedicated radio channels that are not used for carrying user plane (voice or data) or control plane messages. The CBS is not even part of the cellular network. It is an external entity that instructs these dedicated radio resources to send messages.

Therefore, during a disaster, important warning and safety instructions can still reach people who need it. Millions of smart devices will fill in the gaps of sensor networks and the SMS-CB will address the communication gap.

Although SMS uses only the signalling portion of a network, the upstream communication of sending up-to-date information remains a challenge during a disaster, especially if the tower site itself is damaged or if the SMSC itself is overloaded. To address this, the middleware is designed so that the system will work when the cellular network is operational, and when the cellular network is congested, or is out of order. The proposed solution supports a standalone operation, which means that a *macro grid* can function independently when the cellular network is out of order. In this mode, capable smart devices will use alternate licensed or unlicensed wireless technologies to communicate with neighbouring devices in the *macro grid* to share traffic information. The collaborative decision will then be passed to every *micro grid* including the one that have an alternate path to the *core,* which has a connection to the CBS in order to deliver messages to a non-participating device. During a crisis, it is assumed that there will be many emergency vehicles (i.e. police, fire trucks, ambulances) which will be equipped with satellite based phones and internet services that will have access to an alternate network connection. It is also possible to add priority dialing for these devices, which is supported by all current cellular systems.

The middleware was built with limited functionality for proof-of-concept. It was found that smart phones can independently detect congestion and can collaborate with each other to find the q-end location. The middleware was tested in both a simulated environment, as well as in a real UMTS network using different models of BlackBerry

phones. Although the solution proposes the use of cell broadcast technology, SMS was used instead. The same Java package provides the API for both SMS and cell broadcast to many Java based phone operating systems. The message structure developed for CBS application is also supported by SMS. Therefore, the tests results are valid for both technologies.

The solution supports different types of q-end situations. Q-end situations have been broken into the following five scenarios: (a) predictable non-mobile permanent obstruction, (b) predictable non-mobile temporary obstruction, (c) predictable time sensitive obstruction, (d) somewhat predictable mobile obstruction, and (e) unpredictable unknown obstruction. Recommendations have been made regarding when to use smart phones and when to use fixed sensors for congestion detection for each of these scenarios.

The middleware provides different services to different user types. It provides traffic and road hazard information to drivers and can also provide alternate route information via SMS-CB, so that the public can be directed to safe locations in times of a crisis. The middleware allows traffic violator information be sent to law enforcement agencies. It also provides traffic statistics to different road and infrastructure planning groups so they are able to identify what changes are required to the infrastructure.

During field tests, several limitations of the BlackBerry device simulator were found. The GPS route player of the simulator did not provide a smooth transition in terms of speed when multiple hops were used. Although the Blackberry device simulators support GPS route creation using multiple points and assignment of different speeds to each

segment. When the route was activated, it simulated the speed to be close to what was predicted. However, when the device transitioned from one segment to another, it created a spike of very high speed (i.e. above 800km/h) for an extremely short time, which seem like it was trying to catch up to the location of the new segment at a set time. Also, the simulator requires the routes to be played prior to running the application.

The BlackBerry phones with Assisted GPS are able to detect traffic speed quite accurately and are able to synchronise with a GPS satellite relatively quickly with or without the help of the cellular network. It took longer (approximately 12 seconds) to retrieve and launch the application in a real phone compared to the simulated phones in a simulated environment. This is possibly due to the time it takes to be synchronized with the satellites. There was a constant offset of about three to seven km/h on the speed depending on the phone models tested. The location and distance calculations were relatively accurate in most road situations. Some minor discrepancies were observed in roads where there were curvatures or near certain mountainous areas.

The smart UEs are capable of decoding different messages received and extracting the location and speed information. The application was able to deliver accurate warnings to people about traffic conditions. The system was also able to calculate the distance between itself and the reported trouble distance and provide drivers with an updated distance every second. The *micro grids* were able to collaborate with each other to find the actual q-end location. They corrected each other in terms of the location and speed of the congested area.

In order to comply with OASIS Common Alert Protocol v1.2, a mapping scheme was developed as part of the solution. CAPv1.2 ensures international compatibility in terms of message content and message structure. The CBS addresses the multi-language barrier issue. As a result, the solution will work for both inbound and outbound national and international roamers. In order to be able to direct people to safe locations, a format was developed that can send driving directions using 82 byte cell broadcast messages.

Finally, network latencies were measured for SMS messages. An analysis was made to determine the frequency of message broadcast for effective warning of q-end location. Another analysis was made to show how the proposed solution will make efficient use of network resources. The proposed solution will use less messages to detect congestion and will also use significantly less signalling resources. According to this analysis, the amount of uplink messages required was dropped to 13.33% compared to Google solution and 0.17% compared to Pattara-atokom's solution. The downlink message usage dropped to 2.66% when compared with both solutions. The total number of signalling message requirement dropped to 6.87% and 0.18% respectively.

6.1 Future Work

This section provides recommendations for future development. This thesis presented several congestion detection methods. However, only one method was completely developed and tested using a single major platform (i.e. the BlackBerry operating system). Only parts of other congestion detection algorithms were built and tested, but were not used in congestion detection.

The first recommendation is to further develop other congestion detection methods and test them in the same environment for comparison purposes. Secondly, congestion detection mechanisms should be enhanced to accommodate more than one congested spots in a route for both directions. This should also include a congestion clear notification.

Thirdly, two operational modes are proposed for *macro grid* operation but only one mode was developed and tested. More development should be made for standalone operations and then a transition mechanism between the two methods should be tested. This extension should also include security enhancements.

Fourthly, continued development of this application on an Android platform is recommended, since several smart phone manufacturers are adapting the Android platform. This development should extend to tablets as well. Also, there is a need to run traffic capture applications in a Google phone to identify the frequency and amount of data sent or received by the map application, provided it does not infringe on licensing agreements in order to compare the two solutions. Finally, more road tests need to be performed using more phones.

REFERENCES

[1] Public safety, at: http://en.wikipedia.org/wiki/Public_safety, accessed April 2012

[2] Brian O'Shaughnessy, VP, Wireless Technology Bell Mobility, "Public Safety & Disaster Relief - A Wireless Network Operators Perspective", http://www.rabc.ottawa.on.ca/e/Files/020327_PPDR_Bell_.ppt, accessed December 23, 2007

[3] MESA project, at - www.projectmesa.org, accessed April 7, 2012

[4] "Advanced warning of stopped traffic on freeways: current practices and field studies of queue propagation speeds", Report 4413-1, Texas Transportation Institute, College Station, TX, USA - Poonam B. Wiles, Scott A. Cooner, Edward J. Pultorak, Yatin Rathod, and Diana G. Wallace

[5] Risk Management News, Vol. 1, Issue 2, National Electrical Contractors Association (NECA) as sited by in power presentation by www.osha.gov/dcsp/alliances/.../motor_vehicle_safety.ppt on slide #11, accessed December 04, 2011

[6] "The Emergency Alert System (EAS) and All-Hazard Warnings", Linda K. Moore, Congressional Research Service, August 26, 2010

[7] "Common Alerting Protocol Version 1.2", OASIS Standard, 01 July 2010 http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2.pdf

[8] F. H. Khan, Z. Khan, "A Systematic Approach for Developing Mobile Information System based on Location Based Services", Macrothink Institute™ , Network Protocols and Algorithms, ISSN 1943-3581, 2010, Vol 2, Issue 2

[9] “Mastering UMTS Radio Networks & Signaling (R99)”, Awards Solutions Inc., 2008.

[10] Guillaume Leduc, "Road Traffic Data: Collection Methods and Applications", JRC Technical Notes, Working Papers on Energy, Transport and Climate Change N.1, JRCJRC 47967 - 2008

[11] Peter T. Martin, Yuqi Feng, and Xiaodong Wang, “Detector Technology evolution,” Traffic Lab, University of Utah, Technical Report, November 2003

[12] Project aimed at avoiding traffic, Updated 6/13/2008 11:42 AM,By Larry Copeland, USA TODAY - http://www.usatoday.com/news/nation/2008-06-12-I95-traffic-side_N.htm - Nov 06, 2010

[13] “A Wireless Mesh Network Platform for Vehicle Positioning and Location Tracking”, Mohamed El-Darieby, Hazem Ahmed, Mahmoud Halfawy, Ahmed Amer, Baher Abdulhai

[14] Sananmongkhonchai, S.; Tangamchit, P.; Pongpaibool, P.; , "Cell-based traffic estimation from multiple GPS-equipped cars," *TENCON 2009 - 2009 IEEE Region 10 Conference* , vol., no., pp.1-6, 23-26 Jan. 2009

[15] Huasong Cao; Hui, R.; Leung, V.C.M.; , "Cell link: Real-time data tracking of automobiles via cell phones," *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on* , vol., no., pp.305-306, 9-13 Jan. 2010

[16] Junqiang Guo; Fasheng Liu; Chengjiang Li; Zhiqiang Zhu; , "Using GSM Technologies to Collect and Supply Expressway Traffic Information," *Wireless*

Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on , vol., no., pp.1-4, 24-26 Sept. 2009

[17] Hongsakham, W.; Pattara-atikom, W.; Peachavanish, R.; , "Estimating road traffic congestion from cellular handoff information using cell-based neural networks and K-means clustering," *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on* , vol.1, no., pp.13-16, 14-17 May 2008

[18] Poolsawat, A.; Pattara-Atikom, W.; Ngamwongwattana, B.; , "Acquiring road traffic information through mobile phones," *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on* , vol., no., pp.170-174, 24-24 Oct. 2008

[19] Guo Li mei; Luo Da yong; , "Apply Cellular Wireless Location Technologies to Traffic Information Gathering," *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on* , vol.3, no., pp.499-502, 10-11 Oct. 2009

[20] Chenyi Chen; Jianming Hu; Yin Wang; , "Cell-based simulation and estimation of urban traffic network," *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* , vol., no., pp.7-12, 19-22 Sept. 2010

[21] Staňková, K.; De Schutter, B.; , "On freeway traffic density estimation for a jump Markov linear model based on Daganzo's cell transmission model," *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* , vol., no., pp.13-18, 19-22 Sept. 2010

[22] Bauza, R.; Gozalvez, J.; Sanchez-Soriano, J.; , "Road traffic congestion detection through cooperative Vehicle-to-Vehicle communications," *Local Computer*

Networks (LCN), 2010 IEEE 35th Conference on , vol., no., pp.606-612, 10-14 Oct. 2010

[23] Where does NAVTEQ Traffic.com get traffic information?, http://bhelp.traffic.com/where-does-trafficcom-get-traffic-information, accessed October 13, 2011

[24] NAVTEQ Maps and Traffic, at: http://www.navteq.com/, accessed April 2012

[25] TrafficLand.com Traffic Cameras, Traffic Video, Live Traffic Cams, at: http://www.trafficland.com/, accessed October 2011

[26] We're the traffic experts, so you don't have to be, at: http://www.inrix.com/trafficinformation.asp, accessed November 2010

[27] TrafficSense - Road Traffic Monitoring and Traffic Information Services, at: http://www.cellint.com/traffic_data/traffic_system.html, accessed November 2010

[28] Testing Cellular-Based Traffic Data Collection Technologies - Performance Required for Road Management and Traveler Information, at: http://www.cellint.com/traffic_data/how_to_test_data.html, accessed October 2011

[29] Government - IMS works with the governments, transportation authorities, and public sector organizations, to jointly develop Intelligent Transportation Systems, at: http://www.intellimec.com/ims-and-you/government/, accessed October 2011

[30] Dave Barth, The bright side of sitting in traffic: Crowdsourcing road congestion data, posted on 8/25/2009 08:16:00 AM, at: http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html, accessed October 17, 2011

[31] Daniel Terdiman, Staff Writer, Google Maps adds real-time traffic data, CNET News, February 28, 2007 3:30 PM PST at: http://news.cnet.com/Google-Maps-adds-real-time-traffic-data/2100-1032_3-6163203.html, accessed Oct 17, 2011

[32] Khan, A.M.; , "Intelligent infrastructure-based queue-end warning system for avoiding rear impacts," *Intelligent Transport Systems, IET* , vol.1, no.2, pp.138-143, June 2007

[33] Birk, W.; Eliasson, J.; Lindgren, P.; Osipov, E.; Riliskis, L.; , "Road Surface Networks technology enablers for enhanced ITS," *Vehicular Networking Conference (VNC), 2010 IEEE* , vol., no., pp.152-159, 13-15 Dec. 2010

[34] Intrado to Offer Life-Saving Mobile Alerting System to Wireless Service Providers, LONGMONT, CO, Oct. 12, 2010 , Press release. http://www.intrado.com/news/archives/10102010.asp - March 07, 2011 at 22:35

[35] "Cell Broadcast Technology for Emergency Alert Notifications: Reach Many or Millions: It's About Time," CellCast Technologies, White Paper by Paul Klein, 2007, - http://www.fcc.gov/pshs/docs/advisory/cmsaac/pdf/CellCastComment070307.pdf - Last accessed Oct 18, 2011

[36] "Cell Broadcast in Public Warning Systems: Reaching Millions in a Matter of Seconds", issued by Cell Broadcast Forum, November 2005, .CBF-PUB(05)02R0.2

[37] "Cell Broadcast Technology for Emergency Alert Notifications, Reach Many or Millions. It's About Time," white paper Paul Klein, CellCast Technologies

[38] Sanders, P.; , "Displaying Cell Broadcast messages," *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on* , vol., no., pp.1-5, 3-7 Oct. 2011

[39] Wijesinghe, L.; Siriwardena, P.; Wijeratne, S.; Purasinghe, H.; Dias, D.; , "Disaster and Emergency Warning Network (DEWN): Harnessing Cellular Technologies for Early Warning Dissemination," *Global Humanitarian Technology Conference (GHTC), 2011 IEEE* , vol., no., pp.476-480, Oct. 30 2011-Nov. 1 2011

[40] Integrating Communications for enhanced environmental risk management and citizens safety (CHORIST), European Commission – 033685, at: http://www.chorist.eu/doc/CHORIST-SP3.D18-V1.1-PUBLIC.pdf , accessed March 07, 2011

[41] D. Gundlegård, "Automotive Telematics Services based on Cell Broadcast," M.S. Thesis, Dept. Sci. and Technology, Linköping Univ., Norrköping, Sweden, 2003

[42] Service, http://www.trafficsense.co.uk/services.htm, accessed October 13, 2011

[43] B. E. Rashid "Service-oriented wireless grid architecture," M.A.Sc. Thesis, Dept. Elect. Sys. Eng., Univ. Of Regina, Regina, SK, 2007

[44] El-Darieby, M.; Krishnamurthy, D.; , "A Scalable Wide-Area Grid Resource Management Framework," *Networking and Services, 2006. ICNS '06. International conference on* , vol., no., pp.76, 16-18 July 2006

[45] U.S. Environmental Protection Agency on average acceleration rate, at: http://www.epa.gov/otaq/regs/ld-hwy/ftp-rev/ftp-summ.txt,

[46] Consumer Braking Information Initiative performed by U.S. Army Aberdeen Test Center, on Fall 1998 on wet road condition with no payload, at: http://www.nhtsa.gov/cars/testing/brakes/, accessed November 22, 2011.

[47] What is the formula for calculating acceleration?, at: http://wiki.answers.com/Q/What_is_the_formula_for_calculating_acceleration#ixzz1eS2a1RFP, accessed October 2011

[48] Calculate distance, bearing and more between Latitude/Longitude points, at: http://www.movable-type.co.uk/scripts/latlong.html, accessed October 2011

[49] Listening CBS messages in application..., at: http://supportforums.blackberry.com/t5/Java-Development/Listening-CBS-messages-in-application/td-p/346258, accessed November 27, 2011

[50] Support for Java ME APIs, at: http://docs.blackberry.com/en/developers/deliverables/30946/Support_for_standard_Java_APIs_1777916_11.jsp, accessed April 07, 2012

[51] Wikipedia:WikiProject Geographical coordinates, http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Geographical_coordinates#Precision, accessed March 07, 2010

[52] LoacatioProvider (BlackBerry JDE 6.0.0 API Reference), at: http://www.blackberry.com/developers/docs/6.0.0api/javax/microedition/location/LocationProvider.html, accessed March 21, 2012

[53] Vehicle Lengths, at: http://www.dot.ca.gov/hq/traffops/trucks/trucksize/length.htm, accessed September 18, 2009

[54] Road Safety: Speeding - the facts, at: http://www.dtei.sa.gov.au/roadsafety/safer_speeds/speeding, accessed February 25, 2011

[55] Protocols.com, “The world of UMTS”, at: http://www.protocols.com/posters/UMTSposter.pdf, - accessed April 08, 2012

[56] 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; "Technical realization of Cell Broadcast Service (CBS)" (Release 9), 3GPP TS 23.041 V9.6.0 (2010-12).

BIBLIOGRAPHY

[1] R. Kreher, T. Rödebusch, *UMTS Signaling UMTS Interferences, Protocols, Message Flows and Procedures Analyzed and Explained*, Second Edition, John Wiley & Sons, Ltd, 2007

[2] "Vocabulary for 3GPP Specifications", ETSI TR 121 905 V9.4.0, 2010-01), (3GPP TR 21.905 version 9.4.0 Release 9)

[3] Wacker, A.; Laiho-Steffens, J.; Sipila, K.; Heiska, K.; , "The impact of the base station sectorisation on WCDMA radio network performance," *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th* , vol.5, no., pp.2611-2615 vol.5, 1999

[4] "ETSI TS 123 003 V9.5.0 (2011-01), Technical Specification, Digital cellular telecommunications system (Phase 2+);Universal Mobile Telecommunications System (UMTS);Numbering, addressing and identification (3GPP TS 23.003 version 9.5.0 Release 9)," GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS

[5] H. Holma, A. Toskala, *LTE for UMTS : OFDMA and SC-FDMA Based Radio Access*. Hoboken, NJ, Wiley, 2009.

[6] Radio access network, at: http://en.wikipedia.org/wiki/Radio_access_network, accessed April 10, 2011

[7] "Advantages and Services Using Cell Broadcast: Reaching Millions in a Matter of Seconds" , February 2002, issued by Cell Broadcast Forum, CBF-PUB(02)3R2.1

[8] "Handset Requirements Specification: Reaching Millions in a Matter of Seconds", CBF-PUB(02)2R2.4, October 2006, Issued by Cell Broadcast Forum

[9] Mark Wood, "CELL BROADCAST IN IS95 CDMA, FOR LOW COST WARN ACT COMPLIANCE," CellCast Technologies, 2009 - http://www.cellcastcorp.com/pdf/Cell%20Broadcast%20in%20CDMA%20for%20Low%20Cost%20Warn%20Act%20Compliance%20White%20Paper.pdf, accessed September 25, 2011

[10] LTE Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Overall description, (3GPP TS 36.300 version 10.4.0 Release 10), Technical Specification, Stage 2, ETSI TS 136 300 V10.4.0 (2011-06)

[11] W. 1. Franz, H. Hartenstein, B. Bochow, "Internet on the Road via Inter-Vehicle Communications," GI Jahrestagung (1) 2001: 577-584, 2001

[12] Seetha Ramanjaneyulu, B.; Gopinathan, E.; , "Analyzing the interference patterns of Bluetooth sensors in high-speed data acquisition systems," *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on* , vol., no., pp. 83a- 83f, 4-7 Jan. 2005

[13] Shuaib, K.; Boulmalf, M.; Sallabi, F.; Lakas, A.; , "Co-existence of Zigbee and WLAN - a performance study," *Wireless and Optical Communications Networks, 2006 IFIP International Conference on* , vol., no., pp.5 pp.-5, 0-0 0

[14] Xiangpeng Jing; Anandaraman, S.S.; Ergin, M.A.; Seskar, I.; Raychaudhuri, D.; , "Distributed Coordination Schemes for Multi-Radio Co-existence in Dense Spectrum Environments: An Experimental Study on the ORBIT Testbed," *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on* , vol., no., pp.1-10, 14-17 Oct. 2008

[15] Zeghdoud, M.; Cordier, P.; Terre, M.; , "Impact of Clear Channel Assessment Mode on the Performance of ZigBee Operating in a WiFi Environment," *Operator-Assisted (Wireless Mesh) Community Networks, 2006 1st Workshop on* , vol., no., pp.1-8, Sept. 2006

[16] Bertocco, M.; Gamba, G.; Sona, A.; , "Is CSMA/CA really efficient against interference in a wireless control system? An experimental answer," *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on* , vol., no., pp.885-892, 15-18 Sept. 2008

[17] Vanheel, F.; Verhaevert, J.; Moerman, I.; , "Study on Distance of Interference Sources on Wireless Sensor Network," *Microwave Conference, 2008. EuMC 2008. 38th European* , vol., no., pp.175-178, 27-31 Oct. 2008

[18] Eren, H.; Fadzil, E.; , "Technical Challenges for Wireless Instrument Networks - A Case Study with ZigBee," *Sensors Applications Symposium, 2007. SAS '07. IEEE* , vol., no., pp.1-6, 6-8 Feb. 2007

[19] Stopping (Braking) Distance Calculator, at: http://forensicdynamics.com/stopping-distance-calculator, Accessed Jan 15, 2011