CHAPTER I

INTRODUCTION

Vehicular communications (VC) lay at the core of a number of industry and academic research initiatives aiming to enhance safety and efficiency of transportation systems. Vehicles and road-side infrastructure units (RSUs), i.e., network nodes, will be equipped with on-board processing and wireless communication modules. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication will enable applications that provide warnings on environmental hazards (e.g., blocks on the pavement), as well as traffic and road conditions (emergency braking, congestion, or construction sites).

VC offer a rich set of tools to drivers and administrators of transportation systems but, at the same time, they make possible a formidable set of abuses and attacks. Nodes that 'contaminate' large portions of the vehicular network with false information, or the deployment of nodes that collect VC messages, track the location and transactions of vehicles and infer sensitive information about their drivers. Worse even, vehicles and their processing and sensing equipment can be physically compromised, while any wireless-enabled device could pose a threat to the VC system.

These simple examples of exploits indicate that under all circumstances VC systems must be secured. Otherwise antisocial and criminal behavior could be made easier, actually jeopardizing the benefits of the VC system deployment. A comprehensive set of security mechanisms is thus critical, and facilities and protocols that mitigate attacks are necessary. Securing vehicular communications is a hard problem, due to the tight coupling between applications and the networking fabric, as well as additional societal, legal, and economic considerations, which raise a unique combination of operational and security requirements.

There has been constant perusal in VANET community to establish alternate mechanisms instead of adding the overhead of encryption and decryption to individual nodes to reduce fault reduction.

The key factors that differentiate and in turn complicate vehicular communication are

- Epidemic Communication
- Ephemerality of Networks
- Mechanisms to read traffic conditions autonomously.

However, there are still some major limitations. Existing methods may not have the capacity to support vehicular communication barring all these under consideration. So instead of trying to add the overhead of encryption there has been repeated proposal for using alternate statistical inference tactics to support decision making in vehicular communication.

The traditional notion of trust as a relation among entities, while useful, becomes insufficient for emerging data-centric mobile ad hoc networks. In these systems, setting the data trust level equal to the trust level of the data providing entity would ignore system salient features, rendering applications ineffective and systems inflexible. This would be even more so if their operation is ephemeral, i.e., characterized by short-lived associations in volatile environments. The challenge has been addressed by extending the traditional notion of trust to data-centric trust: trustworthiness attributed to node reported data per se. A framework for data-centric trust establishment: First, trust in each individual piece of data is computed; then multiple, related but possibly contradictory, data are combined; finally, their validity is inferred by a decision component based on one of several evidence evaluation techniques has been established.

The entire project has been composed of three phases

- Architecture Deployment
- Data Centric Trust establishment
- Designing a VANET helper application for Personal Digital Assistant’s

1.1. ARCHITECTURE DEPLOYMENT

The first phase of the project is to deploy an efficient architecture for the VANET with detailed evaluation on the message protocol and Information sharing decision idea.

1.2. DATA–CENTRIC TRUST ESTABLISHMENT

The crucial phase of our project is to justify the transition from traditional trust establishment mechanisms to data–centric approach in which the decision function takes care of the position and ability of the reporting node to collect information about the situation.

1.3. PRACTICAL VANET UTILITY APPLICATION FOR MOBILES

The final phase of the project involves creating a VANET application to support sharing of information through Bluetooth as a preliminary sort of communication which can be further enhanced. The utility function need to be portable and flexible which can be plugged anywhere in java platform.

1.4. CHALLENGES

The key challenges associated with the project are

i. Message Integrity within the network.
ii. Message redundancy to avoid within the network.
iii. Accuracy of the Decision function to help Network.
iv. Portability of the mobile PDA device to support.
v. Extendibility to support various modes of communication.

The project concentrates on all these challenges to support a universal message sharing protocol to effectively study VANET protocols further. A message sharing protocol has been established which has been used throughout our evaluation mechanisms.

The wide range of consolidations that is drawn out of our simulation results need to be consistent enough to arrive at solid conclusion.

A set of security protocols has been showed that they protect privacy and their robustness has been analyzed, and a quantitative assessment of the proposed solution has been designed. A realistic simulation environment to study our ideas in the wake of problems in realistic simulation of the road network and its environment has been established.

1.5. ORGANISATION OF THE PROJECT

The report is organized as follows: Chapter II describes the Conceptual overview. Chapter III describes the Implementation details. Chapter IV describes the Results obtained and the Analysis of results. Chapter V concludes this project giving the scope for future enhancements.

CHAPTER II

CONCEPTUAL OVERVIEW

2.1 DATA – CENTRIC TRUST ESTABLISHMENT

In all traditional notions of trust, data trust (e.g., trust in the identity or access/attribute certificates) was based exclusively on a priori trust relations established with the network entities producing these data (e.g., certification authorities, network nodes) [2]. This was also the case when trust was derived via fairly lengthy interactions among nodes, as in reputation systems [1]. Moreover, any new data trust relationships that needed to be established required only trust in the entity that produced those data. All trust establishment logics proposed to date have been based on entities (e.g., “principals” such as nodes) making statements on data [4], [1]. Furthermore, traditional trust relations evolved generally slowly with time: once established, they lasted a long time and changed only after fairly lengthy operations (e.g., certificate revocation or monitoring and then voting-off of peers). These observations indicate that existing trust notions are entity-centric and slow to change. However, several emerging mobile networking systems are heavily, if not entirely, data-centric in their functionality and operate in ephemeral environments.

In such scenarios, it is more useful to establish trust in data rather than in the nodes reporting them. In vehicular networks, node identities are largely irrelevant; rather, safety warnings and traffic information updates, along with their time freshness and location relevance, are valuable. At the same time, interactions with data reporters do not rely on any prior association, and encounters are often short-lived, especially due to high mobility.

The traditional entity centric trust evaluation mechanism with the more dynamic data centric object oriented approach in which user gets much of the credit for sharing useful message depending upon the state of observation rather than usual ranking mechanisms has been evaluated..

In such scenarios, and unlike in traditional trust establishment schemes, the trust level associated with data is not the same as that of the node that generated the data. More specifically, in the vehicular networks example, vehicles will have

different preset node trust levels (e.g., police cars are more trustworthy than private vehicles), but

(i) Different events reported by the same vehicle may have different levels of trust (due to distance to the event, timeliness of the report, vehicle equipment level) that may differ from that of the vehicle itself.

(ii) The same event reported by multiple vehicles with different preset node trust levels has to be associated with a single trust level that would, of course, differ from some of the levels of the reporting vehicles

(iii) An event reported by a vehicle requires corroboration by other vehicles and hence its level of trust would differ from that of the reporting vehicle.

In other words, the following question arises naturally: how can these emerging systems be effective and trustworthy when their basic operational requirements are not satisfied by existing trust notions? To address this challenge, a clean-slate approach has been advocated.

A data-centric trust establishment: data trustworthiness should be attributed primarily to data per se, rather than being merely a reflection of the trust attributed to data-reporting entities.

The logic proposed, extends the traditional notions of trust and methods of trust establishment in several ways.

First, unlike traditional trust, a priori trust relationships in entities (nodes) represent only one of the default parameters for establishing data trust. For example, our logic, while using nodes' statements on data, does not rely exclusively on such statements. Instead, it takes into account dynamic factors, such as location and time, as well as the number and type of the statements on data, to derive data trust relations.

Second, beyond the traditional time-invariant or slow-evolving trust notions, data-centric trust relations are by definition ephemeral and have to be established and re-established frequently, based on network and perceived environment changes. For example, an event report (e.g., accident report, weather report) that must be believed by recipient nodes in real-time cannot last longer than the lifetime of the event or of the network that generated this event. Multiple rounds of node interactions are typically not possible in such networks.

Third, trust does not stem from a single source of data (e.g., a certification authority) and generally it is application-dependent (in contrast to entity-centric trust when, for example, multiple applications use certificates for their access control and authentication policies).

First trust in data (e.g., reported event) from multiple pieces of evidence (e.g., reports from multiple vehicles). Then, the logic weighs each individual piece of evidence according to well-established rules and takes into account various trust metrics, such as time freshness and location relevance, defined specifically in the context of an application. Then, data and their respective weights serve as inputs to decision logic that outputs the level of trust in these data.

Several techniques, including voting, Bayesian inference, and the Dempster-Shafer Theory of evidence has been evaluated.

Notably, Bayesian inference takes into account prior knowledge, whereas the Dempster-Shafer Theory accounts for the uncertainty about data. More specifically, while trust establishment mechanisms based on popular decision logics, such as voting and Bayesian inference, consider uncertainty as refute of evidence, our framework considers uncertainty as either supporting or refuting evidence, thus making the decision process more realistic.

This distinction affects the flexibility and resilience to attackers in some scenarios. First of all there would be a conceptual explanation of the VANET system to assess the separate evaluation of the separate components of the system.

2.2 GENERAL FRAMEWORK

2.2.1 VEHICULAR ADHOC NETWORKING:

A Vehicular Ad-Hoc Network, or VANET, is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is

created. It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes.

VANET offers countless benefits to organizations of any size.

Automobile high speed Internet access would transform the vehicle's on-board computer from a nifty gadget to an essential productivity tool, making virtually any web technology available in the car. While such a network does pose certain safety concerns (for example, one cannot safely type an email while driving), this does not limit VANET's potential as a productivity tool. It allows for "dead time"—time that is being wasted while waiting for something—to be transformed into "live time"—time that is being used to accomplish tasks.

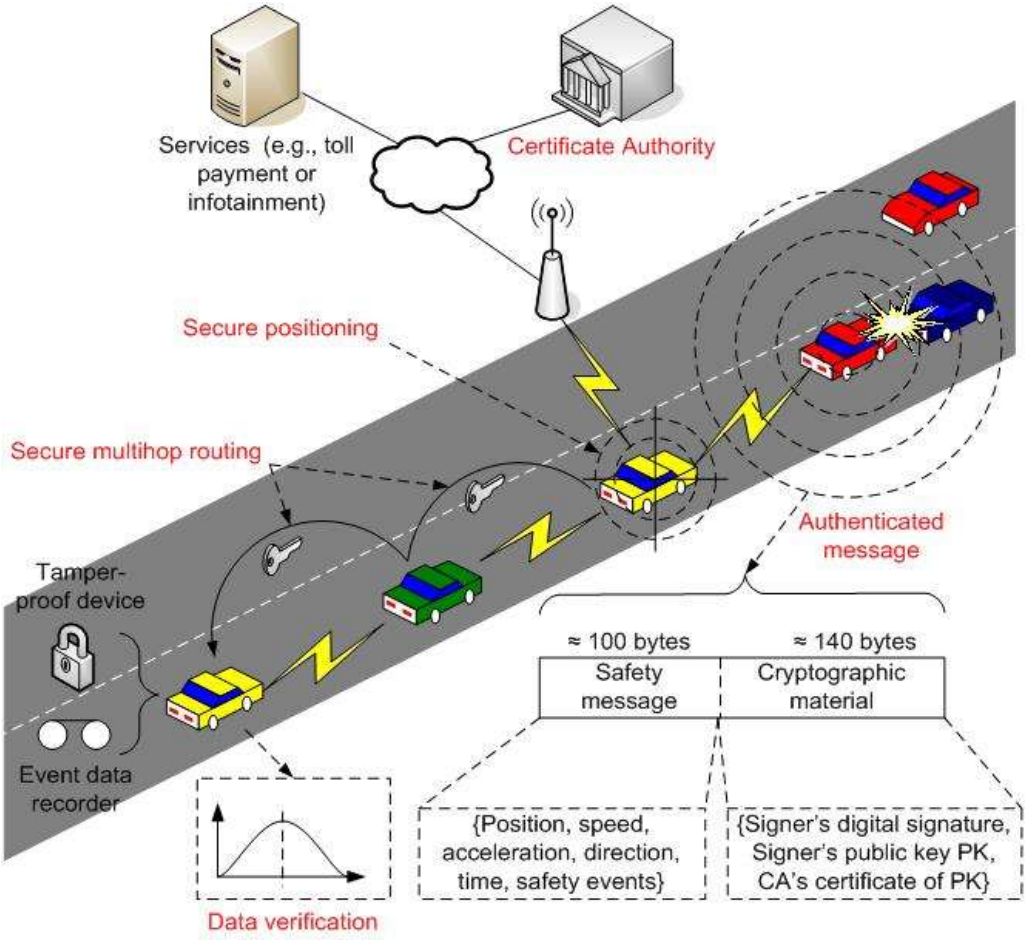


Figure 2.1: VANET SCENARIO & ARCHITECTURE

A commuter can turn a traffic jam into a productive work time by having his email downloaded and read to him by the on-board computer, or if traffic slows to a halt, read it himself. While waiting in the car to pick up a friend or relative, one can surf the Internet.

Even GPS systems can benefit, as they can be integrated with traffic reports to provide the fastest route to work. Lastly, it would allow for free, VoIP services such as Google Talk or Skype between employees, lowering telecommunications costs.

2.2.2 TECHNOLOGY:

IN VANET, or Intelligent Vehicular Ad-Hoc Networking, defines an intelligent way of using Vehicular Networking. INVANET integrates on multiple ad-hoc networking technologies such as Wi-Fi IEEE 802.11p, WAVE IEEE 1609, Wi-MAX IEEE 802.16, Bluetooth, IRA, and ZigBee for easy, accurate, effective and simple communication between vehicles on dynamic mobility. Effective measures such as media communication between vehicles can be enabled as well methods to track the automotive vehicles are also preferred.

- Global positioning system (GPS)
- WIFI (Communication medium)
- Processor with ability to store road map Information.

2.2.2.1 GLOBAL POSITIONING SYSTEM

The Global Positioning System (GPS) is a space-based global navigation satellite system (GNSS) that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver.

A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include

- the time the message was transmitted
- precise orbital information (the ephemeris)
- The general system health and rough orbits of all GPS satellites (the almanac).

The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite. These distances along with the satellites' locations are used with the possible aid of trilateration, depending on which algorithm is used, to compute the position of the receiver. This position is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS units show derived information such as direction and speed, calculated from position changes.

Three satellites might seem enough to solve for position since space has three dimensions and a position near the Earth's surface can be assumed. However, even a very small clock error multiplied by the very large speed of light — the speed at which satellite signals propagate — results in a large positional error. Therefore receivers use four or more satellites to solve for the receiver's location and time. The very accurately computed time is effectively hidden by most GPS applications, which use only the location.

A few specialized GPS applications do however use the time; these include time transfer, traffic signal timing, and synchronization of cell phone base stations. Although four satellites are required for normal operation, fewer apply in special cases. If one variable is already known, a receiver can determine its position using only three satellites. For example, a ship or aircraft may have known elevation. Some GPS receivers may use additional clues or assumptions (such as reusing the last known altitude, dead reckoning, inertial navigation, or including information from the vehicle computer) to give a less accurate (degraded) position when fewer than four satellites are visible.

2.2.3 COMMUNICATION MEDIUMS

2.2.3.1 Ad Hoc Wi-Fi Transmission

Wi-Fi is a trademark of the Wi-Fi Alliance. A Wi-Fi enabled device such as a personal computer, video game console, smartphone, or digital audio player can connect to the Internet when within range of a wireless network connected to the Internet. The coverage of one or more (interconnected) access points — called hotspots when offering public access — generally comprises an area the size of a few rooms but may be expanded to cover many square miles, depending on the number of access points with overlapping coverage.

Wi-Fi also allows communications directly from one computer to another without the involvement of an access point. This is called the ad hoc mode of Wi-Fi transmission. Wi-Fi networks have limited range. A typical wireless router using 802.11b or 802.11g with a stock antenna might have a range of 32 m (120 ft.) indoors and 95 m (300 ft.) outdoors. The IEEE 802.11n however, can exceed that range by more than two times. Range also varies with frequency band. Wi-Fi in the 2.4 GHz frequency block has slightly better range than Wi-Fi in the 5 GHz frequency block. Outdoor ranges - through use of directional antennas - can be improved with antennas located several kilometers or more from their base. In general, the maximum amount of power that a Wi-Fi device can transmit is limited by local regulations, such as FCC Part 15 in USA.

Due to reach requirements for wireless LAN applications, Wi-Fi has fairly high power consumption compared to some other standards. Technologies such as Bluetooth (designed to support wireless PAN applications) provide a much shorter propagation range of <10m and so in general have lower power consumption. Other low-power technologies such as ZigBee have fairly long range, but much lower data rate. The high power consumption of Wi-Fi makes battery life in mobile devices a concern. Due to unreliability of single mode of communication alternate modes are always under perusal.

2.2.3.2 Communication and connection

A master Bluetooth device can communicate with up to seven devices in a piconet. (An ad-hoc computer network using Bluetooth technology) The devices can switch roles, by agreement, and the slave can become the master at any time.

At any given time, data can be transferred between the master and one other device (except for the little-used broadcast mode). The master chooses which slave device to address; typically, it switches rapidly from one device to another in a round-robin fashion.

The Bluetooth Core Specification provides for the connection of two or more piconet's to form a scatter-net, in which certain devices serve as bridges, simultaneously playing the master role in one piconet and the slave role in another.

Many USB Bluetooth adapters or "dongles" are available, some of which also include an IrDA adapter. Older (pre-2003) Bluetooth dongles, however, have limited capabilities, offering only the Bluetooth Enumerator and a less-powerful Bluetooth Radio incarnation. Such devices can link computers with Bluetooth with a distance of 100 meters, but they do not offer as many services as modern adapters do.

2.3 BACKGROUND IDEAS & AGREEMENTS

2.3.1 Preliminaries

First of all systems with an authority responsible for assigning identities and credentials to all system entities has been denoted as nodes. All legitimate nodes are equipped with credentials (e.g., certified public keys) that the authority can revoke. Specific to the system and applications, Set is defined as

Set Ω = {α1, α2, α3 …αI} as mutually exclusive basic events.

Composite events are unions or intersections of multiple basic events. Examples of basic events are “block on the road” and “traffic jam”. If the ice on the road

causes a traffic jam, this becomes the composite event “block on the road and traffic jam ahead”. Each event A is a perceivable event generated by the environment, network, or an application running on vehicles.

There may be multiple applications, each having its own set of relevant events. These sets are overlapping, as their events belong to the pool of basic events. First consider V, the set of nodes vk, classified according to a system-specific set of node types, £ = µ1,µ2,… µn. A function V f (vk) returning the type of node vk. Reports are statements by nodes on events, including related time and geographic coordinates where applicable.

For simplicity, reports on basic events, as reports on composite events are straightforward. Usually its better not to dwell on the exact method for report generation, as this is specific to the application.

2.3.2 Default Trustworthiness

The default trustworthiness of a node vk of type µn as a real value that depends on the attributes related to the designated type of node vk. For all node types, there exists a trustworthiness ranking 0 < µ1 < µ2 < : < µN < 1. For example, some nodes are better protected from attacks, more closely monitored and frequently re-enforced, and, overall, more adequately equipped, with reliable components. As they are less likely to exhibit faulty behavior, they are considered more trustworthy.

There is stress here that the data-centric trust establishment framework does not aim to replace or amend source authentication, as in reputation systems, but uses it as an input to the data trust evaluation function. In fact, if a node reputation system was in place, its output scores could also be used as input to the data trust function. Hence, data trust builds on the information provided by source authentication and reputation systems without trying to supplant them. The choice of the entity trust establishment system is orthogonal to the scope of this project and has been prolifically addressed throughout the foundation discussions.

2.3.3 Event- or Task-Specific Trustworthiness

Nodes in general perform multiple tasks that are system-, node- and protocol-specific actions. Let A be the set of all relevant system tasks. Then for some nodes v1 and v2 with types f (v1) = μ1 and f (v2) = μ2 and default trustworthiness rankings tμ1 < tμ2 , it is possible that v1 is more trustworthy than v2 with respect to a task α1. Reporting data on events is clearly one of the node tasks. For the sake of simplicity, it's usual to talk here about event-specific trustworthiness implying that it is actually task-specific trustworthiness.

Nevertheless, the two can be easily distinguished, when necessary, when tasks include any other protocol specific action such as communication. With the above considerations in mind, the event specific trustworthiness function f : Θ * Λ -> [0, 1]. f has two arguments: the type f(vk) of the reporting node vk and the task ¸j . f does differentiate among any two or more nodes of the same type, and if ¸j = ; (no specific event or task), f is the default trustworthiness f = tf(vk).

2.3.4 Dynamic Trustworthiness Factors

The ability to dynamically update trustworthiness can be valuable, especially for capturing the intricacies of a mobile ad hoc networking environment. For example, nodes can become faulty or compromised by attackers and hence need to be revoked. In addition, the location and time of report generation change fast and are important in assigning trustworthiness values to events.

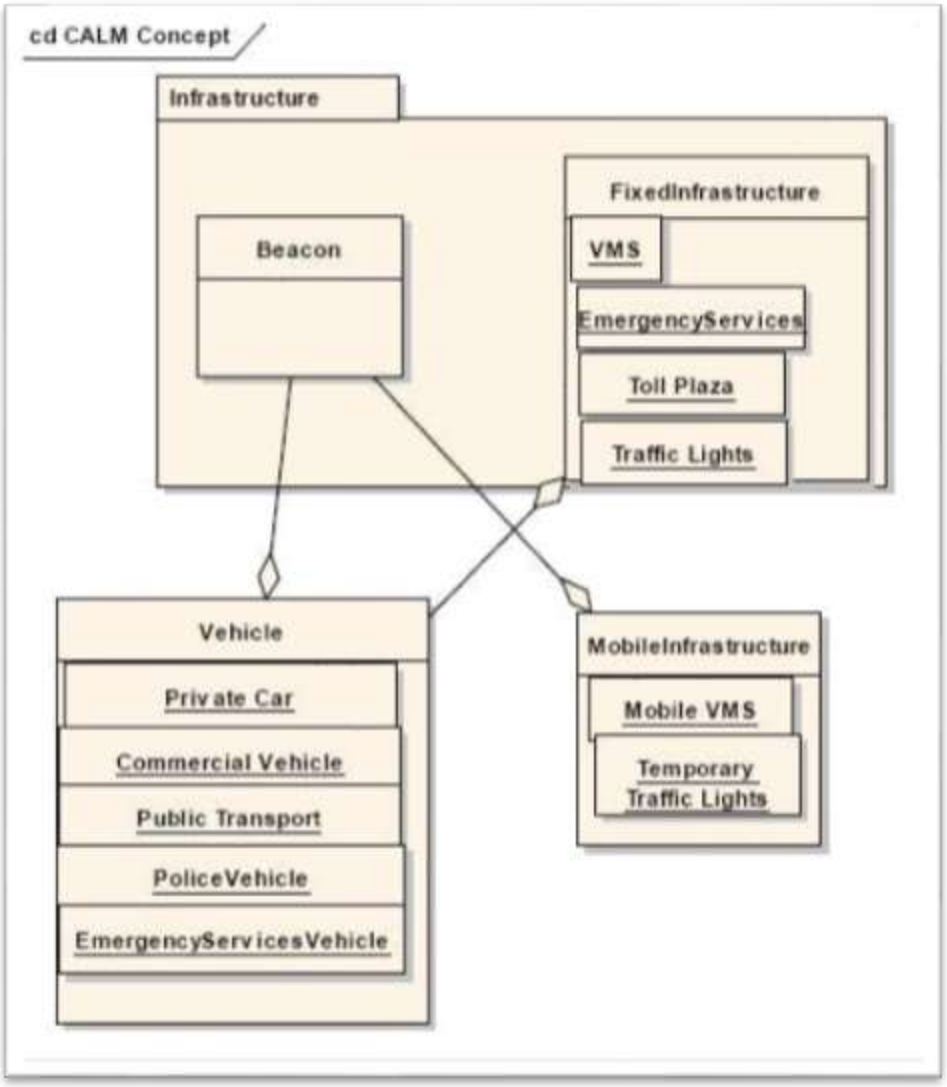


Figure 2.2: Dynamic trust relationship diagram

To capture this, a security status function s: V f(vk) [0; 1]. s (vk) = 0 implies node vk is revoked, and s(vk) = 1 implies that the node is legitimate. Intermediate values can be used by the system designer to denote different trustworthiness levels, if applicable.

Second, a set of dynamic trust metric functions has been evaluated M = f(vk) V £ -> [0; 1] g indexed by a selector l indicating different node attributes (e.g., location) that dynamically change. That is, for each attribute, a different metric ¹l is defined. ¹l takes node vk 2 V and task ¸j 2 ¤ as inputs and returns a real value in [0; 1].

2.3.5 Location and Time

Among the possible values of l for metric ¹l, proximity either in time or geographic location is an attribute of particular importance. Proximity can increase the trustworthiness of a report: The closer the reporter is to the location of an event, the more likely it is to have accurate information on the event. Similarly, the more recent and the closer to the event occurrence time a report is generated, the more likely it is to reflect the system state.

Cryptographic primitives, such as digital signatures, can ensure that location and time information cannot be modified if included in a report. However, the accuracy of such information can vary, due to nodes' differing capabilities or (malicious or benign) faults. This is especially true for reports that depend on fine-grained time and location data. Hence, different types of nodes are more or less trustworthy when reporting such data. In some cases, time- or geo-stamping a report can be a distinct task.

2.3.6 Scheme Overview

The trustworthiness of a report f(vk), generated by node vk and providing supporting evidence for event j , by using both

(i) Static or slow-evolving information on trustworthiness, captured by the default values and the event specific trust f, and

(ii) Dynamically changing information captured by security status s and more so by metric ¹l.

These arguments are combined into a function

F (ejk) = G(s(vk); f(¿ (vk); ¸j) (vk; ¸j)) that returns values in the [0, 1] interval. If vk reports no evidence for j , F(ejk) = 0. These values are calculated locally for each report received from another node and are called the weights (or trust levels) of the report.

Nonetheless, such a message assessment may often be insufficient. It can be hard to decide whether the reported event took place based on a single message, and it is vulnerable to faults (e.g., equipment failures or compromised nodes). Instead, the collection of multiple reports related to the same event and

of their weights, i.e., the accompanying F values, and their combination into a robust decision scheme. Thus, the reports along with their weights are passed to a Decision Logic module that outputs an assessment on the event in question. The way to use such decisions and inferences is specific to particular systems. The above process is tightly related to the multi sensor data fusion techniques. In fact, F can be computed using rule based expert systems; the output of the Decision Logic module can be used by another expert system that makes decisions based on reported events. There are several algorithms to implement the Decision Logic module, in which it compares next a selected subset of these algorithms in the context of data centric trust establishment. It should be stressed here that data-centric trust establishment in wireless networks is a new application of data fusion techniques, to the best of our knowledge.

2.4 EVIDENCE EVALUATION

The literature on trust in ad hoc networks proposes several approaches for trust establishment. A new technique has been established and compared it to four other existing techniques. These techniques are described below.

To mathematically model our approach, assume a node A has to decide among several basic events vk …, based on K pieces of evidence ejk (reports from K distinct nodes).

Let di(vk) denote the combined trust level computed by evaluating evidence corresponding to event i. The Decision Logic module outputs the event that has the highest combined trust level, i.e., maximum (di).

2.4.1. Basic techniques

The following two techniques are used for reference and serve as a basis of comparison for the remaining three techniques.

i)Majority Voting

ii)Most Trusted Report

2.4.1.1 Majority Voting

In this technique, the majority wins. The combined trust level corresponding to event ®i is defined by: di =1K

XKk=1F(eik) (1) where F(eik) = 1 if vk reports ®i and it is 0 otherwise.

2.4.1.2 Most Trusted Report

The Most Trusted Report (MTR) decision logic outputs a trust level equal to the maximum value of trust levels assigned to reports about the event; the point of using MTR is to show the effect of isolated high trust values (in data or entities) on the system. The combined trust level corresponding to event ®i is defined by: di = maxk(F(eik))

2.4.2. Weighted Voting

As its names implies, Weighted Voting (WV) sums up all the votes supporting an event with each vote weighted by the corresponding trust level to output the combined trust level:

di =1 K XK k=1 F(eik) (3)

It should be noted here that decisions on composite events are harder to do using the above three techniques since they do not provide formalisms for handling unions and intersections of events. In contrast, the next two techniques provide such formalisms.

2.4.3. Bayesian Inference

Among the data fusion techniques, Bayesian Inference (BI) [20] is the one most frequently used for trust establishment.

In BI, the combined trust level corresponding to ®i is the posterior probability of ®i given new evidence e = fej 1; ej it is expressed in terms of the prior probability P[®i] using the Bayes' theorem.

Where it is assumed that reports are independent for the sake of mathematical tractability (the receiver cannot sort out the dependencies among

reports from distinct vehicles since such information is not provided in the reports).

The computation of posterior probabilities for composite events ° (recall that they are unions or intersections of basic events) follow the rules of probability theory P[eikj Θ i] is the probability that report k confirms event Θi, given that Θi happened. Using trust levels as weights of reports, this probability is equal to the trust level: P[eikj Θ i] = F(eik).

For j 6= i, P[ejkj Θ i] is the probability that report k does not confirm ®i (hence, it confirms ¹ ®i, the complement of ®i in ,given that ®i happened. This is equivalent to a malfunctioning or cheating node (ideally, a node would report a real event).

2.4.4 Dempster-Shafer Theory

In Dempster-Shafer Theory (DST), evidence evaluation is inspired by human reasoning. More specifically, the lack of knowledge about an event is not necessarily refuted of the event. In addition, if there are two conflicting events, uncertainty about one of them can be considered as supporting evidence for the other. The major difference between BI and DST is that the latter is more suitable for cases with uncertain or no information. More precisely, in DST a node can be uncertain about an event, unlike in BI where a node either confirms or refutes the event. For example, if a node A confirms the presence of an event with probability p, in BI it refutes the existence of the event with probability 1 ¡ p.

In DST, probability is replaced by an uncertainty interval bounded by belief and plausibility. Belief is the lower bound of this interval and represents supporting evidence. Plausibility is the upper bound of the interval and represents non-refuting evidence. Hence, in this example, node A has p degree of belief in the event and 0 degree of belief in its absence.

In DST, the frame of discernment contains all mutually exclusive possibilities related to an observation. Hence, in our context, it is the set - defined previously. The belief value corresponding to an event provided by report k is computed.

Which means it is the sum of all basic belief assignments mk(®q), ®q being all basic events that compose the event.

The plausibility value corresponding to event ®i represents the sum of all evidence that does not refute ®i and is computed.

Belief and plausibility are related by pls(®i) = 1¡bel(¹ ®i). The combined trust level corresponding to event ®i is the belief corresponding to the event.

As before, using trust levels as weights of reports, the basic belief assignment that confirms ®i is equal to the trust level:

For composite events, belief can be computed similarly using the above equations. To illustrate the application and utility of the data trust framework, the following a case study of a real ephemeral ad hoc network instantiation has been presented namely vehicular networks. First the system and adversary models, then explain through examples how the different components of data trust can be practically derived.

2.5 Secure Vehicular Communications System

Vehicular Ad hoc Networks (VANET) and Vehicular Communication (VC) systems [2] are being developed to enhance the safety and efficiency of transportation systems, providing, for example, warnings on environmental hazards (e.g., block on the pavement) and traffic and road conditions (e.g., emergency braking, congestion, or construction sites). From a networking point of view, the nodes are vehicles and road-side infrastructure units (RSUs), all equipped with on-board processing and wireless modules, thus enabling multi-hop communication in general.

Authorities are public agencies or corporations with administrative powers, e.g., city or state transportation authorities entrusted with the management of node identities and credentials.

A subset of the infrastructure nodes serves as a gateway to and from the authorities. Its assumed that each node vk is equipped with a pair of private/public cryptographic keys Prk/Puk, and a certificate issued by an authority X as CertXfPukg. Nodes are equipped with a clock and a positioning system

(such as GPS or Galileo). This allows them to include their time and location information in any outgoing reports. Source authentication, required to prevent Sybil attacks, is achieved by digital signatures according to both industrial and academic proposals [2], [1]. In this example, source authentication identifies the type of the report sender and enables the assignment of default trustworthiness, as explained above.

Unicast and multicast communication is possible; however, local broadcast (single hop) and geocast (flooding to a given geographic area) are predominantly used. Vehicle-specific information (e.g., velocity, coordinates) is transmitted frequently and periodically in the form of safety messages.1 Reports on in-vehicle or network events are included in these messages.

Safety and other messages, generated by vehicles and RSUs, can result in an abundant influx of information about events. It is important to note here that our approach, based exclusively on local processing, does not add any communication overhead and very little computation overhead to a secure VC system where the actual overhead is due to frequent broadcasting and asymmetric cryptography and is inherent in VANETs.

2.6 Adversary Model

Nodes either comply with the implemented protocols (i.e., they are correct) or they deviate from the protocol definition intentionally (attackers) or unintentionally (faulty nodes). Both attackers and faulty nodes can cause damage to the network and hence considered them both as adversaries. The attacks that can be mounted by either internal (equipped with credentials and cryptographic keys) or external adversaries vary greatly. In brief, adversaries can replay any message, jam communications, and modify (yet in a detectable manner due to the digital signatures) messages. More importantly, they can inject faulty data and reports, or control the inputs to otherwise benign nodes and induce them to generate faulty reports.

It’s assumed that at most a small fraction of the nodes are adversaries, and consequently the fraction of the network area affected by them is bounded. This bound on the presence of adversaries could be further refined by distinct

values for different node types. But this assumption does not preclude that a few adversarial nodes surround a correct node at some point in time.

2.7 Framework Instantiation

The focus is on the use of our scheme on board a vehicle. Clearly, it could be run on RSUs; nonetheless, the challenge is to design a scheme practical for nodes that are not part of the system infrastructure.

The forms of the f (event-specific trust), s (security status), ¹l (dynamic trust metric), and G (trust level) functions are determined by the secure VC system: They are either preloaded at the time the node is bootstrapped, or updated after the node joined the system. Their values are either provided by the authorities or distributed by the infrastructure.

To illustrate our instantiation, it is considered as an example scenario: a highway accident in which vehicle B is involved.

Now, let us consider a vehicle A, several communications hops away from the accident location. A receives safety messages indicating that there is an accident on its route and has to decide whether to trust this information. In this case, we assume the event ©1: “There is an accident at location LB”. The granularity of the event location should be properly defined to avoid having reports on several different events while, actually, all these reports refer to the same event but with slightly different locations. Now assume that one or more attackers generate safety messages supporting the null event. There is no accident at location LB”. If there are several events (e.g., several distinct locations, given the defined granularity), the data trust is computed for each of them. The resulting values can be used by the application to decide the consequent action.

Typically, for a highway, every 300ms over a nominal range of 300m.

Two important system parameters are the set of reports and the time needed to make a correct judgment. The reports considered valid for making decisions should be sent by vehicles that are on the communication path between the accident location and A. The time to correct judgment should be

equal to the time needed by the Decision Logic module to converge to a stable output value; this time depends on the frequency of message reception. It is also constrained by the tolerable decision delay (e.g., in critical situations, decisions should be made very fast, given the available data) that depends on the event in question.

CHAPTER III

IMPLEMENTATION

3.1 ARCHITECTURE DEPLOYMENT

A method is designed to allow cars to autonomously collect and share traffic information (i.e., area passage time) using only inter-vehicle communication. In this method, a target geographical region is divided into fixed sub-regions called areas. Each car measures time to pass an area with respect to each pair of roads entering or exiting from the area, and generates traffic information statistics based on the information received from cars which passed the same pair of roads. Cars exchange and aggregate traffic information of each area using inter-vehicle communications, studied under the assumption that a density of cars is large enough. Therefore traffic information statistics tends to be lost in areas with low car density.

3.1.1. Message Ferrying

The message ferrying technique aims to achieve efficient data propagation in disconnected ad hoc networks [2]. In ad hoc networks, communications are often unstable due to the limitation of wireless range and node mobility. The ferry relays messages between nodes which cannot communicate directly. In this technique, all nodes are classified into regular nodes and message ferries. Here, regular nodes move freely, but ferries move regularly along the predetermined routes.

Regular nodes send messages to ferries or receive messages from ferries. Ferries collect messages from regular nodes, move to other disconnected portion of ad hoc networks, and send the collected messages there.

3.1.2 Method for Sharing Traffic Information using Inter-Vehicle Communication among only Cars

The aim is to gather information using short range wireless communication, GPS and small computer on each car, without using fixed

infrastructure on the ground. Similarly, its assumed that each car has an onboard computer with the following functionalities.

• IEEE 802.11 compliant wireless LAN device

• GPS receiver

• Hard disk drive to store traffic information

• Road map data (on HDD)

• Computer with sufficient power for instantly processing received information

It's also assumed that a given road map can be treated as a graph where each node and each link correspond to an intersection and a road between intersections, respectively.

3.1.3 Measuring Area Passage Time

In the proposed method, the target geographical region is divided into square areas with sides of several hundred meters length as shown in figure 3.1. This is to avoid significantly increasing the amount of traffic information and to manage it efficiently. Each area is assigned a unique ID number and that onboard computer on each car has information regarding to locations of areas and their IDs. The size of areas can be changed according to a road density and road shapes.

When each car leaves an area, the car measures elapsed time since it entered the area. The elapsed time is recorded as area passage time. The links along which a car enters and exits an area incoming link and outgoing link, respectively.

A pair of incoming and outgoing links is called link pair. In order to consider waiting time at each intersection caused by traffic lights and queue of cars turning left/right, collect area passage time at every link pair.

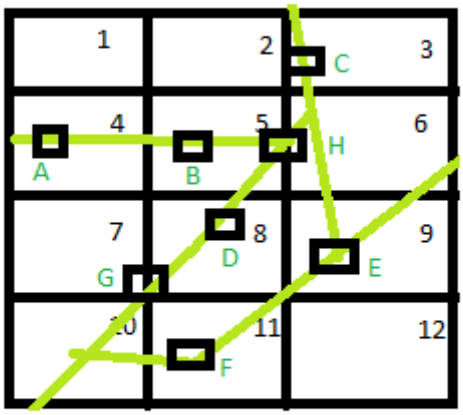


FIGURE 3.1: GRID – ARCHITECTURE FOR VANET

In Figure 3.1, in case a car passed through Area A9 along Links B-H-K-O, the incoming link is B-H and the outgoing link is K-O. Then the link pair is (B-H, K-O). The area passage time of each car in an area passage record has been recorded which consists of the following information.

(AreaID, InlinkID,OutlinkID,APT,MakeTime, hasha)

Here, AreaID denotes the ID of the area, InlinkID and OutlinkID denote the link IDs of incoming and outgoing links to/from the area, respectively. APT denotes the time to pass through the area. MakeTime denotes the time when the car crossed the edge of the area and measured APT. It is used when old records are discarded. hasha denotes a hash value calculated from the car ID and MakeTime.

Each car periodically broadcasts area passage records toward neighboring cars. When a car receives area passage records, the car compares their hash values with its stored records. If the hash values are equal, the received record is discarded to avoid collecting same record redundantly.

3.1.4 Generating and Disseminating Area Passage Statistics

As the number of area passage records retained in a car grows, the amount of data which the car broadcasts also grows. Due to limitation of available bandwidth in wireless network, the amount of data has been reduced. In the proposed method, when the number of area passage records for a link pair retained in a car reaches a predetermined threshold C (which is 3 to 5, typically), the values of these records are averaged by creating a statistics data called area passage statistics, and the original area passage records are removed.

The area passage statistics include the following information (it can contain multiple records for different linkpairs)

(AreaID, inlinkID, outlinkID,AAPT,MakeTime, hashs)

Here, AAPT denotes the average area passage time of C cars.

hashs is calculated from hash values hasha of the original area passage records, and used to avoid redundantly counting the same statistics.

Each car periodically broadcasts area passage records and statistics data which it retains. When a car receives area passage statistics data from another car, it compares the hash value of each statistics data which it retains, and stores the received data if it is not already retained.

Area passage statistics data have expiration time, and they are discarded after the time elapses from their generation.

3.1.5 Data Maintenance and Discard in Inter-Vehicle Communication

It is not reasonable for each car to retain information of all areas for the following reason: Wireless bandwidth of the inter-vehicle communication is limited; storage size of computer equipped at each car is also limited; and information of the areas which are far from current area may not be needed because it is usually old when it is received in current area.

In order to address this problem, each car retains information of only responsible areas (current area and its neighboring areas). However, in this method, traffic information statistics may be propagated to a few cars or may be

lost when a density of cars is low. After a car passes through area A and generates an area passage record, if there are no cars in wireless communication range of the car while it goes through the responsible areas of A, that record is discarded without being received by other cars. Moreover, a car cannot get information of distant areas since each car retains information of only the responsible areas of the current area.

3.1.6 DIFFERENCE BETWEEN REGULAR CARS AND BUSES

Regular Cars Buses Equipment's

- IEEE802.11g compliant wireless LAN
- GPS receiver
- HDD includes Road Map data
- Computer with sufficient power for instantly processing received data

(HDD capacity) about 30GB larger than car

Behavioral depend on travel along predetermined pattern drivers routes periodically.

3.2 Simulation of Inter-Vehicle Communication

The implemented modules and MOBISIM nodes can communicate with each other every second in simulation time scale, and those modules can get position of each car.

3.2.1 Configuration of Inter-Vehicle Communication

The direct communication range of each vehicle (car or bus) is 100 meters. When many cars in their communication ranges each other send data packets simultaneously, packet collisions may occur to lead to congestion of communication. In order to simulate this phenomenon, one second has been divided into 100 time slots with 10 milliseconds lengths and assumes that each packet occupies one time slot when it is broadcasted.

If two and more packets try to be broadcasted simultaneously, they are collided with one another and they cannot be received. This simulation is executed by time slot.

Total simulation time is 60 minutes (i.e., 360,000 time slots). The size of each packet is assumed to be 1500 bytes. It is known that radio field intensity is inversely proportional to the square of distance from a transmitter station.

However, a probability of successful packet reception is not simply proportional to radio field intensity. All cars and all buses are equipped with a system which carries out generation and broadcast of area passage records as well as reception, update and re-broadcast of area passage statistics data for each link pair.

3.2.2 Configuration of Road Map

A general road map consisting of two main roads crossing at the center and several byroads has been used. The map has 1.2km sides, and 29 nodes (intersections) and 39 bidirectional links (roads).

Main roads are represented by thick lines, and they have two lanes each way. The other roads are byroads which have one lane each way. Intersections with a signal are represented by circles of thick line. In an intersection of two main roads, the green signal for each road takes 60 seconds. In the intersection of a main road and a by-road, the green signal for the main road takes 60 seconds and the green signal for the byroad takes 30 seconds.

MOBISIM can simulate realistic traffic flow, including a car recurrence interval, based on a configuration like the maximum car density and the number of lanes of roads and the number of cars on the map. Thus the density of each road is not fixed value but it is changed over time. The maximum speed of each car and bus is 60km per hour. In this experiment, we configured parameters of MOBISIM to make the following two situations and evaluated our proposed method. Those parameter values are obtained by an exploratory experiment.

• Low car density: minimum car density where each car can communicate by using inter-vehicle communication (a car per 94m road-length).

• Very low car density: car density where each car may not communicate by using inter-vehicle communication

In some cases (a car per 313m road-length). Two bus routes a and b, Buses travel along the route a every five minutes and the other buses travel along the route b every seven minutes. This map is a suburban area where several bus routes are provided and about four buses travel along them every hour.

MOBISIM used in this paper can simulate the behavior of a bus by specifying the track of the bus. A timetable to MOBISIM in order to consider bus stops has been given. It indicates when a bus goes through each intersection. Its assumed that when the bus arrives at an intersection regarded as a bus stop ahead of schedule, the bus has to stop at the intersection until scheduled time. On the other hand, when the bus arrives at the intersection behind schedule, the bus has to stop at the intersection for 15 seconds.

3.2.3 Configuration of Area Passage Statistics

As threshold C for generating area passage statistics data, we used C = 5. We assume that a car discards area passage statistics data after a lapse of 10 minutes since the data generated.

3.2.4 Configuration of Control Packet and Transmission Interval

A car broadcasts area passage records five times at random timings for five seconds after it generated them. A car also broadcasts some retained area passage records and area passage statistics data at a random timing every five seconds. A bus broadcasts a bus (control) packet at a random timing every two seconds. A car which received a bus packet has to reply within one second, and it sends a car packet to require specific information to the bus at the same time.

Similarly, a bus which received a car packet has to reply within one second. AreaID of a car packet is one of its responsible areas, and Priority is assigned a random value for simplicity.

3.3 SIMULATION ENVIRONMENT CONSTRUCTION

The Vehicular Ad Hoc Networks Mobility Simulator (VanetMobiSim) is a set of extensions to CanuMobiSim, a framework for user mobility modeling used by the CANU (Communication in Ad Hoc Networks for Ubiquitous Computing) Research Group [1], University of Stuttgart. The framework includes a number of mobility models, as well as parsers for geographic data sources in various formats, and a visualization module. The framework is easily extensible. It is based on the concept of pluggable modules.

The set of extensions provided by VanetMobiSim consists mainly on a vehicular spatial model using GDF-compliant data structures, and a set of vehicular-oriented mobility models. The vehicular spatial model is composed of spatial elements, their attributes and the relationships linking these spatial elements in order to describe vehicular areas. The spatial model is created from topological data obtained in four different ways:

• User-defined – The user defines a set of vertices and edges composing the backbone of the vehicular spatial model.

• Random – The backbone is randomly generated using the Voronoi tessellations.

• Geographic Data Files (GDF) – Backbone data is obtained from GDF files.

• TIGER/line Files – Similar to the previous one, but based on the TIGER/line files from the US Census Bureau.

Any one of those methods MUST be loaded AFTER the Spatial Model, as it controls all data describing the topology. Then, it adds vehicular specific spatial elements such as multi-lane and multi-flow roads, stop signs and traffic lights.

The main component of the vehicular oriented model is the support of a microscopic level mobility model named “Intelligent Driving Model with Intersection Management (IDM_IM)” describing perfectly car-to-car and

intersection managements. In the Intelligent Driving Model with Lane Changing (IDM_LC), an overtaking model (MOBIL) is included, which interacts with IDM_IM to manage lane changes and vehicle accelerations and decelerations. A Spatial Environment MUST is loaded for user-oriented and vehicular oriented mobility models to work. A spatial Environment MAY be loaded for all other mobility models specified in CanuMobiSim.

3.3.1 Format of Simulation Scenario

The simulation scenario for VanetMobiSim is similar to CanuMobiSim's. It is defined in XML format. In the notation below, tags or attributes appearing in square brackets (e.g., [<sample>] are optional.

Specifying a Simulation Area

A simulation area is specified using the <universe> tag.

<universe>

[<dimx>dimension</dimx>]

[<dimy>dimension</dimy>]

[<step>step</step>]

[<seed>seed</seed>]

[<extension>extension_parameters</extension>]

[<node>node_parameter</node>]

[<nodegroup>nodegroup_parameters</nodegroup>]

</universe>

• dimx – specifies the x-dimension of the simulation are (in meters). Only used in the scenarios with rectangular-bounded simulation areas.

• dimy – specifies the y-dimension of the simulation are (in meters). Only used in the scenarios with rectangular-bounded simulation areas.

• step – specifies the duration of single simulation time-step (in s). If omitted, the value of 1 ms is used.

• seed – specifies the seed of the random number generation used by VanetMobiSim.

• extension – adds an instance of a global extension to the simulation.

• node – adds a node to the simulation.

• nodegroup – adds a group of nodes to the simulation.

3.3.2 Adding a Global Extension to Simulation

An instance of global extension is added using the <extension> tag.

<extension class=”class_name” [name=”instance_name”]>

[extension_parameters]

</extension>

• class – specifies the name of class to be instantiated. The class must be derived from de.uni_stuttgart.informatik.canu.mobisim.core.ExtensionModule and be accessible by JVM.

• name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. Most of extensions have their default name predefined.

3.3.3 Adding a Node to Simulation

A node is added to simulation using the <node> tag.

<node [class=”class_name”] id=”node_id”>

[<position>position_parameters</position>]

[<type>type_of_node</position>]

[<extension>extension_parameters</extension>]

</node>

• class – specifies the node's class name. The class must be derived from de.uni_stuttgart.informatik.canu.mobisim.core.Node and be accessible by the JVM. If omitted, de.uni_stuttgart.informatik.canu.mobisim.core.Node is used\

• id – specifies the node's id. Used to uniquely identify and reference the node in simulation

• position – specifies the node's initial position

• type – specifies the node's type. The user can choose among four different types of nodes: ped-car-truck-bus. If omitted, the value "any" is taken by default.

• extension – adds an extension to the node (e.g., instance of mobility model).

3.3.4 Specifying the Node's Initial Position

The node's initial position is specified using the <position> tag.

<position [graph="graph_instance_name"] [random="is_random"]>

[<x>x_value</x>]

[<y>y_value</y>]

</position>

• graph – if the position belongs to the graph, specifies the name of graph instance (class de.uni_stuttgart.informatik.canu.mobisim.extensions.Graph). Used by the graph-based mobility model.

• random – specifies that the position must be chosen randomly. The value is of Boo-lean type. If the position belongs to a graph, it will be chosen randomly from the graph vertices.

• x – specifies the position's x-coordinate (in m).

• y – specifies the position's y-coordinate (in m).

3.3.5 Adding a Group of Nodes to Simulation

Multiple nodes (a group of nodes) are added to the simulation using the <nodegroup> tag.

<nodegroup n=”number_of_nodes” [class=”class_name”] id=”group_id”>

[<position>position_parameters</position>]

[<type>type_of_nodes</position>]

[<extension>extension_parameters</extension>]

</node>

• n – specifies the number of nodes in the group.

• class – specifies the node’s class name. The class must be derived from de.uni_stuttgart.informatik.canu.mobisim.core.Node and be accessible by the JVM. If omitted, de.uni_stuttgart.informatik.canu.mobisim.core.Node is used\

• id – specifies the group id. Used for choosing nodes’ identifiers by concatenating the group id with the node’s sequence number.

• position – specifies the initial position for all nodes in the group.

• type – specifies the node’s type assigned to all member of this group. The user can choose among four different types of nodes: ped-car-truck-bus. If omitted, the value “any” is taken by default.

• extension – adds instances of extension to each node.

3.3.6 Globally Specified Extensions

The extensions can be added globally to a simulation. All of them need de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel.

3.3.6.1 Spatial Environment:

A spatial environment is added with an instance of de.uni_stuttgart.informatik.canu.spatialmodel .core.SpatialModel extension. For a

correct behavior of VanetMobiSim, this extension shall not be omitted. As the spatial environment extension controls all topological and mobility extensions, it MUST be declared before them. It also adds support for multilane or multi-flow roads and traffic lights at intersections. It can be initialized from three different ways. First, it can be initialized from an appropriate geographic data source (GDF or TIGER). Or, it can also be initialized from an appropriate eurecom.usergraph.UserGraph extension. Finally, it can also be initialized from an appropriate eurecom.spacegraph.SpaceGraph extension.

<extensionname="instance_name"
class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel"

[traffic_light=" trafficlight_instance_name"] min_x="value" min_y="value" max_x="value" max_y="value">

[<max_traffic_lights>traffic_lights</max_traffic_lights>]

[<number_lane full="value" max="value" dir="value">lanes_number</number_lane>]

[<reflect_directions>value</reflect_directions>]

</extension>

• name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. The default name is "SpatialModel". Changing the default name is not recommended.

•traffic_light–specifies the name of the eurecom.spatialmodel.extensions.TrafficLight extension if different from the default value.

• min_x – specifies the leftmost x-coordinate of "clipping window" (in m). The coordinate is relative to the source's minimal x-coordinate. Used to process a part of geographic area.

• min_y – specifies the lowermost y-coordinate of "clipping window" (in m). The coordinate is relative to the source's minimal y-coordinate. Used to process a part of geographic area.

• max_x – specifies the rightmost x-coordinate of “clipping window” (in m). The coordinate is relative to the source’s maximal x-coordinate. Used to process a part of geographic area.

• max_y – specifies the uppermost y-coordinate of “clipping window” (in m).

3.3.6.2 Specifying Multi-lane Roads

The characteristics of multi-lane roads in the Spatial model are specified using the <number_lane> tag. When specifying a multi-lane road, the spatial model intends to model highways and therefore will generate a multi-lane highway starting from one border and ending on a different border using the Dijkstra shortest path algorithm.

<number_lane,full=”value”,max=”value”
dir=”value”>lanes_number</number_lane>

• full – specifies whether all roads have multiple lanes or not.

• max – if the <full> attribute is false, specifies the maximum number of roads with multi-lane capability in the graph, i.e. the size of the subset of roads modeled as highways. If omitted, the default value is 4.

• dir – specifies if the spatial model physically differentiates the two traffic flows. If the <full> attribute is true, <dir> and <reflect_directions> are equivalent. If not, <dir> allows the user to differentiate the directional flows of multi-lane roads only. If omitted, the default value is false.

• lanes_number – specifies the number of lanes in multi-lane roads. If omitted, the default value is 1.

3.4 PRACTICAL PDA HELPER TOOL IMPLEMENTATION

3.4.1. J2ME

Java Platform, Micro Edition or Java ME is a specification of a subset of the Java platform aimed at providing a certified collection of Java APIs [11] for the development of software for tiny, small and resource-constrained devices. Target

devices range from industrial control and automotive devices to cell phones and set-top boxes.

Java ME devices implement a profile. The most common of these are the Mobile Information Device Profile aimed at mobile devices, such as cell phones, and the Personal Profile aimed at consumer products and embedded devices like Set-top boxes and PDAs.

Profiles are subsets of configurations, of which there are currently two [10]:

- Connected Limited Device Configuration.
- Connected Device Configuration.

3.4.1.1. Connected Limited Device Configuration

The Connected Limited Device Configuration (CLDC) contains a strict subset of the Java-class libraries, and is the minimum amount needed for a Java virtual machine to operate. CLDC is basically used to classify myriad devices into a fixed configuration.

A configuration provides the most basic set of libraries and virtual-machine features that must be present in each implementation of a J2ME environment. When coupled with one or more profiles, the Connected Limited Device Configuration gives developers a solid Java platform for creating applications for consumer and embedded devices.

3.4.1.2. Connected Device Configuration

The Connected Device Configuration is a subset of Java SE, containing almost all the libraries that are not GUI related. It is richer than CLDC.

3.5. PROGRAM CONCEPTS

3.5.1. Thread

3.5.1.1. The java thread model

The Java run-time system depends on threads for many things, and all the class libraries are designed with multithreading in mind. In fact, Java uses threads to enable the entire environment to be asynchronous. This helps reduce inefficiency by preventing the waste of CPU cycles.

3.5.1.2. The main thread

When a Java program starts up, one thread begins running immediately. This is usually called the main thread the program, because it is the one that is executed when the program begins. The main thread is important for two reasons:

- It is the thread from which other "child" threads will be spawned.
- It must be the last thread to finish execution. When the main thread stops, the program terminates.

The sleep() method causes the thread from which it is called to suspend execution for the specified period of milliseconds.

3.5.1.3. Creating a thread

- Implementing Runnable.
- The easiest way to create a thread is to create a class that implements the Runnable interface.
- After the new thread is created, it will not start running until its start() method is called, which is declared within Thread. In essence, start() executes a call to run().

3.5.1.4. Extending thread:

The second way to create a thread is to create a new class that extends Thread, and then to create an instance of that class.

The extending class must override the run() method, which is the entry point for the new thread. It must also call start() to begin execution of the new thread.

3.5.1.5. Interthread communication:

- wait() tells the calling thread to give up the monitor and go to sleep until some other thread enters the same monitor and calls notify().
- notify() wakes up the first thread that called wait() on the same object.
- notifyAll() wakes up all the threads that called wait() on the same object. The highest priority thread will run first.

3.5.2. MIDlet

J2ME applications referred to as a MIDlet can run on practically any mobile communications device that implements a JVM and MIDP. A MIDlet is not invoked the same way as a J2SE application is invoked because many small computing devices don't have a command prompt. MIDlets are controlled by application management software (AMS). The manufacturer of a small computing device provides AMS, although third-party vendors might also create AMS. AMS interacts with native operations of a small computing device and controls the life cycle of a MIDlet. The life cycle consists of installation and upgrades as well as version management and uninstalling the application. Likewise, AMS is responsible for starting, managing execution, and stopping the MIDlet.

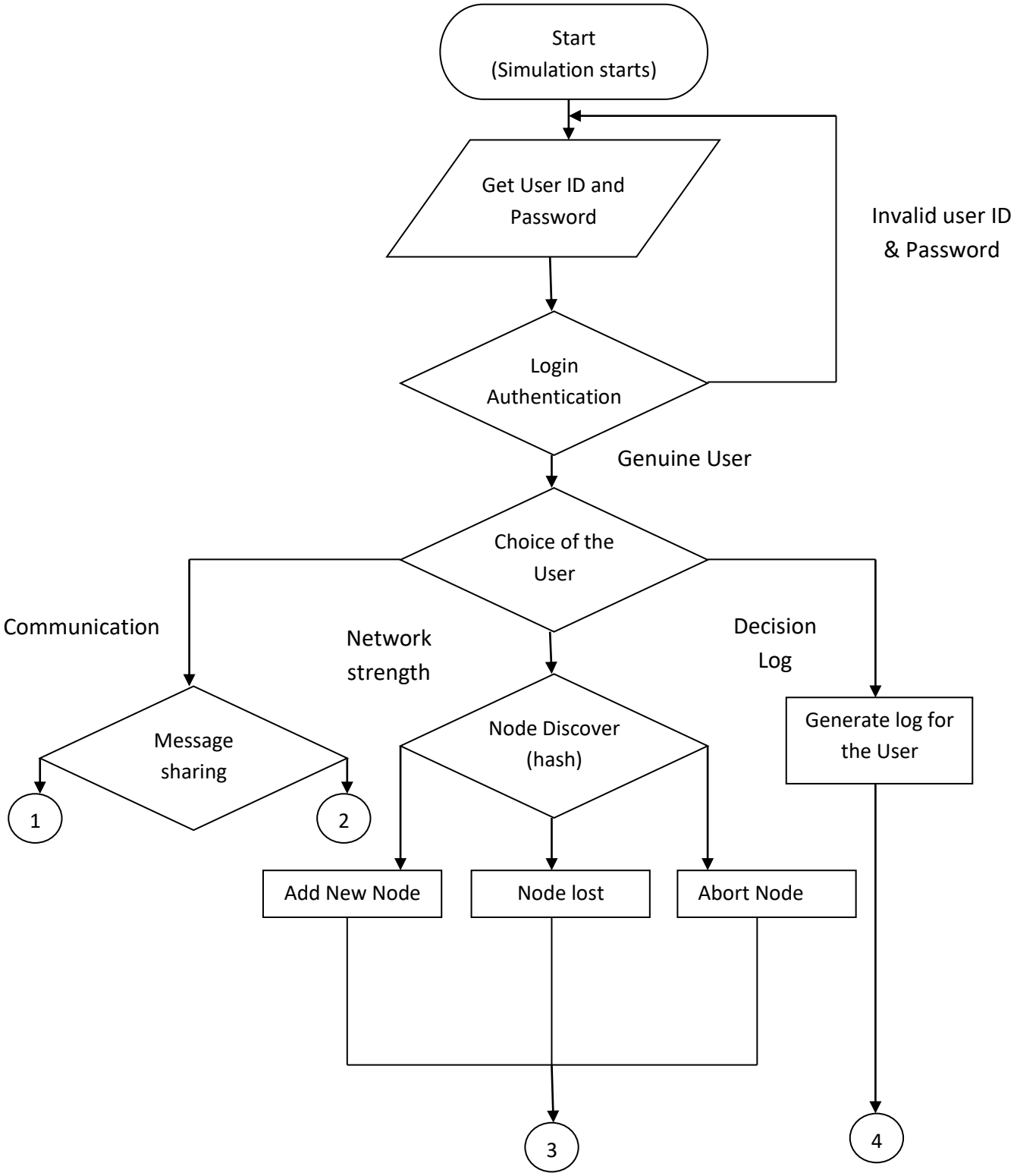
3.5.3.1. Run-Time Environment

A MIDlet is a J2ME application designed to operate on an MIDP small computing device. A MIDlet is defined with at least a single class that is derived from the javax.microedition.midlet.MIDlet abstract class.

Inside the Java Application Descriptor File,

- MIDlet-Name
- MIDlet-Version
- MIDlet-Vendor
- MIDlet-n
- MIDlet-Jar-URL.

3.6. FLOW DIAGRAM



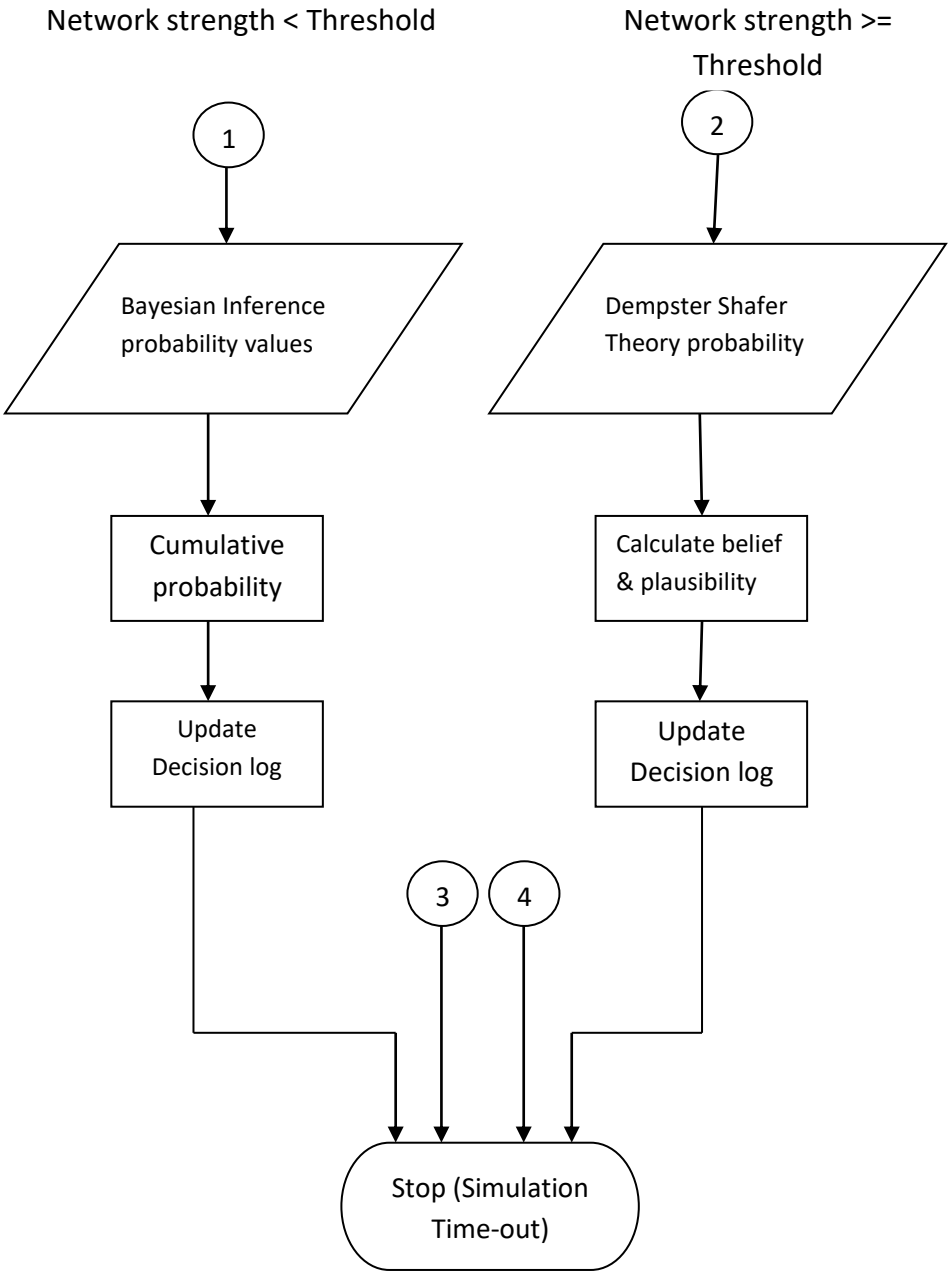


Figure 3.1: Flowchart

CHAPTER IV

RESULTS AND ANALYSIS

4.1 Simulation result analysis

In this section, the performance of the decision logics which is carefully simulated is evaluated. Usually a vehicle computes the combined trust in an event based on the reports it receives from distinct vehicles. The four decision logics: MTR, WV, BI, and DST have been compared against the basic majority voting scheme. The example scenario with two events α1 and α2 as described earlier.

First, trust decisions based on MTR are the most sensitive to different parameters since the MTR is not corroborated by other vehicles in this case.

Second, under realistic conditions, the other three decision logics outperform both majority voting and MTR. Third, there is no clear winner among these decision logics as each performs best in certain scenarios.

Based on the above results, it's obvious that there is no clear winner among the decision logics that fits best all scenarios.

But it's easy to elaborate several guidelines for the evaluation of data-centric trust:

- If the uncertainty in the network is low, BI is the most resilient to false reports.
- To avoid the case of few highly trustworthy false reports (Fig. 2(b)), the decision of BI should be positioned with respect to another logic, such as DST or WV, and the most conservative value (i.e., the one that yields the lowest probability of attack success) should be taken.
- The availability of prior knowledge can further improve the resilience of BI.
- If the uncertainty in the network is high, DST performs consistently better than other methods (MTR does not always yield better results).

4.2. ADVANTAGES OF THE PROPOSED MODEL

4.2.1. Improved Network Performance

The implementation of the data-centric trust establishment reduces the number of retransmissions that are required in case of aborted transmission. As

the mobile node is able to get the data from the data Repository itself, there is severe reduction in the amount of delay that occurs. It helps in eliminating the round trip from the base station to the global server and then the way back.

4.2.2. Improved Consistency

The vehicular ad-hoc framework helps in improvising the data consistency among the data repository by implementing the set of dependencies. This helps in eliminating the request to be routed through the server every time. The updating of the decision logic repository is carried out by means of time-stamps which in turn help in eliminate the problem of timed out or invalid data.

4.2.3. Reduced Load On The Server

The load on the server is greatly reduced because of this method. The server does not have to individually handle the request of each and every mobile node. It just has to keep track of the EI repositories that had obtained the data from it and it has to periodically update it.

4.2.4 Reduced Computing Time

Due to the ephemeral nature of the vehicular ad-hoc networks the need for computing the decisions at the faster rate is felt. Also using data centric approach with probability calculation can reduce the computing need of the node thus helping us to achieve faster decisions.

4.2.5 Eliminating Encryption Overhead

Since the decisions are made based on the basis of probability distribution the need for encrypting the whole transaction is greatly reduced and hence this is an added advantage thus helping the node to come up with faster decisions with little computing time spent in calculations.

4.3. SNAPSHOTS:

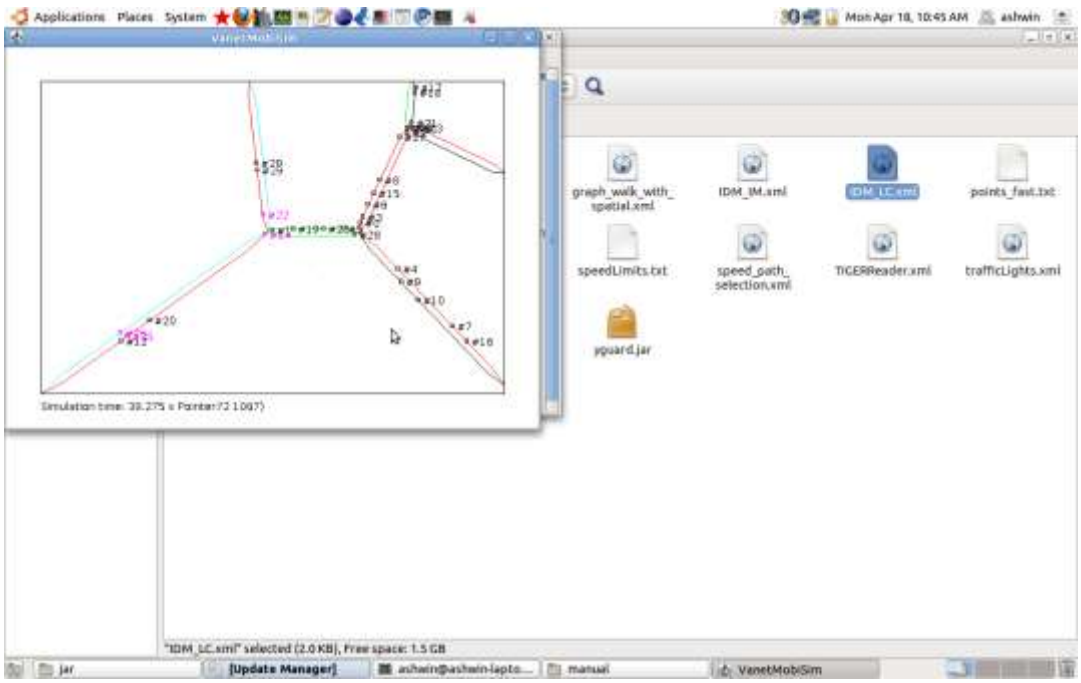


Figure 4.1: Using Simple Voting

Using this simulation result the effect of **simple voting** as decision logic has been studied under varied road map condition.

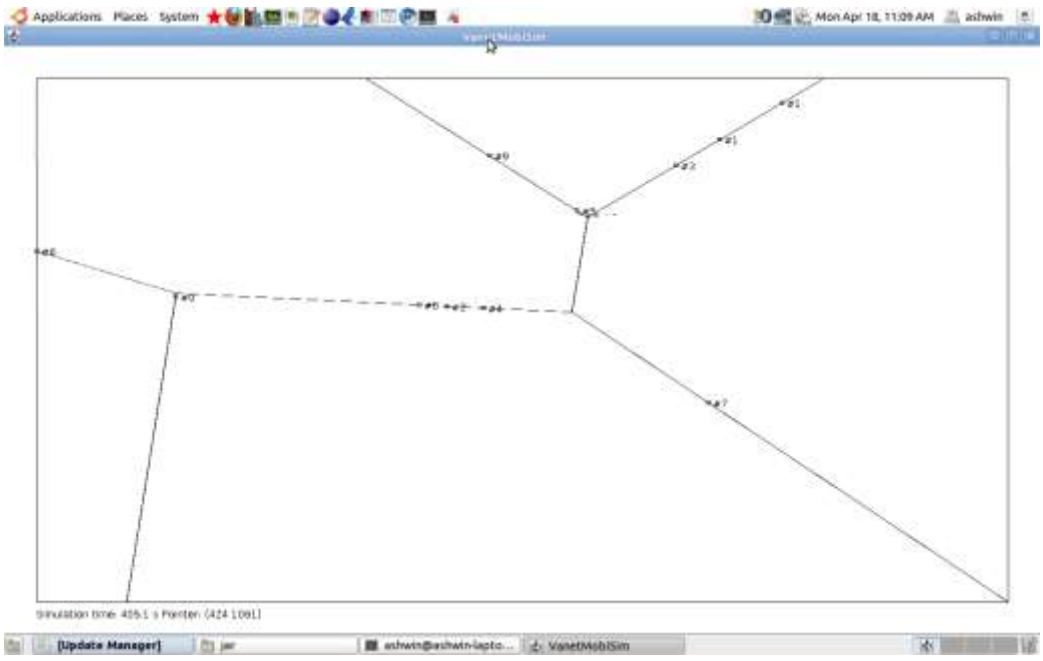


Figure 4.2.: Mobility model simulation

Using this, high mobility model of road-side condition is checked along with adversaries to evaluate simulation results.

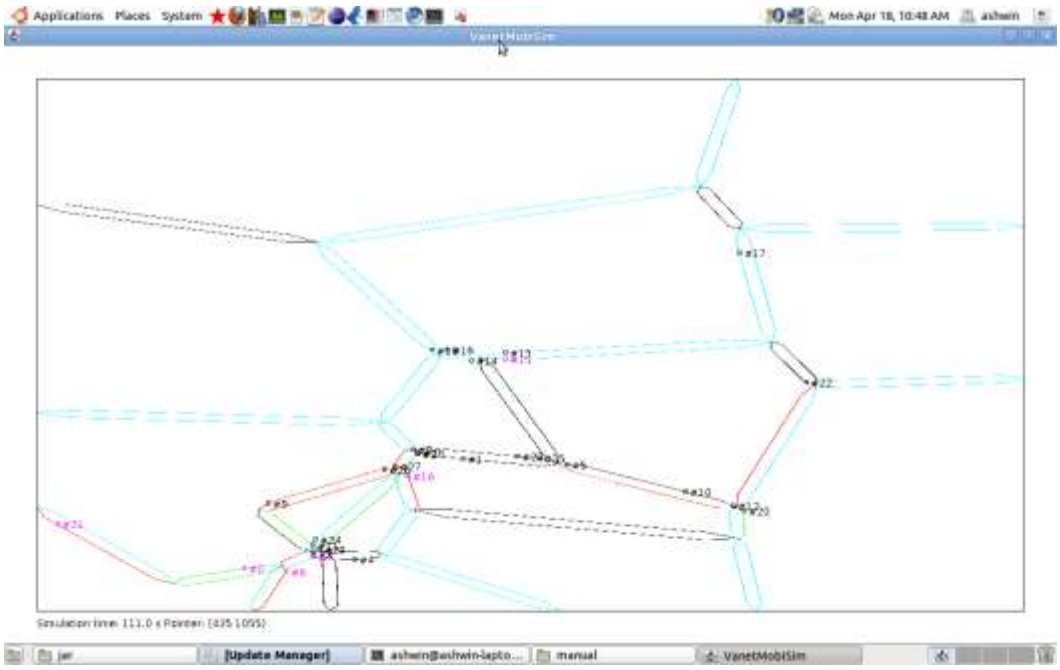


Figure 4.3: Highway - model

Sample high way model simulated using robust node modelling architecture to study our evaluation schemes.

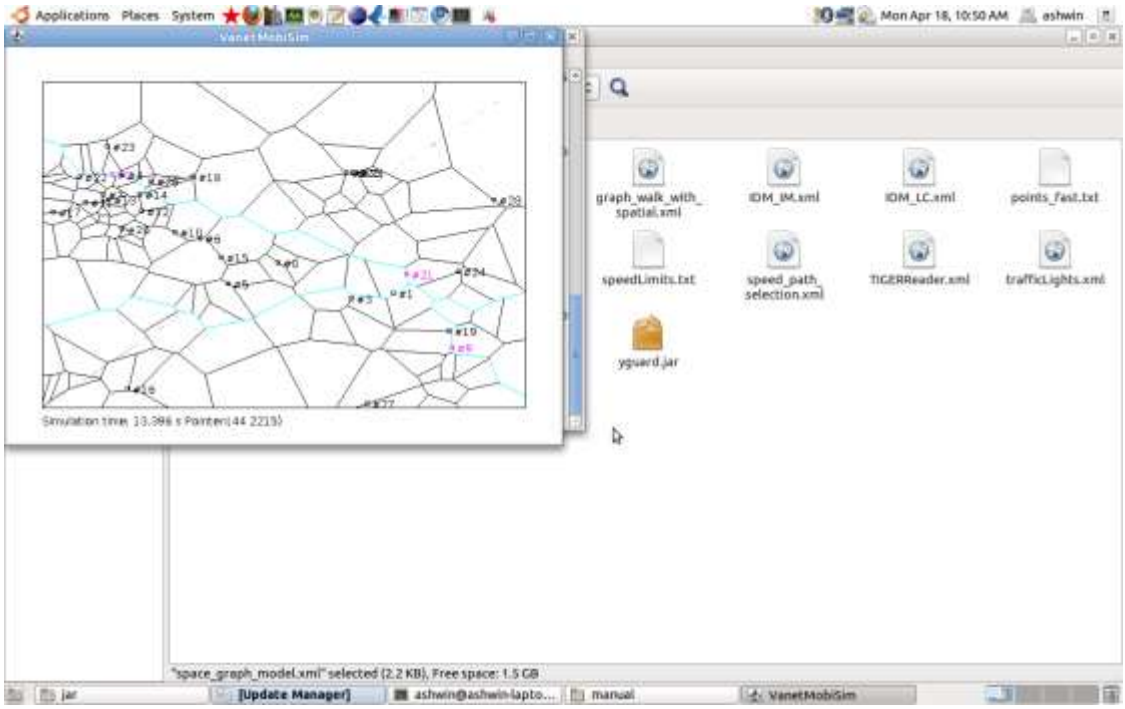


Figure 4.4: Using Bayesian Inference

Bayesian Inference technique & its vital participation in decision module is evaluated to model the highway simulation.

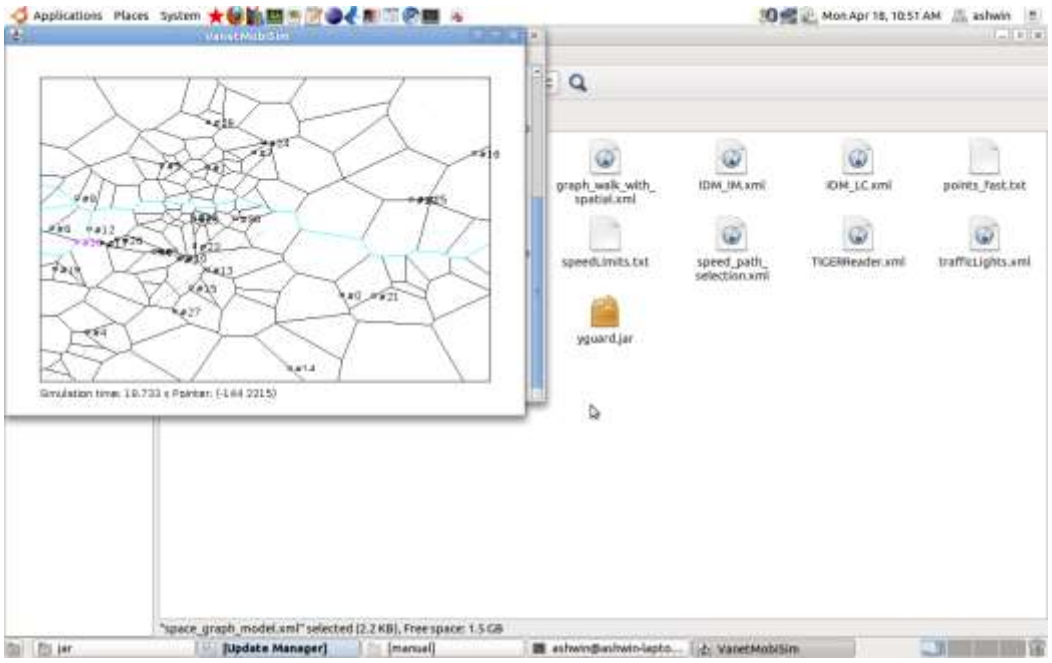


Figure 4.5: Broadway Street Model

Broadway Street model constructed using regular transition from canuMobiSim.

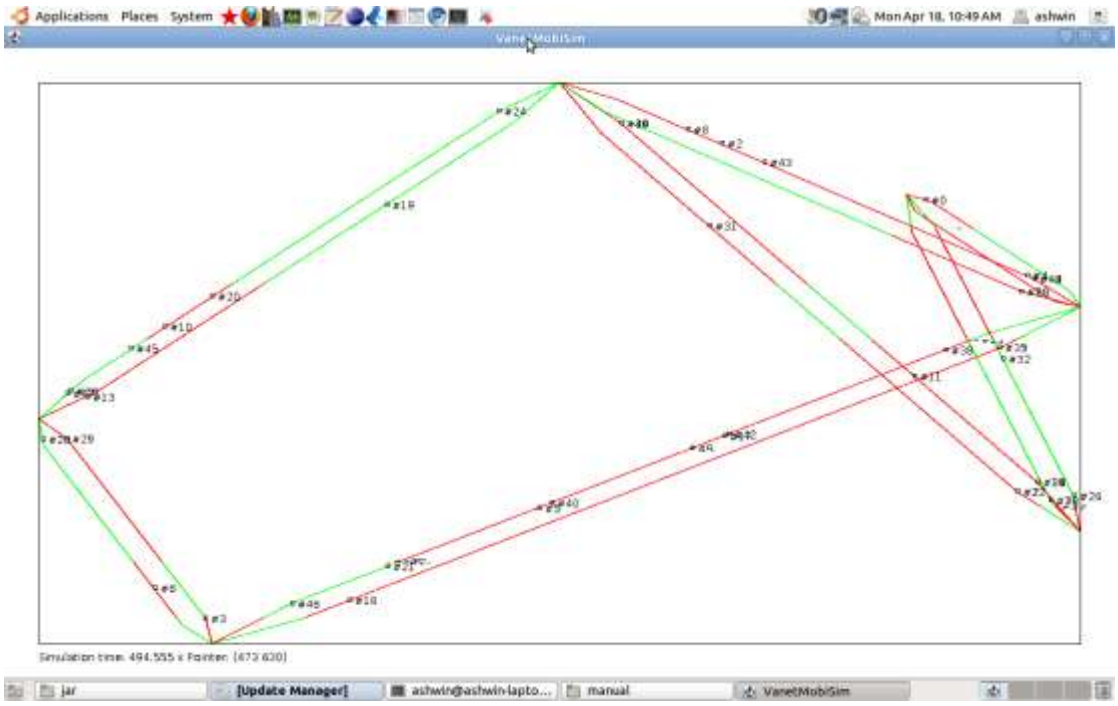


Figure 4.6: Combination of Dempster Shafer theory & Bayesian Inference

A proposal result to combine Bayesian Inference with Dempster Shafer theory based on the network density in current vicinity.

CHAPTER V
CONCLUSION

In this project, the notion of data trust has been established. Also ephemeral networks have been addressed that are very demanding in terms of processing speed. A general framework has been designed by applying it to vehicular networks that are both highly data-centric and ephemeral. Data reports with corresponding trust levels using several decision logics, namely weighted voting, Bayesian inference, and Dempster-Shafer Theory has been evaluated.

Simulation results show that Bayesian inference and Dempster-Shafer Theory are the most promising approaches to evidence evaluation, each one performing best in specific scenarios. More specifically, Bayesian inference performs best when prior knowledge about events is available whereas Dempster-Shafer Theory handles properly high uncertainty about events. In addition, the local processing approach based on either one of the above techniques converges to a stable correct value, which satisfies the stringent requirements of a life-critical vehicular network.

Since the whole decision function modules cannot be based on a single methodology a combination of Bayesian Inference and the Dempster Shafer has been established in which the decision function add weightage to Bayesian Inference in case of high density network and the other Dempster Shafer theory in the case of low density network. The simulation results show that the resulting decisions made using this combo is more robust than existing mechanisms.

REFERENCES:

1. Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems – 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ASTM E2213-03, 2003.

2. IEEE P1609.2 Version 1 - Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. In development, 2006.[3] M. Abdel-Aty, A. Pande, C. Lee, V. Gayah, and C. Dos Santos. Crash risk assessment using intelligent transportation systems data and realtime intervention strategies to improve safety on freeways. Journal of Intelligent Transportation Systems, 11(3):107–120, Jul. 2007

3. Mobile Computing Networks
   Dr.Yatindra Nath Singh
4. On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks Maxim Raya, Panagiotis Papadimitratos, Virgil D. Gligory, Jean-Pierr Hubaux School of Computer and Communication Sciences EPFL, Switzerland
5. Efficient VANET-based Traffic Information Sharing using Buses on Regular Routes Tomoya Kitani∗§, Takashi Shinkawa∗, Naoki Shibata†§, Keiichi Yasumoto∗§, Minoru Ito∗ and Teruo Higashino Graduate School of Information Science, Nara Institute of Science and Technology, Japan
6. www.ieee.org

7. www.wikipedia.org

8. www.roseindia.net

9. www.sun.java.com