EFFICIENT AND SECURE DELIVERY OF AREA-PERSISTENT SAFETY MESSAGES IN VEHICULAR AD HOC NETWORKS

by
Can Berk Güder

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

August 2009

EFFICIENT AND SECURE DELIVERY OF AREA-PERSISTENT SAFETY MESSAGES IN VEHICULAR AD HOC NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi ……
(Thesis Supervisor)

Assoc. Prof. Dr. Özgür Erçetin ……
(Thesis Supervisor)

Assoc. Prof. Dr. Erkay Savaş ……

Asst. Prof. Dr. Hasan Sait Ölmez ……

Assoc. Prof. Dr. Özgür Gürbüz ……

DATE OF APPROVAL: ……

EFFICIENT AND SECURE DELIVERY OF AREA-PERSISTENT SAFETY MESSAGES IN VEHICULAR AD HOC NETWORKS

Can Berk Güder

CS, MS Thesis, 2009

Thesis Supervisors: Assoc. Prof. Dr. Albert Levi, Assoc. Prof. Dr. Özgür Erçetin

Keywords: Vehicular ad hoc networks, driving safety, area-persistent messages, probabilistic algorithms, geocast routing

Abstract

In this thesis, we propose an adaptive mechanism for the delivery of safety messages in vehicular networks in an authenticated and privacy-preserving manner. The traditional approach to message delivery for driving safety applications running on vehicular ad hoc networks (VANETs) has been to increase redundancy, often at the sake of other applications running on the network. We argue that this approach does not accommodate the traffic conditions of crowded cities like İstanbul, and present a probabilistic method for the dissemination of area-persistent safety messages in infrastructure-less vehicular networks that dynamically adapts itself to changing road conditions. Our proposed protocol utilizes short group signatures for privacy-preserving authentication, and keyed-Hash Message Authentication Codes (HMACs) with one-way hash chains to decrease computational load on Onboard Units (OBUs).

We also introduce a vehicular mobility model that creates scenarios of high-speed traffic on crowded highways based on realistic assumptions, and measure the performance of the proposed protocol using scenarios generated by this model. Our simulations show that the proposed method decreases network traffic by up to 82% and shortens delivery delays by up to 13% when compared to non-probabilistic methods in highway scenarios with medium to high vehicle density.

TASARSIZ ARAÇSAL AĞLARDA BÖLGEDE KALICI EMNİYET MESAJLARININ VERİMLİ VE GÜVENLİ DAĞITIMI

Can Berk Güder

CS, Master Tezi, 2009

Tez Danışmanları: Doç. Dr. Albert Levi, Doç. Dr. Özgür Erçetin

Anahtar Kelimeler: Tasarsız araçsal ağlar, sürüş emniyeti, bölgede kalıcı mesajlar, olasılıklı algoritmalar, bölgesel yönlendirme

Özet

Bu tezde, araçsal ağlarda emniyet mesajlarının dağıtımı için değişen koşullara uyum sağlayan bir mekanizma öneriyoruz. Tasarsız araçsal ağlar üzerinde çalışan sürüş emniyeti uygulamaları için mesaj dağıtımına geleneksel yaklaşım, ağ üzerinde çalışan diğer uygulamaların pahasına da olsa, artıklığı arttırmak şeklinde olmuştur. Bu yaklaşımın, İstanbul gibi kalabalık şehirlerin trafik koşullarına uyum sağlamadığını savunuyor, ve altyapısız araçsal ağlarda bölgede kalıcı emniyet mesajlarının yayılımı için değişen yol koşullarına uyum sağlayan, olasılıklı bir yöntem sunuyoruz. Önerdiğimiz bu protokol, mahremiyeti koruyan kimlik denetimi için kısa grup imzalarından, ve araç üstü birimlerdeki işlem yükünü azaltmak için anahtarlı-Özet Mesaj Doğrulama Kodları ve tek yönlü özet zincirlerinden faydalanmaktadır.

Kalabalık otoyollardaki hızlı trafik senaryolarını, gerçekçi varsayımlara dayanarak yaratabilen bir taşıt hareket modeli tanıtıyor, ve önerdiğimiz protokolün performansını bu model tarafından yaratılan senaryoları kullanarak ölçüyoruz. Simülasyonlarımız, önerilen yöntemin orta ve yüksek araç yoğunluğuna sahip otoyol senaryolarında, olasılıklı olmayan yöntemlerle karşılaştırıldığında ağ trafiğini %82'ye kadar azalttığını ve dağıtım sürelerini %13'e kadar kısalttığını gösteriyor.

Acknowledgements

I would like to express my sincerest gratitude to my advisor Albert Levi not only for his guidance of this work, but also for his continuous support, constant encouragement and endless patience throughout the past six years. He has been a great mentor in all matters, academic and personal, and I feel truly honored to have been his student.

I am greatly indebted to my co-advisor Özgür Erçetin, whose valuable advice and dedication to perfection played an important part in the completion of this thesis.

Special thanks are due to Erkay Savaş for taking an interest in this study, parts of which would have been incomplete without his suggestions.

Finally, I would like to thank my thesis committee members Hasan Sait Ölmez and Özgür Gürbüz for their valuable review and comments.

TABLE OF CONTENTS

1. INTRODUCTION 1

2. LITERATURE SURVEY 3

   2.1 Literature on Mobility 3

   2.2 Literature on Routing 4

      2.2.1 Broadcast Routing 4

      2.2.2 Geocast Routing 5

   2.3 Literature on Privacy and Security 7

3. SYSTEM MODEL 10

   3.1 Road Network 10

   3.2 Vehicles 11

   3.3 Events 11

   3.4 Regions 12

      3.4.1 Critical Regions 12

      3.4.2 Message Regions 12

      3.4.3 Inspection Regions 13

   3.5 Messages 13

      3.5.1 Beacons 13

      3.5.2 Key Disclosure Messages 14

      3.5.3 Virtual Sign Messages 15

   3.6 DSRC and WAVE 16

4. MOBILITY 18

   4.1 Vehicle Density and Speed 19

5. PRIVACY AND SECURITY 22

5.1 Requirements 22
5.2 Digital Signatures 23
5.3 Security Model 24
5.4 Location Privacy 28
6. THE ADAPTIVE-P ALGORITHM 29
6.1 Probability Analysis 31
6.2 Flooding and Constant-p Algorithms 33
7. PERFORMANCE EVALUATION 34
7.1 Simulation Setup 34
7.1.1 Latency Due To Cryptographic Operations 35
7.1.2 Antennas 35
7.2 Performance Metrics 35
7.2.1 Awareness Delay 35
7.2.2 Message Traffic 36
7.2.3 Delivery Rate 36
7.2.4 Number of Events Received 37
7.3 Results 37
7.3.1 Manhattan Scenarios 42
7.3.2 Analysis of System Parameter k 45
8. SUMMARY AND CONCLUSIONS 48
9. APPENDIX 50
10. BIBLIOGRAPHY 52

LIST OF TABLES

Table 3.1: Channel allocation for WAVE 17

Table 4.1: *mugen* parameters for all three scenario sets 20

Table 6.1: Overall average probabilities for all three scenario sets 31

Table 7.1: Simulation Results for ρ_v = 2600 veh./h, μ_s = 30 m/s 37

Table 7.2: Simulation Results for ρ_v = 8333 veh./h, μ_s = 20 m/s 38

Table 7.3: Simulation Results for ρ_v = 15000 veh./h, μ_s = 30 m/s 38

Table 7.4: *mugen* parameters for the Manhattan scenario set 43

Table 7.5: Simulation Results for the Manhattan scenario set 43

Table 7.6: Analysis of *k* for ρ_v = 2600 veh./h, μ_s = 30 m/s 45

Table 7.7: Analysis of *k* for ρ_v = 8333 veh./h, μ_s = 20 m/s 45

Table 7.8: Analysis of *k* for ρ_v = 15000 veh./h, μ_s = 30 m/s 46

Table 7.9: Analysis of *k* for the Manhattan scenario set 46

LIST OF FIGURES

Figure 2.1: Different routing protocols 6

Figure 3.1: Road network 11

Figure 3.2: Events 11

Figure 3.3: Structure of a beacon 14

Figure 3.4: Structure of a Virtual Sign message 15

Figure 3.5: The OSI stack for ITS 16

Figure 4.1: Average number of vehicles in the network 21

Figure 6.1: Average broadcast probability (p_b) for all three scenario sets 32

Figure 6.2: Average forwarding probability (p_f) for all three scenario sets 32

Figure 7.1: Delivery rates for ρ_v = 2600 veh./h, μ_s = 30 m/s 39

Figure 7.2: Delivery rates for ρ_v = 8333 veh./h, μ_s = 20 m/s 39

Figure 7.3: Delivery rates for ρ_v = 15000 veh./h, μ_s = 30 m/s 40

Figure 7.4: Average number of events received for ρ_v = 2600 veh./h, μ_s = 30 m/s 41

Figure 7.5: Average number of events received for ρ_v = 8333 veh./h, μ_s = 20 m/s 41

Figure 7.6: Average number of events received for ρ_v = 15000 veh./h, μ_s = 30 m/s 42

Figure 7.7: Manhattan road network 43

Figure 7.8: Delivery rates for the Manhattan scenario set 44

Figure 7.9: Average number of events received for the Manhattan scenario set 44

LIST OF ALGORITHMS

Algorithm 5.1: INITIALIZE-HASH-CHAIN() 25

Algorithm 5.2: SEND-BEACON() 26

Algorithm 5.3: SEND-KEY-DISCLOSURE(*key, seq*) 26

Algorithm 5.4: RECEIVE-BEACON(*b*) 27

Algorithm 5.5: RECEIVE-KEY-DISCLOSURE(*kd*) 27

Algorithm 6.1: RECEIVE-VIRTUAL-SIGN(*m*) 29

Algorithm 6.2: DETECT-EVENT(*e*) 30

Algorithm 6.3: PROCESS-BEACON(*b*) 31

Algorithm A.1: MUGEN(*P, R, G,* t_{sim}, ρ_v, σ_a, μ_s, σ_s) 50

Algorithm A.2: FIND-PATHS(P, R) 50

Algorithm A.3: GENERATE-MOVEMENT(*S, T,* ρ_v, σ_a, μ_s, σ_s) 50

Algorithm A.4: TRIM(*V,* t_{trim}) 51

Algorithm A.5: ADD-COLLISIONS(*V, G,* t_{sim}) 51

Chapter 1

INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) are a form of Mobile Ad-hoc Networks (MANETs) where the nodes in the network are the vehicles on the road. VANETs provide communication among vehicles on the road, or between vehicles and a roadside infrastructure, with the purpose of increasing the safety and comfort of drivers and passengers alike. The applications running on VANETs are therefore commonly classified as (driving) safety applications and comfort applications. The first category might include applications like Electronic Brake Lights (EBL) or radar-like applications, while the second category includes Internet access, online gaming, etc.

Driving safety applications are arguably the most important applications running on a vehicular network. One such application could be a "Virtual Traffic Sign" application that provides the same function as physical traffic signs: providing drivers with information regarding road conditions and emergency situations. Virtual traffic signs, unlike physical traffic signs, can be generated dynamically by the vehicles using the road, thereby providing users with unsurpassed flexibility.

The main challenge with virtual traffic sign applications on vehicular networks without a reliable infrastructure (or none at all) is keeping the virtual traffic sign alive for as long as the information conveyed is current and relevant. The traditional approach to this challenge has been to increase redundancy, even at the sake of other applications running on the network.

In this thesis, we show that this approach does not accommodate the extreme traffic conditions of crowded cities like İstanbul, resulting in higher delays and lower delivery rates. We present a probabilistic method of disseminating virtual traffic sign messages that decreases network traffic by up to 82% and shortens delivery delays by up to

13% when compared to non-probabilistic approaches in highway scenarios with low or medium vehicle density.

To be able to clearly demonstrate our contribution, we also introduce a new mobility generator called *mugen* that creates mobility scenarios based on realistic assumptions.

Finally, we propose the use of keyed-Hash Message Authentication Codes (HMACs), one-way hash chains and group signatures for an efficient security framework that provides message authentication and conditional privacy.

The rest of the thesis is organized as follows. We present an overview of previous research in Chapter 2. In Chapter 3, we describe our system model. In Chapter 4, we explain our mobility model, and introduce our mobility generator, *mugen.* In Chapter 5, we describe our security and privacy framework. In Chapter 6, we introduce our main contribution, the Adaptive-*p* algorithm. In Chapter 7, we evaluate the performance of the proposed algorithm. We conclude the thesis in Chapter 8.

Chapter 2

LITERATURE SURVEY

2.1 Literature on Mobility

Node movement is one of the most important differences between vehicular networks and other wireless ad-hoc networks. In wireless sensor networks (WSNs) for example, nodes are typically considered stationary[1]. In wireless mesh networks, node movement is little (if any) and slow.

In VANETs, node movement is continuous and usually very fast, as compared to other forms of MANETs. Due to the unique nature of node movement, different mobility models tend to give remarkably different results in VANET simulations, making the use of a realistic mobility model vital to any study of these networks.

The most basic approach to node mobility is the Random Waypoint (RWP) model [1], where nodes move between randomly selected points in space at a uniform speed. In [2], Choffnes and Bustamante demonstrate that this model fails to produce realistic results.

In [3], Saha and Johnson present a more realistic model based on the RWP model. They use real road maps from the U.S. Census Bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) system, and develop a mobility generator that has vehicles moving on the shortest path between two random points on the road network at a uniform speed. In [2], Choffnes and Bustamante present an even more realistic model that considers the number of lanes on each road, inter-vehicular distance and traffic congestion. We classify these two mobility models as *enhanced RWP* models.

---

[1] Possibly except for a few mobile nodes or coincidental movement caused by environmental factors.

With the recent increase in GPS (Global Positioning System) usage in cars, real trace data became a viable alternative to synthesized vehicle movement. In [4], Li *et al.* use real GPS trace data gathered from the taxis of Shanghai, China. In [5], Füßler *et al.* use *reality-audited* movement data from German autobahns. In [6], Jetcheva *et al.* use GPS trace data of the fleet of city buses in Seattle, WA.

Note that in [4] and [6], only a subset of the vehicles occupying the road network is traced, and that in [5] a synthesized model is based on GPS trace data. Tracing all vehicles is impractical, if not impossible.

As an alternative to GPS tracing, Raney *et al.* [7] introduce a *Multi-agent Microscopic Traffic Simulator (MMTS)* that runs on a Beowulf cluster and simulates the vehicle traffic on all of Switzerland's roads. To the best of our knowledge, the MMTS is the most realistic synthesized vehicular mobility model proposed so far. However, we only consider short simulations of highway segments, while the MMTS creates real-time simulations of whole cities or countries, requiring immense computing power. This makes the use of such a simulator superfluous for the purposes of this thesis.

In Chapter 4, we present our own enhanced RWP model based on the work by Saha and Johnson in [3].

2.2 Literature on Routing

For the purposes of our thesis, we concentrate solely on broadcast and geocast routing protocols.

2.2.1 Broadcast Routing

Broadcasting is a popular routing method in VANETs, and the simplest broadcasting protocol is flooding, where each vehicle re-broadcasts each message it receives. While flooding works well for small networks, network traffic increases exponentially with the number of nodes, and performance degrades quickly.

A number of attempts have been made to develop a high-performance flooding-like broadcast algorithm. In [8], Sun *et al.* propose the *Vector-based Tracking Detection* (V-TRADE) and *History-enhanced V-TRADE* (HV-TRADE) algorithms that take advantage of location information to divide neighbor nodes into forwarding groups. This way, the number of re-broadcasting nodes is limited, and bandwidth utilization is notably improved.

In the *Urban Multi-Hop Broadcast* (UMB) protocol [9], the originating node picks a single neighbor node (the farthest node) to re-broadcast the message. Compared with two topology-unaware flooding protocols (802.11-distance and 802.11-random), UMB shows significant improvements on success rate, load and dissemination speed.

Another approach to limiting the number of re-broadcasts is through the clustering of nodes. In [10], Durresi *et al*. propose ICE, which organizes vehicles into clusters (*cells*) according to their locations. The vehicle closest to the cluster center is self-elected as the *cell leader*, and handles all inter-cluster communication. When compared with DOLPHIN [11], ICE shows significant improvements in delay and load.

2.2.2 Geocast Routing

Geocast routing [12] is, in essence, broadcast routing limited to a geographical region, usually called the *geocast region* or the *Zone of Relevance (ZoR)* [13]. In this sense, broadcast routing can be seen as a special case of geocast routing where the geocast region is the whole network.

Figure 2.1 shows the difference between unicast, broadcast and geocast routing. In this figure, the black rectangles are the source vehicles, and the dark gray rectangles are the destination vehicles. The shaded area in the third subfigure is the geocast region.

In [14], Briesemeister *et al.* propose a basic geocast scheme where the node that is furthest from the source node rebroadcasts the message. The number of rebroadcasts is limited using a hop count. In this protocol, no distinct geocast region information is contained in the messages, but the location information of the message originator is, and the geocast region is assumed to be centered on this location.

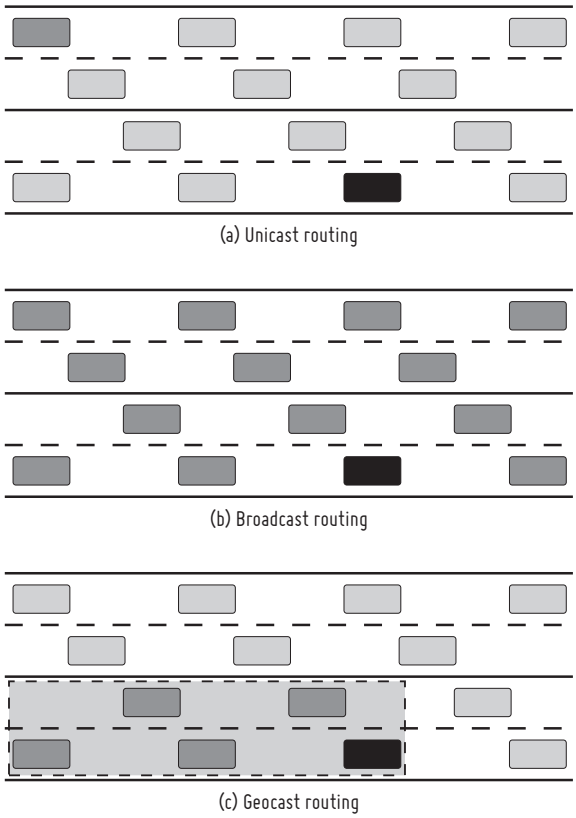

Figure 2.1: Different routing protocols

Bachir and Benslimane [15] propose the *Inter-Vehicle Geocast (IVG)* protocol, the idea behind which is very similar to that behind [14], but receiving vehicles drop certain messages if they deem irrelevant to themselves, possibly further reducing the number of re-broadcasts.

Maihöfer and Eberhardt [16] propose a *cached greedy geocast* protocol. In their approach, greedy routing (where packets are always forwarded to the neighboring node that is closest to the packet's final destination) is enhanced by adding a cache at the routing layer, which stores messages that cannot be forwarded due to local maxima. Simulation results show that the proposed caching mechanism increases delivery rates by up to 300 percent.

In [17], Maihöfer *et al.* propose three approaches for an *abiding geocast* protocol in which messages are delivered not only to the vehicles inside the geocast region at the time of delivery, but also to those vehicles that have been or will be inside the geocast region for some time during a predefined message lifetime. The three proposed approaches are: (1) storing the messages on intfrastructure-provided servers, (2) storing the messages on elected nodes, and (3) storing the messages on all nodes.

The routing requirements of our virtual traffic sign application are quite similar to the requirements in [17]. However, we assume an infrastructureless network, therefore the messages must be stored on the nodes. Instead of electing storage nodes or storing the messages on all nodes, we let the nodes decide which messages are relevant, and should be stored. This is similar to the approach in [15].

2.3 Literature on Privacy and Security

Symmetric cryptosystems, such as the AES [18], are the de-facto choice for resource-constrained systems with a limited number of nodes (such as WSNs) for their performance and ease of use. In these systems, a key pre-distribution scheme [19] is usually employed, where nodes are dealt random encryption keys before being deployed. Each node must be given a certain number of keys (that increases with the number of nodes in the network) to guarantee connectivity, and some of these keys must be revoked if a node is compromised. The sheer number of nodes in VANETs makes it infeasible to use such a key pre-distribution scheme. Also, since node movement is very fast in VANETs, connectivity might only be maintained for short periods of time in certain situations, and therefore the nodes might not have enough time to agree on a mutual encryption key. For these reasons, public-key cryptography is usually preferred to symmetric cryptography in VANETs[2].

The concept of public-key cryptography, put forward by Diffie and Hellman in 1976 [21], is considered to be the most important breakthrough in the history of cryptography [22]. The development of public-key cryptography led to two important applications: asymmetric encryption and digital signatures. In our work, we focus on the latter.

After Diffie and Hellman's work, a number of public-key cryptosystems emerged. Among these cryptosystems, the RSA [23] and ElGamal [24] cryptosystems have become the most popular and widely adopted [25]. The security of most of these early public-key cryptosystems depend on the integer factorization problem (RSA) or the dis-

[2] Some hybrid approaches, such as the *Secure Anonymous Broadcasting (SAB)* protocol [20], have also been proposed.

crete logarithm problem on finite cyclic groups (ElGamal, Diffie-Hellman key exchange, DSA).

In the late 80s, Miller [26] and Koblitz [27] independently suggested the use of elliptic curves in cryptography, which led to elliptic-curve public-key cryptosystems such as the Elliptic-curve Digital Signature Algorithm (ECDSA). Elliptic-curve cryptosystems, requiring much less storage space than cryptosystems like RSA or DSA[3], became especially popular in resource-constrained systems, such as embedded systems or WSNs.

Strong digital signatures provide authentication, data integrity and non-repudiation, but they do not provide anonymity unless they're used with anonymous key pairs. In [28], Raya and Hubaux proposed using a large number of anonymous key pairs at every node, but this approach shares the same practical problems with a symmetric key pre-distribution method [29]. In [29], Lu *et al.* propose *Efficient Conditional Privacy Preservation (ECPP)* protocol that relies on Roadside Units (RSUs) to provide short-time anonymous keys.

A number of group-based approaches were proposed to provide anonymity in vehicular networks. In [30], Wasef and Shen propose the *Privacy Preserving Group Communications Protocol for Vehicular Ad Hoc Networks (PPGCV),* which is based on the *GKMPAN* protocol by Zhu *et al.* [31]. Both works are mainly concerned with group rekeying.

Introduced by Chaum and van Heyst in [32], group signatures allow members of a predefined group to digitally sign messages on behalf of the whole group, providing anonymity for the signer. In the case of a dispute, a previously selected group manager can reveal the identity of a signature's originator.

To the best of our knowledge, the use of group signatures in a vehicular context was first proposed by Boneh *et al.* in [33]. However, their work concentrates on the development of a group signature scheme that creates shorter signatures compared to pre-

[3] ECDSA with a 160-bit key provides roughly the same security level as RSA or DSA with 1024-bit keys. [22] Signatures generated using DSA or ECDSA are also much shorter than signatures generated using RSA: about 320 bits for 160-bit ECDSA or 1024-bit DSA, compared to at least 1024 bits for 1024-bit RSA.

vious approaches, rather than the application of group signature schemes in vehicular contexts.

In [34], Lin *et al.* propose the *Group Signature and Identity-based Signature (GSIS)* scheme, where they employ the short group signature scheme proposed in [33] among vehicles, and identity-based signatures between vehicles and RSUs.

In [35], Lin *et al.* propose the *TESLA based Secure Vehicular Communication (TSVC)* scheme that employs the TESLA broadcast authentication protocol [36] with anonymous key pairs for the conventional digital signature scheme.

Our proposed security model, explained in Chapter 5, employs group signatures and the TESLA broadcast authentication protocol, and is mainly based on the work by Lin *et al.* in [35].

Chapter 3

SYSTEM MODEL

Our system model consists of four basic entity sets (events, regions, messages and vehicles) residing on a road network.

Vehicles are nodes traveling on the road network according to the mobility model. In compliance with the *Dedicated Short-range Communications (DSRC)* specifications [37], they periodically broadcast short messages called *beacons*.

Events are phenomena tied to certain parts of the road network called critical regions. When a vehicle's onboard sensors or driving safety mechanisms detect an event, the vehicle defines a zone of relevance for the detected event, and sends a *Virtual Sign* message to the vehicles inside this region. We call these regions *message regions.* Vehicles receiving a Virtual Sign message try to keep the contents of the message persistent in the message region defined by the originating vehicle.

Finally, we define an inspection region and limit some of our performance metrics to only the vehicles inside this region.

3.1 Road Network

Our road network is created to imitate a common highway segment with entrance and exit ramps on each side of the road.

Figure 3.1 shows the road network used in our simulations: a 3 kilometer-long highway with two entrance and two exit ramps, one critical region, one inspection region, and an example message region. The message region depicted is for a message broadcast at the point where the northern side of the road meets the critical region.

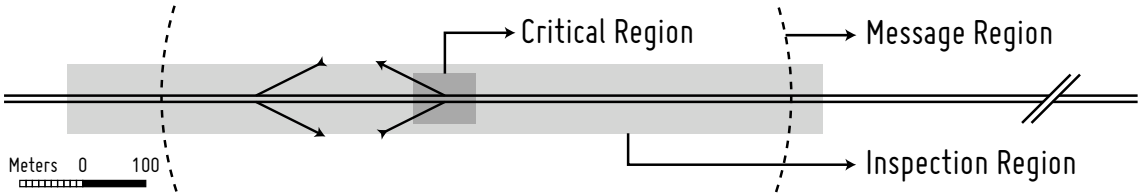

Figure 3.1: Road network

3.2 Vehicles

Vehicles in our system model are considered to be dimensionless points moving on the road network, according to the scenario generated by the mobility generator. There is no distinction between different vehicle types such as motorcycles, cars and trucks. All vehicles are considered to be 1.5 meters tall, and antenna placement is done accordingly.

3.3 Events

An event is any phenomenon that causes a vehicle to emit a Virtual Sign message when observed by the vehicle's onboard sensors. These phenomena could be traffic jams, accidents, slippery road conditions, etc. Events are tied to regions on the road called critical regions, and each such region can have one or more events attached to it.

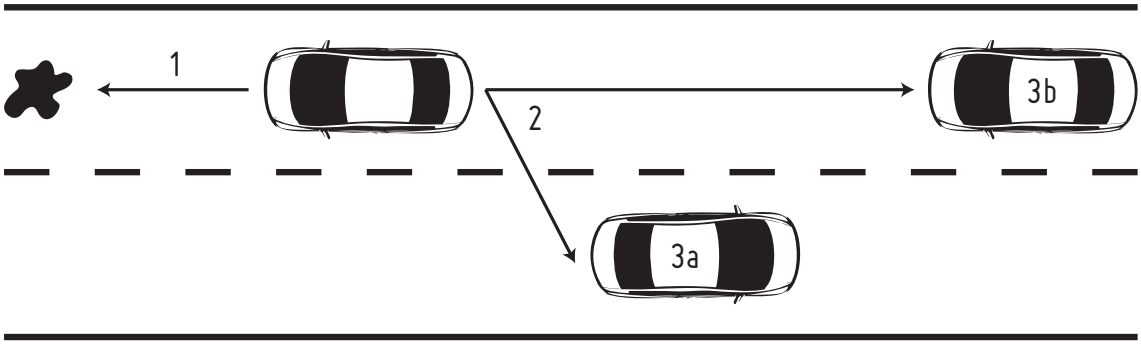

Figure 3.2: Events

Figure 3.2 shows the actions taken by the vehicles in the network when a vehicle observes an event. In this scenario, the vehicle's Electronic Stability Control (ESC) system detects the slippery road conditions and activates (1). The vehicle's Onboard Unit (OBU), connected to the ESC system, creates a Virtual Sign message and broadcasts this message (2). Receiving vehicles process and store this message for forwarding (3a) or direct usage (3b).

3.4 Regions

We define three types of regions: critical regions, message regions and inspection regions. We consider the first two region types to be part of any real-life application, but the third region type is only used for performance evaluation purposes.

3.4.1 Critical Regions

Critical regions represent parts of the road network associated with one or more events that approaching drivers should be aware of, such as construction or accident sites, roadblocks, detours, slippery turns, etc. When a vehicle enters a critical region, its onboard sensors detect one or more phenomena, causing the vehicle to broadcast Virtual Sign messages.

In our simulations, we define a single critical region with 5 events.

3.4.2 Message Regions

A message region is the zone of relevance for a single message, defined by the originator of the message. When a vehicle detects an event, it creates a new message and makes a decision about the area for which the information contained in the message is relevant. This area is called the message region for that message.

Message regions play an important role in our message forwarding mechanism: vehicles forward messages only when themselves or the intended recipient is inside the

message region for the message in question. Vehicle behavior related to message regions is further explained in Chapter 6.

In our simulations, message regions are circular regions centered on the point where the vehicle first detects the event, and have a radius of 500 meters[4].

3.4.3 Inspection Regions

An inspection region represents the area in which the messages generated by reason of a critical region are relevant. In this sense, an inspection region is to a critical region what a message region is to an event.

Unlike critical regions or message regions, inspection regions are not part of the simulations, but rather the performance evaluation process. Therefore, entering or leaving an inspection region has no effect on a vehicle's behavior.

In our simulations, we define a single inspection region that is 1100 meters long and 100 meters wide, covering all the entrance and exit ramps, the critical region, and 500 meters[4] of straight road segment on each side of the critical region. We then base three of the four performance metrics defined in Section 7.2 on this inspection region.

3.5 Messages

3.5.1 Beacons

Beacons are one-hop broadcast messages sent by each vehicle in the network at regular intervals, containing information such as the current position, direction and speed of the vehicle. According to the DSRC specifications, beacons must be sent every 100 to 300 ms. In our simulations, beacons are sent every 200 ms.

---

[4] 500 meters is the decision sight distance [38] for a vehicle traveling at 120 km/h, the speed limit for cars on Turkish highways.

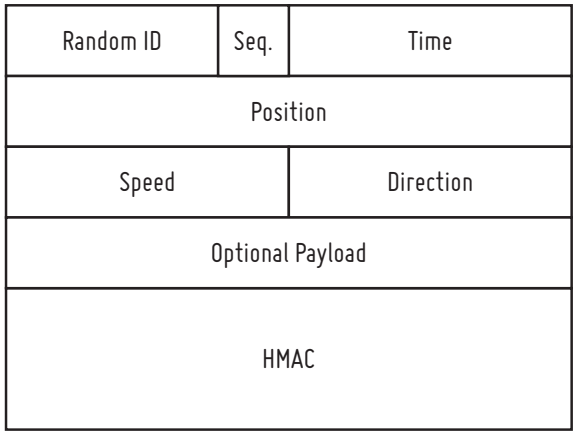

Figure 3.3: Structure of a beacon

Figure 3.3 shows the beacon structure used in our application. It contains a random node ID, an increasing sequence number, a timestamp, and the position, speed and direction information of the sending vehicle, obtained from the GPS system. The final field in a beacon is a keyed-Hash Message Authentication Code (HMAC) [39] of the beacon's contents.

The primary purpose of beacons is to provide environmental awareness, but other information can be piggy-backed on beacons. For example, in our application, we include the fingerprints of the previously received Virtual Sign messages as a payload in beacons. When a vehicle receives a beacon from a neighboring vehicle, it compares these fingerprints with the Virtual Sign messages it has stored before and forwards the messages that the beacon's sender has not yet received, but might be interested in. This behavior is further explained in Chapter 6.

3.5.2 Key Disclosure Messages

For beacons, we use an authentication mechanism based on the TESLA broadcast authentication protocol [36]. In the TESLA broadcast authentication protocol, messages contain an authentication code created using an HMAC algorithm, and the key for the HMAC algorithm is revealed in a second message called a key disclosure message.

Key disclosure messages are sent after a short delay δ following the beacon for which they reveal the HMAC key. This value is set to 20 ms in our simulations.

Besides the random node ID, the HMAC key and a sequence number, some key disclosure messages also contain a group signature. The purpose and contents of key disclosure messages is further explained in Chapter 5.

3.5.3 Virtual Sign Messages

Virtual Sign messages are broadcast when vehicles observe an event on the road. They contain the same basic information (time, position, speed and direction) as a beacon, but they also carry detailed information about the observed event.

Figure 3.4 shows the structure of a Virtual Sign message. The content and functionality of the time, position, speed and direction fields are the same as a beacon. The event type field contains the type of the observed event, such as a traffic accident or slippery road conditions. The message region field contains the message region associated with the observed event. The event details field is optional, and might include information such as measurements from onboard sensors, etc. Finally, the signature field contains a digital signature over the contents of the Virtual Sign message.

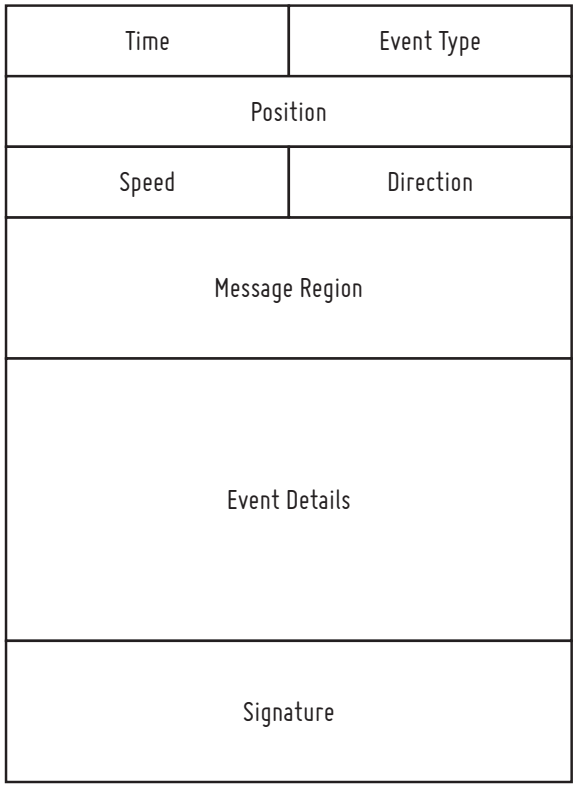

Figure 3.4: Structure of a Virtual Sign message

Note that Virtual Sign messages contain no ID information. Together with group signatures, this provides complete anonymity for these messages. This is further explained in Section 5.4.

3.6 DSRC and WAVE

The network stack for vehicular communications is defined by two sets of standards. The physical and data link layers are defined by the IEEE 802.11p draft standard [40], an amendment to the IEEE 802.11 standard [41] based on the DSRC specifications [37]. The upper layers are defined by the IEEE 1609 standards [42-45], also commonly known as WAVE (Wireless Access in Vehicular Environments). Figure 3.5 shows the OSI stack for ITS (Intelligent Transportation Systems).

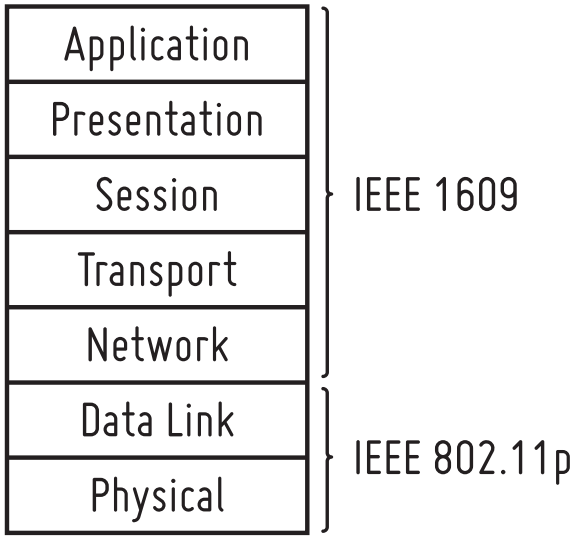

Figure 3.5: The OSI stack for ITS

DSRC devices operate in the licensed 5.9 GHz ITS band using one control channel dedicated to control frames, and six general-purpose service channels, for a total of seven 10 MHz channels. Unlike 802.11a/b/g/n stations, 802.11p stations use all the channels defined in the standard. Channel time is divided into 100 ms sync intervals that consist of a control interval and a service interval. During the control interval, all stations are required to tune to the control channel, on which high-priority frames are transmitted. After the control interval, stations can tune to service channels for the re-

mainder of the 100 ms. This allows the stations to achieve data rates of up to 27 Mbps, or 54 Mbps using the optional 20 MHz channels. Table 3.1 shows channel allocations for both modes of operation.

Table 3.1: Channel allocation for WAVE

Channel Number	Center Frequency (MHz)	10 MHz Mode	20 MHz Mode
172	5860	Service channel	Service channel
174	5870	Service channel	-
175	5875	-	Service channel
176	5880	Service channel	-
178	5890	Control channel	Control channel
180	5900	Service channel	-
181	5905	-	Service channel
182	5910	Service channel	-
184	5920	Service channel	Service channel

In vehicular networks, the nodes are highly mobile, and therefore the OBUs usually have very limited time for transmission. For this reason, any communication overhead must be minimized. The IEEE 802.11p MAC tries to achieve this by dropping MAC authentication, SSID association and beacon frames (not to be confused with DSRC beacons), allowing fully independent operation. Also, 802.11e Quality-of-Service (QoS) is supported with 4 priority levels defined in the standard. These changes allow the vehicles to achieve the sub-50 ms communication latencies required by the DSRC specifications.

The DSRC specifications also state that all DSRC devices must be able to achieve a Packet Error Rate (PER) of less than 10% for a Physical-layer Service Data Unit (PSDU) length of 1000 bytes at 85 miles per hour, and for a PSDU length of 64 bytes at 120 miles per hour. In [46], Gukhool and Cherkaoui compare IEEE 802.11a and 802.11p in VANET simulations, and conclude that using 802.11p decreases packet loss significantly. In our simulations, we assume the use of IEEE 802.11p for all communications.

Chapter 4

MOBILITY

In Section 2.1, we emphasized the importance of a realistic mobility model, and observed that the existing approaches to vehicle mobility fall under four main classes: Random Waypoint (RWP), enhanced RWP, GPS traces and micro-simulations. We also noted that the RWP model fails to generate realistic scenarios for vehicular networks, and that it is practically impossible to acquire GPS traces for all the vehicles on the road network.

Lacking the time and resources required for micro-simulations, we decided to use an enhanced RWP model. Of the two enhanced RWP models we inspected, STRAW [2] was not compatible with our simulator (ns2), and Saha and Johnson's model [3] had certain drawbacks. We therefore introduce our own mobility generator, *mugen*.

mugen is inspired by, and works in a similar fashion to, the Saha and Johnson model, but has several advantages:

- The whole scenario is generated at once, instead of 10-second steps, resulting in smoother vehicle movement.
- Instead of a fixed number of vehicles, normally distributed inter-arrival times are considered, resulting in more variance between different scenarios.
- Vehicles are added to, and removed from, the network as necessary, resulting in more realistic scenarios for simulations lasting more than a few seconds.
- Vehicles trigger certain events upon entering predefined regions, allowing us to simulate critical regions.
- Vehicles only use predefined entrance and exit points, allowing us to create realistic highway scenarios.

Unlike STRAW, *mugen* does not consider the number of lanes or the interaction between vehicles. While this may be seen as a shortcoming of *mugen,* it should be noted that since we mainly focus on highway scenarios, vehicle interaction is minimal.

mugen takes the following parameters as input:

- A road network consisting of entrance, exit and intermediate points (P), and the roads between these points (R)
- One or more critical regions (G)
- The duration of the simulation (t_{sim})
- The vehicle density (ρ_v, in vehicles per hour) and the standard deviation of the inter-arrival times (σ_a, in seconds)
- The mean vehicle speed (μ_s, in meters per second) and the standard deviation of the vehicle speed (σ_s, in meters per second)

Using these parameters, *mugen* produces a partial ns2 script, along with a plot of the number of vehicles in the network at any given time during the simulation. Note that, unlike most other mobility generators, *mugen* does not take as input the number of vehicles, since this is computed dynamically using the road network size, vehicle density and vehicle speed.

The way *mugen* works is explained in Algorithms A.1–A.5 in the appendix. The full source code for *mugen* is released under the GNU General Public License 3.0, can be found at *http://github.com/cbguder/mugen*.

4.1 Vehicle Density and Speed

For vehicle density and speed, we tried to imitate İstanbul's traffic conditions. In [47], Wisitpongphan *et al.* analyze real-world data from Berkeley Highway Lab (BHL) and find a maximum vehicle density of about 3500 vehicles per hour during the morning rush hours. The average speed during this time seems to cluster around 40 mph (≈17.88 m/s). Their measurements between 10:00 AM and 12:00 PM show a vehicle density of 2619 vehicles per hour, and an average speed of 29.15 m/s.

The traffic conditions in İstanbul, however, are very different from Berkeley, CA. More than 12 million vehicles use the two bridges over the Bosporus in İstanbul [48] per month. This corresponds to an average of about 8333 vehicles per bridge per hour. Our own observations on the other hand, show a vehicle density of up to 15000 vehicles per hour during the afternoon hours on the Trans-European Motorway (TEM), with virtually no traffic congestion.

Drivers in İstanbul are used to crowded highways, and are more comfortable driving at higher speeds even when the inter-vehicle distance is very short. The mean vehicle speed therefore, was taken to be 20 m/s (72 km/h) in the slowest scenarios.

For our simulations, we considered three different cases, and generated 10 scenarios for each case, for a total of 30 distinct scenarios. The first scenario set represents night traffic with low vehicle density (2600 veh./h) and high speed (108 km/h). The second scenario set represents a typical situation near the bridges, with medium vehicle density (8333 veh./h) and lower speed (72 km/h). The third scenario set is based on our measurements of afternoon traffic on TEM, with high vehicle density (15000 veh./h) and high speed (108 km/h). Note that the low density (2600 veh./h) scenarios can also be interpreted as high density (15000 veh./h) scenarios where only 17% of the vehicles are equipped with OBUs.

The standard deviation of vehicle inter-arrival time was decreased with increasing vehicle density, while the standard deviation of vehicle speed was kept constant. Table 4.1 shows the parameters given to *mugen* for all three scenario sets.

Table 4.1: *mugen* parameters for all three scenario sets

ρ_v (veh./h)	μ_a (s)	σ_a (s)	μ_s (m/s)	σ_s (m/s)
2600	1.38	0.8	30	4
8333	0.43	0.4	20	4
15000	0.24	0.3	30	4

Figure 4.1 shows the number of vehicles in the network for all three scenario sets, averaged over 10 scenarios.

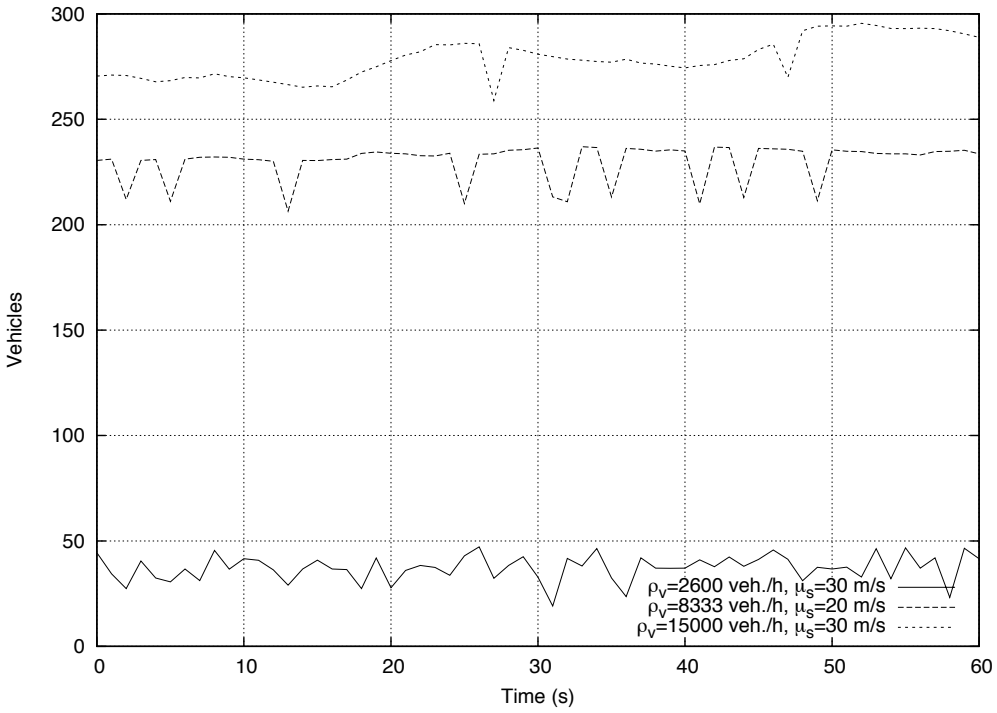

Figure 4.1: Average number of vehicles in the network

Figure 4.1 shows that the average number of vehicles in the network is around 275 in the most crowded scenario set. Assuming 3 lanes per side of the road, this number corresponds to an average of 45.83 vehicles per lane, and therefore an average inter-vehicle distance of about 65 meters for our 3 kilometer-long road network. Note that this inter-vehicle distance actually allows much higher speeds, and therefore less congestion and higher delivery rates.

Chapter 5

PRIVACY AND SECURITY

Driving safety applications, such as our Virtual Sign application, are of vital importance: they are the software counterpart of early warning and recovery technologies like anti-lock braking systems (ABS) or ESC. Any successful attack on these applications could potentially result in the injury or death of the driver and/or passengers in the vehicle. Therefore, we need to set strict security requirements for our application.

5.1 Requirements

We set four basic requirements for our application: authentication, data integrity, non-repudiation and conditional privacy.

Authentication is the act of confirming that received messages are generated by the legitimate users of the network (i.e. registered vehicles), and is arguably the most crucial security requirement for all driving safety applications. If an adversary can successfully masquerade as a legitimate user, he could inject false or misleading data into the network. This could cause traffic accidents, or cause the drivers to change their routes, which could potentially help the adversary or adversaries commit other felonies, such as an assassination.

Data integrity is the act of ensuring the received data is complete, valid and unaltered. This is to prevent malicious or accidental (e.g. transmission errors) modification of the messages.

Non-repudiation is assuring that users cannot later deny sending certain messages, and is required to enforce liability on the users for possibly malicious behavior in the contexts of vehicle or network traffic.

Our final requirement is conditional privacy. Most driving safety applications, including our Virtual Sign application, depend on vehicles to broadcast their location information periodically in messages called beacons. This information, albeit readily collectible without a vehicular network, can be used to track vehicles, threatening user privacy. While we want our users to remain anonymous, we also want to grant certain authorities (e.g. the police) the power to reveal the identity of certain users. This is referred to as conditional privacy.

Note that while non-repudiation and privacy can be seen as contradicting requirements, this is not the case in our system, as we do not require total privacy, but only conditional privacy. All communication among vehicles should be completely anonymous, therefore it is not possible to speak about non-repudiation for inter-vehicle communications. However, if the authorities decide to reveal the origin of a certain message, then the user should not be able to deny sending the message in question, and liability should be enforced.

5.2 Digital Signatures

In Section 2.3, we explained that a purely cryptographic protocol based on conventional (i.e. non-group) digital signatures fails to satisfy our requirement of conditional privacy. In a group signature scheme, on the other hand, group managers can reveal the identity of a message's signer.

Although group signatures satisfy all our requirements, public-key cryptography is notorious for being computationally expensive, and this is especially important in real-time applications running on embedded processors we assume will be used in OBUs.

Our measurements show that, given our assumptions of vehicle density and beacon period, vehicles receive an average of 50 key disclosure messages per second, and a maximum of 245 key disclosure messages per second. If every key disclosure message

was digitally signed, and a whole CPU was dedicated to verifying digital signatures, the dedicated CPU would have little more than 4 ms to verify a signature. However, our benchmarks reveal a 40.60 ms verification delay for the short group signature scheme proposed in [33]. The details of this benchmark is given in Section 7.1.1.

These numbers show that, in order to avoid dropping beacons, we need to either dramatically relax the beacon period, or decrease the number of group signature verifications by an order of at least 10. Opting for the latter, we suggest the use of one-way hash chains along with group signatures in beacons.

5.3 Security Model

The use of one-way hash chains for one-time password authentication was first proposed by Lamport in [49]. Later, Haller defined the S/KEY one-time password system [50,51]. In [36], Perrig *et al.* suggested the use of hash chains as part of the TESLA broadcast authentication protocol. In the TESLA broadcast authentication protocol, all messages include an authentication code generated using a keyed-Hash Message Authentication Code (HMAC) function, and the symmetric keys for the HMAC are taken from a one-way hash chain in reverse order. These keys are then revealed after a short waiting period in key disclosure messages. Public-key cryptography is only used for time synchronization messages.

In [35], Lin et al. propose using the TESLA authentication scheme in a vehicular context. In their proposed TSVC protocol, each vehicle periodically broadcasts the first element (tip) of the hash chain, so that new vehicles entering its transmission range can authenticate messages. Messages including the tip of the hash chain are signed using an anonymous key pair. We suggest a similar authentication protocol for beacons, using group signatures.

Let H(x) be a one-way hash function [52] with the following properties:

- H takes as input a message x of arbitrary length and produces a fixed-length digest H(x)

- Given x, it is computationally easy to compute $H(x)$, but it is computationally infeasible to find any x such that $H(x) = h$, given h. This is referred to as *preimage resistance*.
- Given x, it is computationally infeasible to find $x' \neq x$ such that $H(x') = H(x)$. This is referred to as *weak collision resistance*.
- It is computationally infeasible to find any pair x, x' such that $H(x) = H(x')$. This is referred to as *strong collision resistance*.

We can then define a hash chain of length N as the list of hash values $H(x)$, $H^2(x)$, $H^3(x)$,…, $H^N(x)$, where $H^n(x) = H(H^{n-1}(x)) = H(H(H(\ldots(H(x))\ldots)))$ (n times).

In our application, every vehicle generates a hash chain (Algorithm 5.1), and uses the elements of this hash chain in reverse order (i.e. from $H^N(x)$ to $H(x)$) as the keys of a HMAC function in the beacons it broadcasts (Algorithm 5.2). The HMAC key for each beacon is revealed shortly after the beacon in a compact key disclosure message (Algorithm 5.3). The security of our protocol, and all hash chain-based authentication protocols in general, depend mainly on the preimage resistance property of hash functions.

Algorithm 5.1 INITIALIZE-HASH-CHAIN()
(1) $ID \leftarrow$ random number
(2) $x \leftarrow$ random byte string
(3) $h_k \leftarrow H(x)$
(4) $i \leftarrow k - 1$
(5) **while** $i > 0$ **do**
(6) $\quad h_i \leftarrow H(h_{i+1})$
(7) $\quad i \leftarrow i - 1$
(8) **end for**
(9) $last\_signed\_seq \leftarrow 1 - k$
(10) $next\_seq \leftarrow 1$

A random node ID (or pseudonym) and a sequence number is used to associate beacons with key disclosure messages, and key disclosure messages with previously received key disclosure messages. This ID is changed every time the hash chain is re-initialized. Old beacons and key disclosure messages are automatically purged by a background process to avoid treating a single vehicle as two vehicles due to an ID change.

Algorithm 5.2 SEND-BEACON()

(1) $key \leftarrow h_{next\_seq}$
(2) create new beacon *b*
(3) *b.time* ← system time
(4) *b.source* ← *ID*
(5) *b.position* ← current position from GPS
(6) *b.speed* ← current speed from speedometer or GPS
(7) *b.direction* ← current direction from GPS
(8) *b.messages* ← {(*m.type, m.region*) | $m \in M$}
(9) *payload* ← (*b.source* ‖ *b.time* ‖ *b.position* ‖ *b.speed* ‖ *b.direction* ‖ *b.messages*)
(10) $b.hmac \leftarrow \text{HMAC}_{key}(payload)$
(11) SEND(*b*)
(12) wait for key disclosure interval δ
(13) SEND-KEY-DISCLOSURE(*key, next_seq*)
(14) **if** *next_seq* = *k* **then**
(15) INITIALIZE-HASH-CHAIN()
(16) **else**
(17) *next_seq* ← *next_seq* + 1
(18) **end if**

Algorithm 5.3 SEND-KEY-DISCLOSURE(*key, seq*)

(1) create new key disclosure message *kd*
(2) *kd.source* ← *ID*
(3) *kd.seq_no* ← *seq*
(4) *kd.key* ← *key*
(5) **if** *seq* = *last_signed_seq* + *k* **then**
(6) *kd.signature* ← SIGN(*kd.source* ‖ *kd.seq_no* ‖ *kd.key*)
(7) *last_signed_seq* ← *seq*
(8) **end if**
(9) SEND(*kd*)

Authentication is provided by the combination of the one-way hash chain and group signatures. One in every *k* key disclosure messages is signed using a group signature, where *k* is a system parameter that can have a fixed value or can be determined dynamically[5]. In the case that *k* is determined dynamically, the decision to change the value of *k* for future messages must be announced in a signed message. Note that *k* also determines the length of the hash chain: the hash chain cannot be made shorter than *k*, and making it longer does not provide additional benefits.

When a vehicle receives a beacon (Algorithm 5.4), the beacon is stored, but temporarily ignored. To be able to verify the HMAC value in a beacon, the receiver must

[5] In our simulations, *k* is constant at 10.

have received at least one signed key disclosure message from the same sender, so all beacons and key disclosure messages are discarded until a signed key disclosure message is received.

Algorithm 5.4 RECEIVE-BEACON(b)
(1) $B_{b.source} \leftarrow b$

Algorithm 5.5 RECEIVE-KEY-DISCLOSURE(kd)
(1) **if** kd is signed **then**
(2) **if** $kd.signature$ is valid **then**
(3) $K_{kd.source} \leftarrow kd.key$
(4) **else**
(5) delete $K_{kd.source}$
(6) **end if**
(7) **else if** $K_{kd.source}$ exists **then**
(8) **if** H($kd.key$) = $K_{kd.source}$ **then**
(9) $K_{kd.source} \leftarrow kd.key$
(10) **else**
(11) delete $K_{kd.source}$
(12) **end if**
(13) **end if**
(14) **if** $K_{kd.source}$ exists **and** $B_{kd.source}$ exists **then**
(15) $b \leftarrow B_{kd.source}$
(16) $key \leftarrow K_{kd.source}$
(17) $payload \leftarrow (b.source \parallel b.time \parallel b.position \parallel b.speed \parallel b.direction \parallel b.messages)$
(18) **if** $\text{HMAC}_{key}(payload) = b.hmac$ **then**
(19) PROCESS-BEACON(b)
(20) **end if**
(21) **end if**
(22) delete $B_{kd.source}$

Upon reception of a signed key disclosure message (Algorithm 5.5), the group signature on the message is verified, and the disclosed key is used to verify the HMAC on the last beacon received from the same sender. If both verifications succeed, the newly disclosed key is stored. For subsequent beacon/key disclosure message pairs, the disclosed key is compared with the previously stored key using the hash algorithm defined by the protocol, and the stored key is updated. If one of these verifications (group signature, hash or HMAC), the beacon, the key disclosure message and the previously stored key (if any) are immediately discarded.

Using this protocol, we are able to decrease the computational load on OBUs by a factor of almost k, and the average delay between receiving a beacon and processing

this beacon by more than 40%. In a pure group signature-based protocol, there would be a group signature verification delay for each beacon. In our scheme, this delay is amortized by the following *k*-1 beacons. Hash and HMAC delays are negligible.

5.4 Location Privacy

Linkability is defined as being able to link two messages sent by the same OBU at different times. If messages can be linked, then eavesdroppers can track vehicles by linking all the messages sent by a single OBU. Assuring unlinkability is an important step in providing location privacy in VANET applications.

On the other hand, some VANET applications require short-term linkability. For example, in our proposed security model, we need to be able to link two consecutive key disclosure messages, and a beacon with its corresponding key disclosure message. While our proposed security model does not aim to provide complete unlinkability, it does provide complete anonymity, and we take certain measures to avoid disclosing more information than necessary.

The group signature scheme we employ creates unlinkable signatures. As explained in Section 3.5.3, Virtual Sign messages do not contain any identity information, so no two Virtual Sign messages can be linked.

For beacons and key disclosure messages on the other hand, we use a random ID (or a pseudonym) to provide short-term linkability. This ID is changed every time the hash chain is reinitialized (every *k* beacons), and therefore only messages using the same hash chain can be linked. Note that these messages could still be linked by using the elements of the hash chain, even if we did not include a vehicle ID.

Two beacons, one sent just before an ID change and one sent right after, can still be linked, especially when the vehicle density is low. This is due to the deterministic nature of beacons: when and where the next beacon will be broadcast can be determined with very high accuracy. Our security model does not employ a mechanism to prevent this, but existing mechanisms, such as silent periods [53] or mix-zones [54] can be used in conjunction with our model.

Chapter 6

THE ADAPTIVE-P ALGORITHM

Vehicles running the Virtual Sign application broadcast Virtual Sign messages on two occasions: upon entering a critical region, and upon receiving a beacon.

When a vehicle enters a critical region, its onboard sensors and/or driving assistance systems such as ABS and ESC detect one or more events, and the vehicle broadcasts Virtual Sign messages containing information about these events. On the other hand, since Virtual Sign messages are meant to be area-persistent, vehicles might forward relevant Virtual Sign messages to other vehicles from time to time. When and how these Virtual Sign messages are broadcast is determined by the routing algorithm.

We propose a probabilistic routing algorithm called the "Adaptive-*p*" algorithm that takes into account the road and network conditions to decide whether or not to broadcast Virtual Sign messages. The Adaptive-*p* algorithm works thusly:

When a vehicle receives a Virtual Sign message, it first verifies the group signature on the message. If the signature is valid, the message is processed, and its contents, along with the time the message is received, are stored. If an equivalent message (i.e. a message having the same event type and region) has been stored before, only the time is updated. This is shown in Algorithm 6.1.

Algorithm 6.1 RECEIVE-VIRTUAL-SIGN(m)
(1) **if** *m.signature* is valid **then**
(2) **if** $\exists\, m' \in M \mid m \sim m'$ **then**
(3) $m'.time \leftarrow m.time$
(4) **else**
(5) $M \leftarrow M \cup \{m\}$
(6) **end if**
(7) **end if**

When a vehicle detects an event, it creates a new Virtual Sign message, and checks to see if it has previously stored an equivalent message. If no equivalent message has been stored before, the message is broadcast. If, however, an equivalent message has been stored before, the message is broadcast with a probability p_b proportional to the amount of time since the last such message is stored. So if an event is detected at t_e and an equivalent message has been stored at t_m, we have

$$p_b = \frac{t_e - t_m}{60}$$

Algorithm 6.2 DETECT-EVENT(e)
(1) create new Virtual Sign message m
(2) $m.type \leftarrow e.type$
(3) $m.region.center \leftarrow$ current position from GPS
(4) $m.region.radius \leftarrow$ 500m
(5) **if** $\exists\, m' \in M \mid m \sim m'$ **then**
(6) $\quad p_b \leftarrow (e.time - m'.time) / 60.0$
(7) $\quad$ **if** RANDOM() $< p_b$ **then**
(8) $\quad\quad$ SEND(m)
(9) $\quad$ **end if**
(10) **else**
(11) $\quad$ SEND(m)
(12) **end if**

When a vehicle receives a beacon, it compares the Virtual Sign message fingerprints contained in the beacon with the Virtual Sign messages it has stored before. If a Virtual Sign message has previously been stored by the receiving vehicle, and its fingerprint is not included in the beacon, the vehicle proceeds to check the message's region. If the message region contains either of the two vehicles (beacon sender and receiver), then the message is broadcast with a probability p_f inversely proportional to the number of vehicles around (i.e. inside its transmission range) the forwarder. So if the number of vehicles around the forwarder is n, we have

$$p_f = \frac{1}{n}$$

Algorithm 6.3 PROCESS-BEACON(b)
(1) $p_f \leftarrow 1.0 / n$
(2) *position* ← current position from GPS
(3) **for all** $m \in M$ **do**
(4) **if** $\nexists\, m' \in b.messages \mid m \sim m'$ **then**
(5) **if** *position* $\in m.region$ **or** $b.position \in m.region$ **then**
(6) **if** RANDOM() $< p_f$ **then**
(7) FORWARD(m)
(8) **end if**
(9) **end if**
(10) **end if**
(11) **end for**

6.1 Probability Analysis

We analyze the probabilities (p_b and p_f) produced by the Adaptive-p algorithm. Table 6.1 shows the average values and standard deviations (σ) for p_b and p_f, as an average of 10 scenarios.

Table 6.1: Overall average probabilities for all three scenario sets

ρ_v (veh./h)	μ_s (m/s)	Mean p_b	σ(p_b)	Mean p_f	σ(p_f)
2600	30	0.244	0.407	0.059	0.022
8333	20	0.242	0.374	0.063	0.082
15000	30	0.358	0.443	0.064	0.090

Figures 6.1–6.3 show how the average values for p_b and p_f change over time. The values are averaged over 10 scenarios. The increase in p_b with increasing vehicle density can be explained by the increased load on the network, and therefore some messages taking longer to reach certain vehicles.

Although p_f is inversely proportional to vehicle density, it increases over time for medium and high vehicle densities. This is a direct result of the increasing network congestion. As the network becomes more congested, the number of beacons that cannot be authenticated also increases, causing forwarders to increase p_f. The zigzag pattern is caused by our choice of k and beacon period. p_f increases until a signed key disclosure message arrives, and decreases after, creating a pattern with a period of 2 seconds.

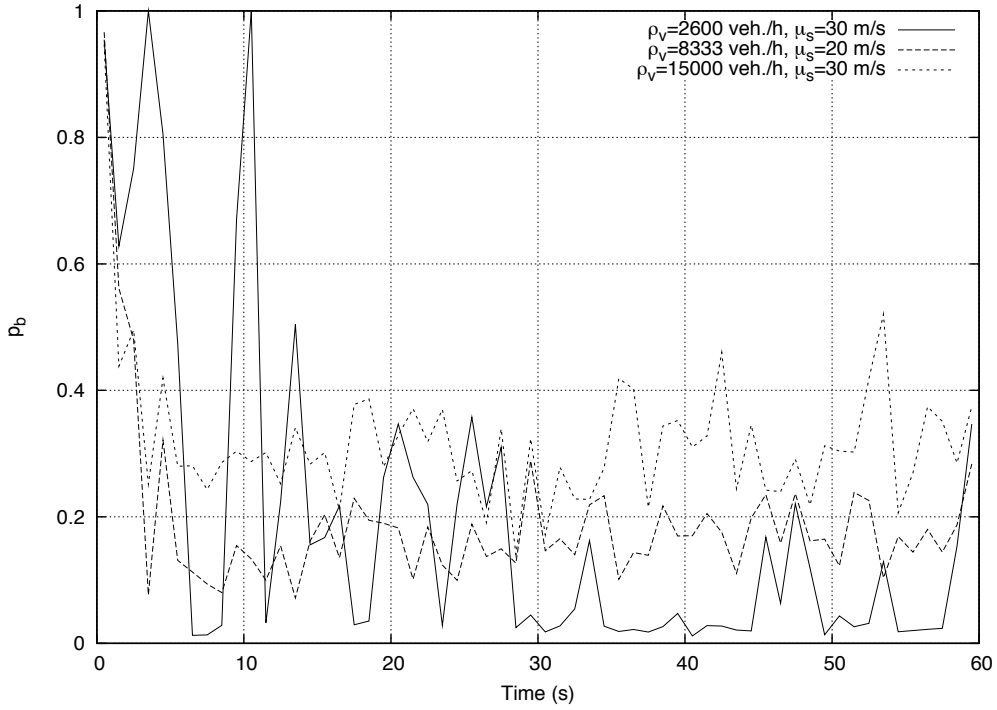

Figure 6.1: Average broadcast probability (p_b) for all three scenario sets

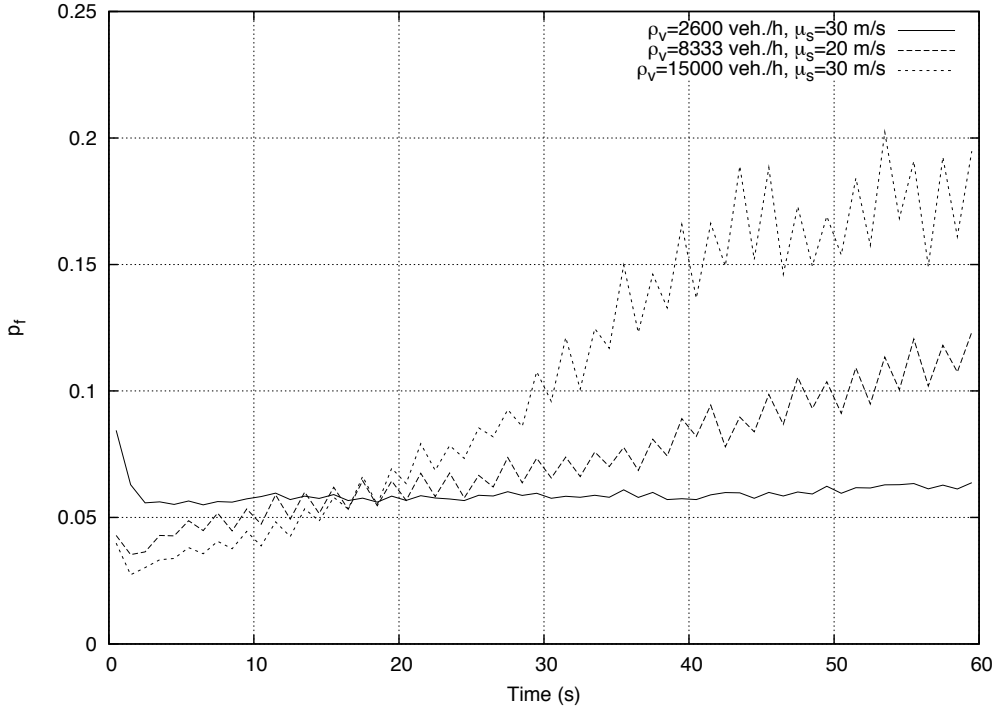

Figure 6.2: Average forwarding probability (p_f) for all three scenario sets

6.2 Flooding and Constant-p Algorithms

We compare our Adaptive-*p* algorithm with two other routing algorithms: flooding and Constant-*p*.

The Constant-*p* algorithm is the same as the Adaptive-*p* algorithm, except the broadcasting and forwarding probabilities are always constant, so we have $p_b = p_f = p$ at all times, for a predefined value of p.

As a base case for performance evaluation, we consider a flooding algorithm, where all beacons are ignored and message forwarding is only done upon reception of a new Virtual Sign message. Note that the definition of a "new" (i.e. not received before) Virtual Sign message is per the Adaptive-*p* algorithm: messages contents are compared rather than unique packet IDs, making our flooding implementation much more efficient than the common perception of a flooding algorithm.

Chapter 7

PERFORMANCE EVALUATION

7.1 Simulation Setup

We used ns-2.33 as our simulator, with the final patch and IEEE 802.11p parameters for ns2 from [55].

As explained in Section 4.1, we generated 10 scenarios of 60 seconds each for three parameter sets using *mugen*. The first scenario set has a vehicle density of ρ_v = 2600 veh./h and a mean vehicle speed of μ_s = 30 m/s, the second scenario set has a vehicle density of ρ_v = 8333 veh./h and a mean vehicle speed of μ_s = 20 m/s, and the third scenario set has a vehicle density of ρ_v = 15000 veh./h and a mean vehicle speed of μ_s = 30 m/s.

All simulations are run on a personal computer with the following configuration:

- Ubuntu 9.04 (64-bit)
- Intel Core 2 Duo E6600 Processor at 2.4 GHz
- 6 GB CL6 DDR2 SDRAM at 800 MHz
- GCC 4.3.3

All 30 scenarios are run with 5 different algorithms: flooding, Constant-*p* (for *p* = 0.2, 0.5 and 1.0) and Adaptive-*p*.

7.1.1 Latency Due To Cryptographic Operations

For group signatures, we assume a 50.91 ms signing delay, and a 40.60 ms verification delay. These values were obtained by benchmarking the short group signature scheme from [33] using the Pairing-Based Cryptography (PBC) Library [56] on the personal computer with the configuration given in Section 7.1, and taking the average of 10 benchmarks.

We used the parameter set a included in PBC, for a base field size of 512 bits and an embedding degree of 2 on the curve $y^2 = x^3 + x$. The group order is a 160-bit Solinas prime [57].

7.1.2 Antennas

We assume one omnidirectional antenna per vehicle, placed on the roof of the car, 1.5 meters from the ground. The transmission rate is set to 6 Mbps, resulting in an average transmission range of 600 meters using the Nakagami radio propagation model [58] implementation from [55] with the default parameters.

7.2 Performance Metrics

We consider four metrics for each configuration: awareness delay, message traffic, delivery rate and the number of events received.

7.2.1 Awareness Delay

We define awareness delay as the amount of time it takes a vehicle to reach full awareness after it enters the inspection region, where full awareness is defined as having received at least one message regarding each event.

This value is considered to be 0 for vehicles that have full awareness before entering the inspection region. This is only possible if vehicles enter the message regions be-

fore they enter the inspection region, and in our road network, only vehicles utilizing the entrance ramps can reach full awareness before they enter the inspection region.

The awareness delay for vehicles that leave the network in the first 10 seconds of the simulation, and the vehicles that enter the network in the last 10 seconds of the simulation are discarded to be able to capture the stable state of the network. Almost none of the vehicles in these two sets ever reach full awareness, as they simply do not spend enough time in the network to. Assuming the road conditions subject to the Virtual Sign messages sent predate the beginning of our simulations, most (if not all) of the vehicles in the first set should have already reached full awareness by the time the simulation started. The vehicles in the second set, on the other hand, would have reached full awareness if the simulations lasted longer.

7.2.2 Message Traffic

We define message traffic as the average number of Virtual Sign messages sent by each vehicle upon entering a critical region (and therefore detecting certain events), or forwarded in response to beacons. Note that the beacons are not considered, since they are mandated by the DSRC, and therefore the same amount of beacon traffic would exist without our Virtual Sign application. Also note that the average number of beacons sent by each vehicle stays almost constant between different scenarios, since beacons are sent at regular intervals, and never forwarded.

7.2.3 Delivery Rate

We define the delivery rate as the number of fully aware vehicles inside the inspection region divided by the total number of vehicles inside the inspection region at any given time.

Let E be the set of all events associated with the critical region, and $e(v,t)$ be the set of unique events received by vehicle v on or before time t. Also let $I(t)$ be the set of vehicles inside the inspection region at time t. The delivery rate at time t can then be defined as

$$\mathrm{DR}(t) = \frac{\left|\{v \in I(t) \mid E = e(v,t)\}\right|}{\left|I(t)\right|}$$

7.2.4 Number of Events Received

We measure the average number of events received by each vehicle inside the inspection region as a measure of how close these vehicles are to full awareness. The average number of events received at time t is defined as

$$\mathrm{NER}(t) = \frac{\sum_{v \in I(t)} e(v,t)}{\left|I(t)\right|}$$

7.3 Results

Tables 7.1–7.3 show the mean awareness delay, the standard deviation of awareness delay, and average message traffic for the five algorithms compared. Table 7.1 shows the results for the night scenarios (ρ_v = 2600 veh./h, μ_s = 30 m/s), Table 7.2 shows the results for the bridge scenarios (ρ_v = 8333 veh./h, μ_s = 20 m/s), and Table 7.3 shows the results for the TEM scenarios (ρ_v = 15000 veh./h, μ_s = 30 m/s).

Table 7.1: Simulation Results for ρ_v = 2600 veh./h, μ_s = 30 m/s

Algorithm	Mean Delay (s)	Std. Dev. of Delay (s)	Traffic (packets)
Flooding	0.62	1.29	21.30
Constant-*p* = 0.2	0.12	0.21	33.68
Constant-*p* = 0.5	0.11	0.19	65.97
Constant-*p* = 1.0	0.11	0.18	118.45
Adaptive-*p*	0.14	0.24	20.85

Table 7.2: Simulation Results for ρ_v = 8333 veh./h, μ_s = 20 m/s

Algorithm	Mean Delay (s)	Std. Dev. of Delay (s)	Traffic (packets)
Flooding	1.15	2.11	27.91
Constant-*p* = 0.2	0.35	0.57	34.98
Constant-*p* = 0.5	0.34	0.60	72.08
Constant-*p* = 1.0	0.30	0.57	139.36
Adaptive-*p*	0.26	0.52	50.97

Table 7.3: Simulation Results for ρ_v = 15000 veh./h, μ_s = 30 m/s

Algorithm	Mean Delay (s)	Std. Dev. of Delay (s)	Traffic (packets)
Flooding	0.99	1.69	26.98
Constant-*p* = 0.2	0.53	1.04	40.51
Constant-*p* = 0.5	0.60	1.24	87.04
Constant-*p* = 1.0	0.50	1.08	158.04
Adaptive-*p*	0.46	1.07	77.85

Tables 7.2 and 7.3 show that the Adaptive-*p* algorithm delivers the shortest awareness delays in the medium and high density scenario sets, and decreases the message traffic by more than 50% when compared to the algorithm delivering the second shortest awareness delays (Constant-*p* = 1.0) in both cases. In the low density scenario, the Adaptive-*p* algorithm slightly increases the awareness delay, but causes minimal message traffic.

Figures 7.1–7.3 show the average delivery rates for all the algorithms against time. The delivery rates depicted in the graphs are the averages of 10 scenarios.

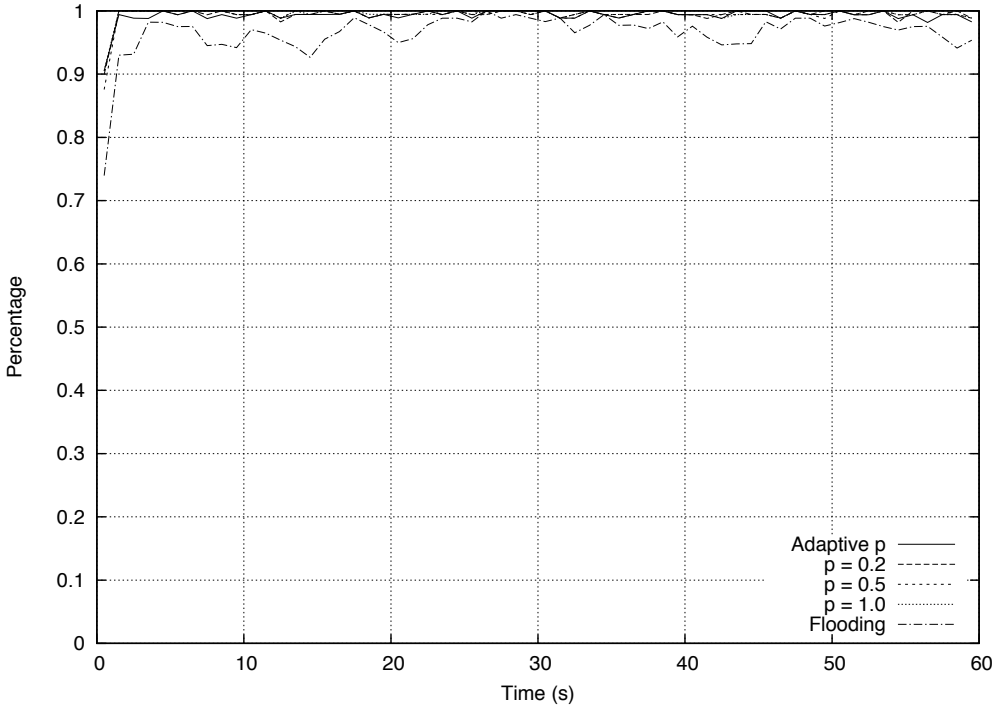

Figure 7.1: Delivery rates for ρ_v = 2600 veh./h, μ_s = 30 m/s

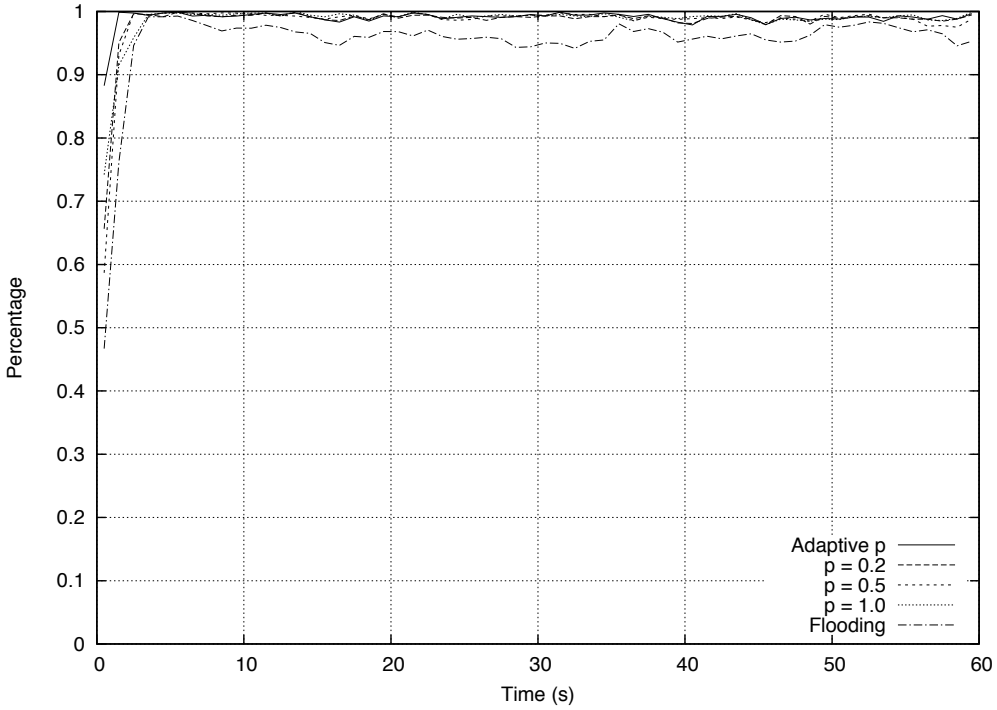

Figure 7.2: Delivery rates for ρ_v = 8333 veh./h, μ_s = 20 m/s

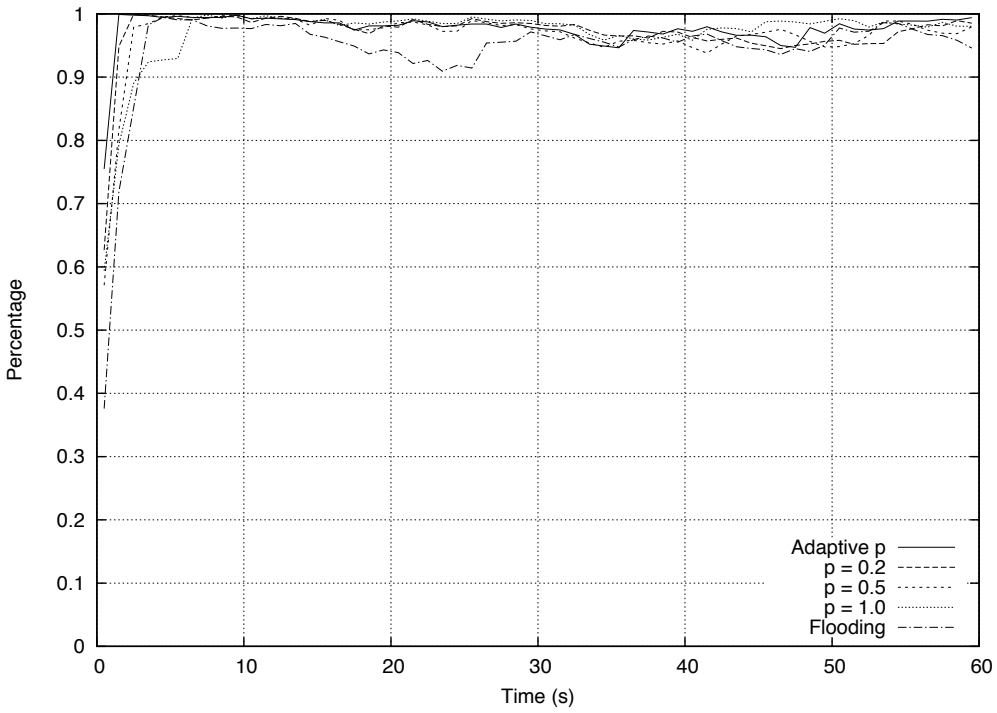

Figure 7.3: Delivery rates for ρ_v = 15000 veh./h, μ_s = 30 m/s

Figures 7.1–7.3 show that the Adaptive-*p* and Constant-*p* algorithms result in comparable delivery rates. Note that it takes longer for algorithms with higher *p* to reach a stable state, with the Adaptive-*p* algorithm stabilizing in less than 2 seconds in all scenario sets. The flooding algorithm gives the worst overall delivery in all scenario sets.

As explained in Section 7.2, Figures 7.1–7.3 show the percentage of fully aware vehicles inside the inspection region at any given time. However, there is a small percentage of vehicles inside the inspection region that are not fully aware (i.e. have not received information regarding all events). Our final performance metric reveals how close these vehicles are to full awareness.

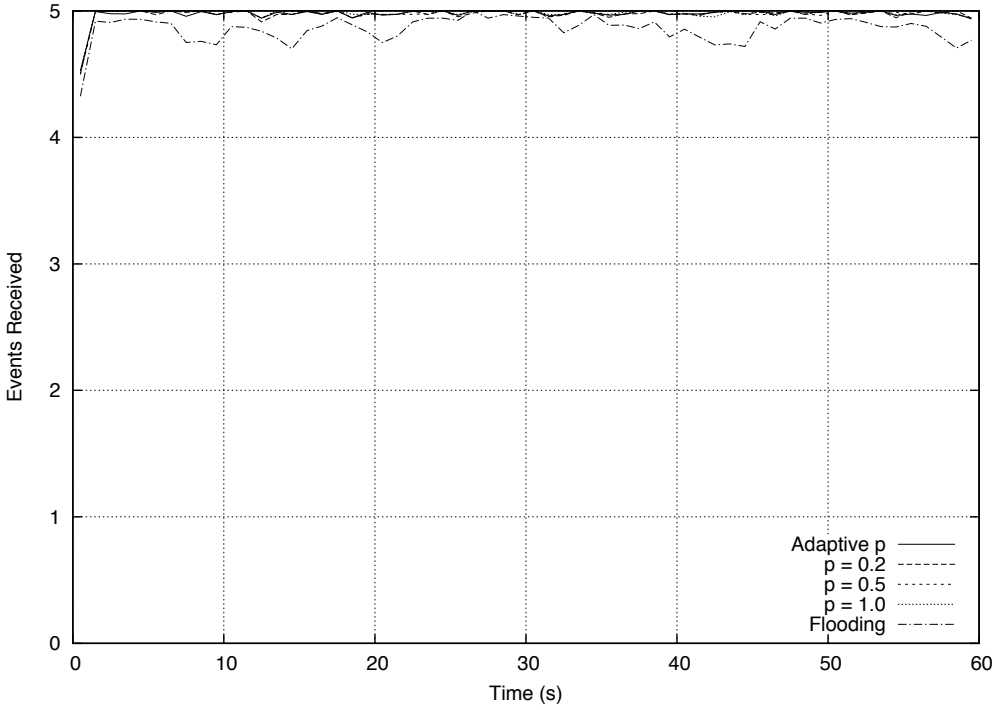

Figure 7.4: Average number of events received for ρ_v = 2600 veh./h, μ_s = 30 m/s

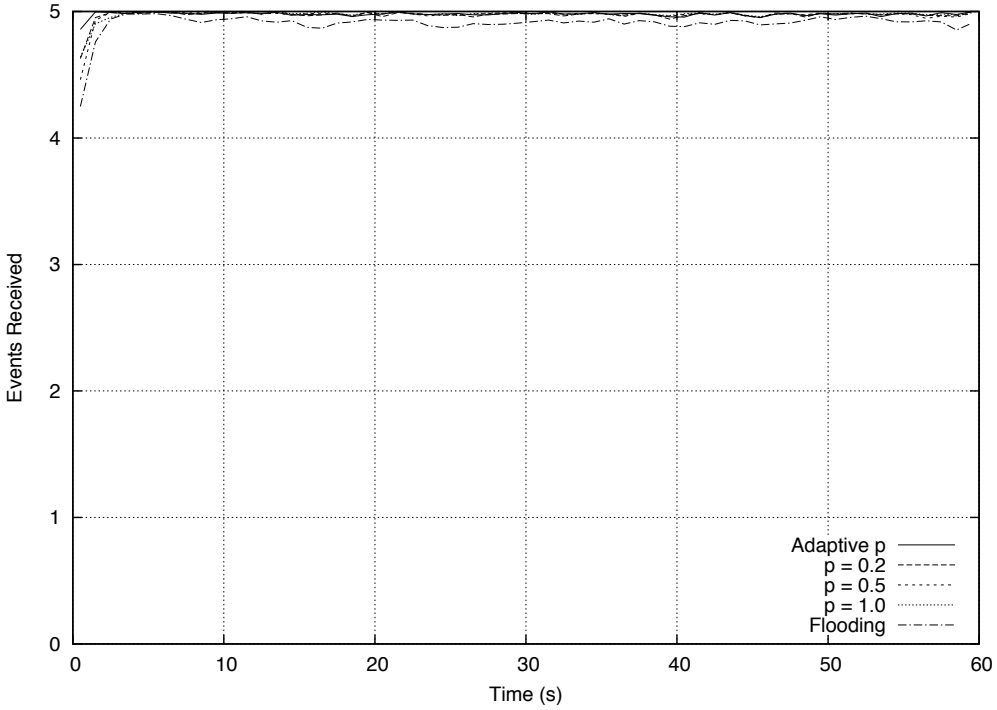

Figure 7.5: Average number of events received for ρ_v = 8333 veh./h, μ_s = 20 m/s

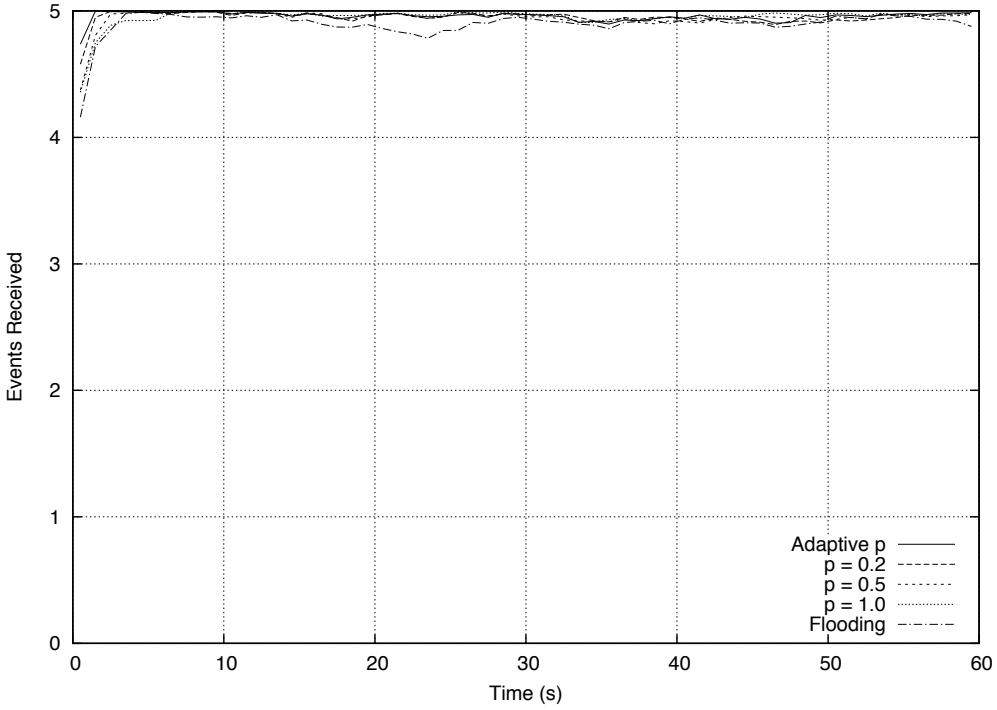

Figure 7.6: Average number of events received for ρ_v = 15000 veh./h, μ_s = 30 m/s

Figures 7.4–7.6 show very similar results with figures 7.1–7.3. Again, the probabilistic algorithms show comparable performance, with the flooding algorithm delivering the lowest results.

7.3.1 Manhattan Scenarios

Although *mugen* is not yet fully optimized for non-highway traffic, we wanted to test the Adaptive-*p* algorithm in a Manhattan scenario, and created a road network (Figure 7.7) that represents a 4-by-8-block Manhattan neighborhood. Each block was taken to be 280 by 80 meters, a common block size in Manhattan, New York. The roads between the blocks are one-way streets, with each street running in the opposite direction as its parallels. The critical region was selected to cover the middle intersection, while the inspection covers up to 500 meters of the two roads approaching the critical region.

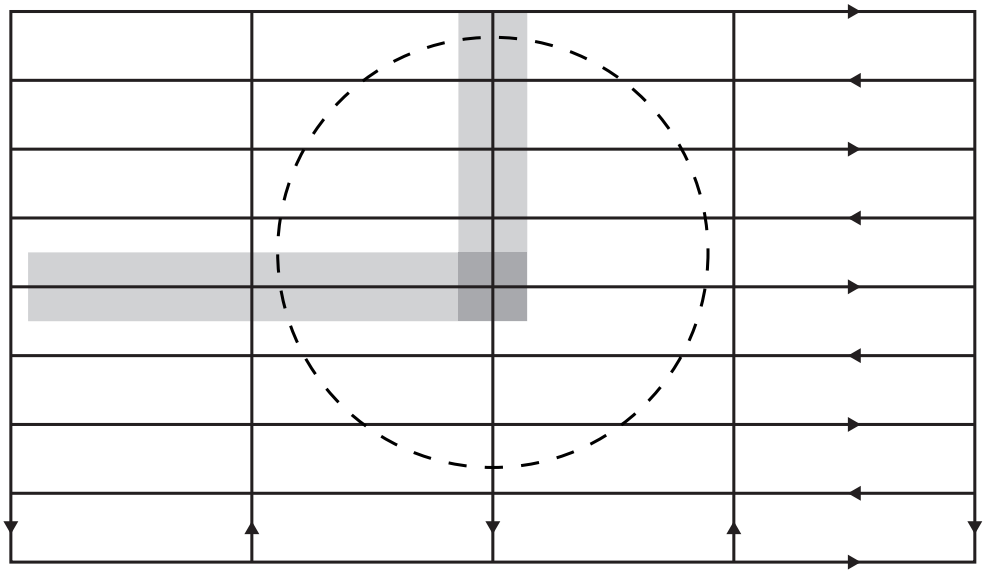

Figure 7.7: Manhattan road network

Table 7.4: *mugen* parameters for the Manhattan scenario set

ρ_v (veh./h)	μ_a (s)	σ_a (s)	μ_s (m/s)	σ_s (m/s)
2600	1.38	0.8	10	4

Table 7.5: Simulation Results for the Manhattan scenario set

Algorithm	Mean Delay (s)	Std. Dev. of Delay (s)	Traffic (packets)
Flooding	0.96	3.48	19.33
Constant-*p* = 0.2	0.46	1.10	12.89
Constant-*p* = 0.5	0.54	1.33	25.46
Constant-*p* = 1.0	0.25	0.55	44.00
Adaptive-*p*	0.29	0.71	19.13

Table 7.5 shows that the non-probabilistic algorithm delivers shorter awareness delays than the Adaptive-*p* algorithm in the Manhattan scenarios, albeit at the cost of a 130% increase in message traffic. Unlike the highway scenarios, Virtual Sign traffic is low and the beacon traffic is high in this scenario set. The total (as opposed to per vehicle) Virtual Sign traffic is 25% of the high-density highway set, and only twice that of the low-density highway set. This characteristic prevents the non-probabilistic algorithm from causing significant network congestion, and therefore allows it to perform better than it does in the highway scenarios.

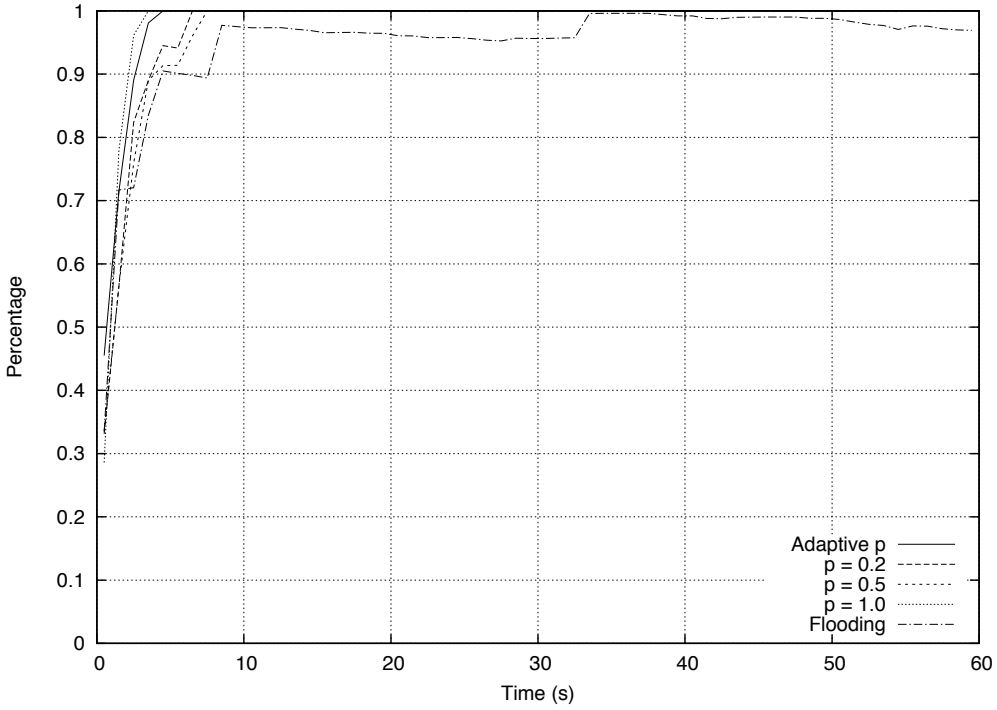

Figure 7.8: Delivery rates for the Manhattan scenario set

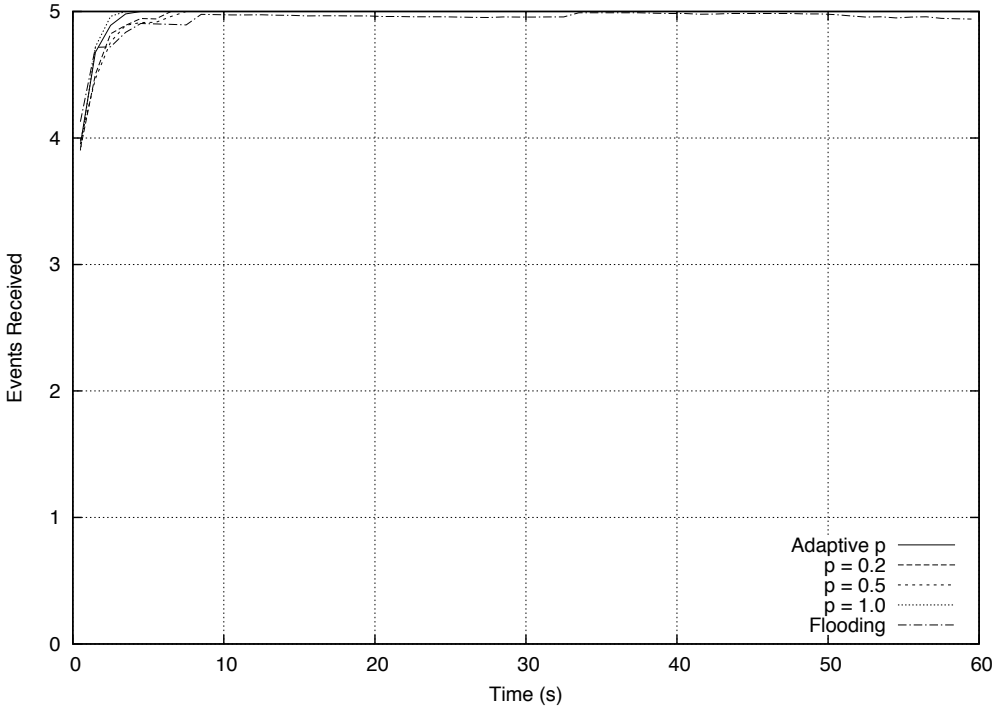

Figure 7.9: Average number of events received for the Manhattan scenario set

7.3.2 Analysis of System Parameter *k*

In sections 5.2 and 5.3, we defined *k* as a crucial system parameter, and explained why we chose the value 10 for *k* in our simulations. Although the cryptographic benchmarks we have conducted do not allow for lower values of *k,* higher values can still be used to further decrease the computational load on OBUs, possibly at the cost of higher awareness delays.

To confirm this, we ran all our simulations for two additional values of *k:* 15 and 20. The simulation results show no significant change in delivery rates, but awareness delays and message traffic were affected. For the sake of brevity, we omit the delivery graphs, and present the numeric results in Tables 7.6-7.9.

Table 7.6: Analysis of *k* for ρ_v = 2600 veh./h, μ_s = 30 m/s

	Mean Delay (s)			**Traffic (packets)**		
Algorithm	***k* = 10**	***k* = 15**	***k* = 20**	***k* = 10**	***k* = 15**	***k* = 20**
Flooding	0.62	0.56	0.56	21.30	21.98	21.87
Constant-*p* = 0.2	0.12	0.12	0.12	33.68	33.89	33.69
Constant-*p* = 0.5	0.11	0.10	0.11	65.97	60.68	65.73
Constant-*p* = 1.0	0.11	0.11	0.11	118.45	105.62	109.71
Adaptive-*p*	0.14	0.13	0.14	20.85	21.96	22.24

Table 7.7: Analysis of *k* for ρ_v = 8333 veh./h, μ_s = 20 m/s

	Mean Delay (s)			**Traffic (packets)**		
Algorithm	***k* = 10**	***k* = 15**	***k* = 20**	***k* = 10**	***k* = 15**	***k* = 20**
Flooding	1.15	1.03	0.98	27.91	27.47	28.44
Constant-*p* = 0.2	0.35	0.41	0.48	34.98	30.34	28.67
Constant-*p* = 0.5	0.34	0.49	0.38	72.08	62.80	58.21
Constant-*p* = 1.0	0.30	0.43	0.49	139.36	128.31	107.49
Adaptive-*p*	0.26	0.31	0.39	50.97	44.51	43.49

Table 7.8: Analysis of k for ρ_v = 15000 veh./h, μ_s = 30 m/s

	Mean Delay (s)			Traffic (packets)		
Algorithm	***k* = 10**	***k* = 15**	***k* = 20**	***k* = 10**	***k* = 15**	***k* = 20**
Flooding	0.99	1.00	0.87	26.98	28.16	27.42
Constant-*p* = 0.2	0.53	0.56	0.67	40.51	33.02	29.57
Constant-*p* = 0.5	0.60	0.62	0.65	87.04	73.89	62.23
Constant-*p* = 1.0	0.50	0.58	0.83	158.04	131.52	114.13
Adaptive-*p*	0.46	0.55	0.56	77.85	57.60	51.11

Table 7.9: Analysis of k for the Manhattan scenario set

	Mean Delay (s)			Traffic (packets)		
Algorithm	***k* = 10**	***k* = 15**	***k* = 20**	***k* = 10**	***k* = 15**	***k* = 20**
Flooding	0.96	0.44	0.51	19.33	19.17	19.60
Constant-*p* = 0.2	0.46	0.25	0.46	12.89	12.00	11.03
Constant-*p* = 0.5	0.54	0.35	0.24	25.46	22.87	20.80
Constant-*p* = 1.0	0.25	0.30	0.40	44.00	40.65	37.57
Adaptive-*p*	0.29	0.23	0.39	19.13	16.40	17.14

Tables 7.6-7.9 show that the awareness delay increases while the message traffic decreases with increasing k for the Adaptive-p and Constant-p algorithms in the medium (8333 veh./h) and high (15000 veh./h) density highway scenarios. This is expected, since as k increases the number of beacons that cannot be authenticated, and therefore discarded, also increase. This causes less Virtual Sign messages to be forwarded, decreasing message traffic and increasing the awareness delay.

In the low density (2600 veh./h) highway scenario, increasing values of k do not cause any significant change in awareness delays or message traffic. Also note that regardless of the values of k and p, the Adaptive-p and Constant-p algorithms deliver very similar awareness delays in this scenario set, although the message traffic increases with increasing p. This is because the total network traffic in this scenario set is low enough

to avoid most collisions and packet drops due to overfilled network interface queues, but the vehicle density is high enough not to create a disconnected network.

The flooding algorithm shows a trend of decreasing awareness delays with increasing *k*. Since beacons are ignored in the flooding algorithm, this can only be caused by the overall decrease in total network traffic, and therefore less network congestion.

In Section 5.2, we explained why *k* cannot be smaller than 10 for our parameters and assumptions, and this minimal value proved to be optimal for the Adaptive-*p* and Constant-*p* algorithms in the medium and high density highway scenarios, and near-optimal in the low density highway scenarios. The results for the Manhattan scenarios on the other hand, are irregular, with $k = 15$ delivering the best results overall. In Section 5.3, we suggested that *k* can be made adaptive, and having the highest beacon traffic in all scenario sets, the Manhattan scenarios are a great example of where *k* might be increased voluntarily.

Chapter 8

SUMMARY AND CONCLUSIONS

In this thesis, we proposed an enhanced random waypoint mobility model, a security model based on group signatures and one-way hash chains, and a probabilistic method for the dissemination of area-persistent safety messages in vehicular ad hoc networks.

Simulation results show that the Adaptive-*p* algorithm ensures the prompt delivery of all Virtual Sign messages to virtually all vehicles approaching the critical region, giving the drivers more than 485 meters of decision distance in all scenarios considered. Compared with the non-probabilistic (Constant-$p = 1.0$) algorithm, the Adaptive-*p* algorithm decreased the message traffic by more than 50% in all scenarios, and shortened the awareness delay by up to 13%.

In Chapter 4, we presented an enhanced RWP model that creates highway and manhattan scenarios, based on realistic assumptions. We noted that while *mugen* does not currently consider interactions between vehicles, the number of lanes or traffic congestion, these were expected to be minimal in the cases inspected in this thesis. Work is currently underway to add support for simulated driver reactions to received Virtual Sign messages, along with all the aforementioned features to *mugen.* This would allow us to create even more realistic scenarios.

For our security model, we used group signatures and an authentication protocol based on the TESLA broadcast authentication protocol. We defined a system parameter k for our beacon authentication protocol in Section 5.3. The value of k was constant at 10 in our simulations, and while any value lower than 10 is unrealistic when we consider the computational load on the OBUs, we inspected the consequences of using higher values for k in Section 7.3.2. We concluded that while the lowest possible value

of *k* is optimal for highway scenarios, using higher values of *k* can decrease awareness delays and message traffic in Manhattan scenarios. We also noted that our proposed protocol also allows *k* to be changed dynamically, and we believe this makes an interesting subject for possible future work.

Similarly, we assumed 5 events per critical region in order to test the performance of the Adaptive-*p* algorithm under increased network load. Considering the number of driving safety systems and sensors on today's vehicles, we believe this number is an upper bound on the number of events that can be associated with a single critical region, but less events per region and multiple critical regions can be inspected.

In our simulations, we assumed the use of one omnidirectional antenna per vehicle, which is partly due to the limitations of our simulator of choice, ns2. We believe the use of directional antennas (two bumper antennas in particular) can be an interesting subject for future work, and increase the performance of the Adaptive-*p* algorithm.

APPENDIX

Algorithm A.1 MUGEN(P, R, G, t_{sim}, ρ_v, σ_a, μ_s, σ_s)

(1) $S \leftarrow$ FIND-PATHS(P, R)
(2) $t_{trim} \leftarrow$ MAX($\{\ell(s) \mid s \in S\}$)$/\mu_s$
(3) $T \leftarrow t_{sim} + t_{trim}$
(4) $V \leftarrow$ GENERATE-MOVEMENT(S, T, ρ_v, σ_a, μ_s, σ_s)
(5) $V \leftarrow$ TRIM(V, t_{trim})
(6) ADD-COLLISIONS(V, G, t_{sim})

Algorithm A.2 FIND-PATHS(P, R)

(1) $S \leftarrow \varnothing$
(2) $E_n \leftarrow P \setminus \{p \in P \mid \exists\, r \in R, r.start = p\}$
(3) $E_x \leftarrow P \setminus \{p \in P \mid \exists\, r \in R, r.end = p\}$
(4) **for all** $r \in R \mid r.start \in E_n$ **do**
(5) $\quad s \leftarrow \{r.start, r.end\}$
(6) $\quad S \leftarrow S \cup \{s\}$
(7) **end for**
(8) **while** $\exists\, s \in S \mid s_n \notin E_x$ **do**
(9) $\quad C \leftarrow \{r \in R \mid r.start = s_n\}$
(10) $\quad$ **for all** $c \in C$ **do**
(11) $\quad\quad S \leftarrow S \cup \{\{s_1, s_2, \ldots, s_{n-1}, c.end\}\}$
(12) $\quad$ **end for**
(13) $\quad S \leftarrow S \setminus s$
(14) **end while**
(15) **return** S

Algorithm A.3 GENERATE-MOVEMENT(S, T, ρ_v, σ_a, μ_s, σ_s)

(1) $\mu_a \leftarrow 3600/\rho_v$
(2) $t \leftarrow 0$
(3) $V \leftarrow \varnothing$
(4) **while** $t < T$ **do**
(5) $\quad$ create vehicle v
(6) $\quad v.speed \leftarrow$ GAUSS(μ_s, σ_s)
(7) $\quad v.arrival \leftarrow t +$ GAUSS(μ_a, σ_a)
(8) $\quad v.path \leftarrow$ pick $s \in S$ uniformly at random
(9) $\quad V \leftarrow V \cup \{v\}$
(10) $\quad t \leftarrow v.arrival$
(11) **end while**
(12) **return** V

Algorithm A.4 TRIM(V, t_{trim})

(1) $V' \leftarrow \varnothing$
(2) **for all** $v \in V$ **do**
(3) $t \leftarrow v.arrival - t_{trim}$
(4) $T \leftarrow \{t\}$
(5) **for all** $p_i, p_{i+1} \in v.path$ **do**
(6) $t \leftarrow t + \delta(p_i, p_{i+1}) / v.speed$
(7) $T \leftarrow T \cup \{t\}$
(8) **end for**
(9) **if** $T_1 < t_{sim}$ **and** $T_n \geq 0$ **then**
(10) **if** $T_1 < 0$ **then**
(11) $f \leftarrow$ smallest i such that $T_i \geq 0$
(12) $p_0 \leftarrow$ interpolate position at $t = 0$
(13) $v.arrival \leftarrow 0$
(14) $v.path \leftarrow \{p_0, v.path_f, \ldots, v.path_n\}$
(15) $T \leftarrow \{0, T_i, \ldots, T_n\}$
(16) **end if**
(17) $v.times \leftarrow T$
(18) $V' \leftarrow V' \cup \{v\}$
(19) **end if**
(20) **end for**
(21) **return** V'

Algorithm A.5 ADD-COLLISIONS(V, G, t_{sim})

(1) **for all** $v \in V$ **do**
(2) **for all** $p_i, p_{i+1} \in v.path$ **do**
(3) **for all** $g \in G$ **do**
(4) $c \leftarrow$ COHEN-SUTHERLAND(g, $\{p_i, p_{i+1}\}$)
(5) **if** $c \neq$ NULL **then**
(6) $c.time \leftarrow c.time + v.arrival$
(7) $v.collisions \leftarrow v.collisions \cup \{c\}$
(8) **end if**
(9) **end for**
(10) **end for**
(11) **end for**

BIBLIOGRAPHY

[1] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile Computing,* Boston: Kluwer Academic Publishers, 1996, pp. 153–181.

[2] D. R. Choffnes and F. E. Bustamante, “An integrated mobility and traffic model for vehicular wireless networks,” in *VANET ’05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks.* New York, NY, USA: ACM, 2005, pp. 69–78.

[3] A. K. Saha and D. B. Johnson, “Modeling mobility for vehicular ad-hoc networks,” in *VANET ’04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks.* New York, NY, USA: ACM, 2004, pp. 91–92.

[4] D. Li, H. Huang, X. Li, M. Li, and F. Tang, “A distance-based directional broadcast protocol for urban vehicular ad hoc network,” in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, Sept. 2007, pp. 1520–1523.

[5] H. Füßler, M. Torrent-Moreno, M. Transier, R. Krüger, H. Hartenstein, and W. Effelsberg, “Studying vehicle movements on highways and their impact on ad-hoc connectivity,” *SIGMOBILE Mobile Computing and Communications Review,* vol. 10, no. 4, pp. 26–27, 2006.

[6] J. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. Saha, and D. Johnson, “Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture,” in *Mobile Computing Systems and Applications, 2003. Proceedings. Fifth IEEE Workshop on,* Oct. 2003, pp. 32–43.

[7] B. Raney, A. Voellmy, N. Cetin, M. Vrtic, and K. Nagel, “Towards a microscopic traffic simulation of all of Switzerland,” in *ICCS '02: Proceedings of the International Conference on Computational Science-Part I.* London, UK: Springer-Verlag, 2002, pp. 371–380.

[8] M.-T. Sun, W.-C. Feng, T.-H. Lai, K. Yamada, H. Okada, and K. Fujimura, “Gps-based message broadcasting for inter-vehicle communication,” in *Parallel Processing, 2000. Proceedings. 2000 International Conference on,* 2000, pp. 279–286.

[9] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, “Urban multi-hop broadcast protocol for inter-vehicle communication systems,” in *VANET ’04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks.* New York, NY, USA: ACM, 2004, pp. 76–85.

[10] M. Durresi, A. Durresi, and L. Barrolli, “Emergency broadcast protocol for inter-vehicle communications,” in *Parallel and Distributed Systems, 2005. Proceedings. 11th International Converence on,* vol. 2, July 2005, pp. 402–406.

[11] K. Tokuda, M. Akiyama, and H. Fujii, “Dolphin for inter-vehicle communications system,” in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE,* 2000, pp. 504–509.

[12] C. Maihöfer, “A survey of geocast routing protocols,” *IEEE Communications Survey & Tutorials,* vol. 6, no. 2, pp. 32–42, 2004.

[13] F. Li and Y. Wang, “Routing in vehicular ad hoc networks: A survey,” *Vehicular Technology Magazine, IEEE,* vol. 2, no.2, pp. 12–22, June 2007.

[14] L. Briesemeister, L. Schäfers, and G. Hommel, “Disseminating messages among highly mobile hosts based on inter-vehicle communication,” in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE.* 2000, pp. 522–527.

[15] A. Bachir and A. Benslimane, “A multicast protocol in ad hoc networks inter-vehicle geocast,” in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual,* vol. 4, Apr. 2003, pp. 2456–2460.

[16] C. Maihöfer and R. Eberhardt, “Geocast in vehicular environments: caching and transmission range control for improved efficiency,” in *Intelligent Vehicles Symposium, 2004 IEEE,* June 2004, pp. 951–956.

[17] C. Maihöfer, T. Leinmüller, and E. Schoch, “Abiding geocast: time-stable geocast for ad hoc networks,” in *VANET ’05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks.* New York, NY, USA: ACM, 2005, pp. 20–29.

[18] “Advanced Encryption Standard (AES),” NIST FIPS 197, Nov. 2001.

[19] R. Blom, “An optimal class of symmetric key generation systems,” in *EUROCRYPT ’84 – A Workshop on the Theory and Application of Cryptographic Techniques, Proceedings of,* Springer-Verlag, 1985, pp. 335–338.

[20] C. Laurendeau and M. Barbeu, “Secure anonymous broadcasting in vehicular networks,” in *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on.* Oct. 2007, pp. 661–668.

[21] W. Diffie and M. Hellman, “New directions in cryptography,” *Information Theory, IEEE Transactions on,* vol. 22, no. 6, pp. 644–654, Nov. 1976.

[22] W. Stallings, *Cryptography and Network Security,* 4th ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2006.

[23] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM,* vol. 21, no. 2, pp. 120–126, 1978.

[24] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *Information Theory, IEEE Transactions on,* vol. 31, no. 4, pp. 469–472, July 1985.

[25] D. R. Stinson, *Cryptography Theory and Practice,* 3rd ed. Boca Raton, FL: Chapman & Hall/CRC, 2006.

[26] V. S. Miller, “Use of elliptic curves in cryptography,” in *CRYPTO ’85: Advances in Cryptology.* London, UK: Springer-Verlag, 1986, pp 417–426.

[27] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of Computation,* vol. 48, no. 177, pp. 203–209. Jan. 1987.

[28] M. Raya and J.-P. Hubaux, “Securing vehicular ad hoc networks,” *Journal of Computer Security,* vol. 15, no. 1, pp. 39–68, 2007.

[29] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, “ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE,* Apr. 2008, pp. 1229–1237.

[30] A. Wasef and X. Shen, “PPGCV: Privacy preserving group communications protocol for vehicular ad hoc networks,” in *Communications, 2008. ICC '08. IEEE International Conference on,* May 2008, pp. 1458–1463.

[31] S. Zhu, S. Setia, S. Xu, and S. Jajodia, “GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks,” in *Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004, Proceedings of the First International Conference on,* 2004, pp. 42–51.

[32] D. Chaum and E. van Heyst, “Group signatures,” in *EUROCRYPT 1991, Proceedings of,* London, UK: Springer-Verlag, 1991, pp. 257–265.

[33] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” in *CRYPTO ’04, Proceedings of,* London, UK: Springer-Verlag, 2004, pp. 41–55.

[34] X. Lin, X. Sun, P.-H. Ho, and X. Shen, “GSIS: A secure and privacy-preserving protocol for vehicular communications,” *Vehicular Technology, IEEE Transactions on,* vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[35] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, “TSVC: Timed efficient and secure vehicular communications with privacy preserving,” *Wireless Communications, IEEE Transactions on,* vol. 7, no. 12, pp. 4987–4998, Dec. 2008.

[36] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA Cryptobytes,* vol. 5, no. 2, pp. 2–13. 2002.

[37] "Standard specification for telecommunications and information exchange between roadside and vehicle systems – 5 GHz band dedicated short range communications (DSRC) medium access (MAC) and physical layer (PHY) specifications," ASTM E2213-03, 2003.

[38] American Association of State Highway and Transportation Officials, *A Policy on Geometric Design of Highways and Streets 1984,* American Association of State Highway and Transportation Officials, 1984.

[39] "The Keyed-Hash Message Authentication Code," NIST FIPS 198, Mar. 2002.

[40] "IEEE draft standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 7: Wireless access in vehicular environments," *IEEE Unapproved Draft Std P802.11p/D7.0, May 2009.* 2009.

[41] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).* Dec. 2007.

[42] "IEEE Std. 1609.1 - 2006 IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager," *IEEE Std 1609.1-2006.* 2006.

[43] "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," *IEEE Std 1609.2-2006.* 2006.

[44] "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services," *IEEE Std 1609.3-2007.* 2007.

[45] "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation," *IEEE Std 1609.4-2006.* 2006.

[46] B. S. Gukhool and S. Cherkaoui, "IEEE 802.11p modeling in NS-2," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on,* Oct. 2008, pp. 622–626.

[47] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *Selected Areas in Communications, IEEE Journal on,* vol. 25, no. 8, pp. 1538–1556, Oct. 2007.

[48] Karayolları Genel Müdürlüğü, "Otoyol ve köprü gelirleri (net) (2009)." [Online]. Available: http://www.kgm.gov.tr/fr5.asp?tt=0307

[49] L. Lamport, “Password authentication with insecure communication,” *Communications of the ACM,* vol. 24, no. 11, pp. 770–772. 1981.

[50] N. Haller, “The S/KEY one-time password system,” in *Internet Society Symposium on Network and Distributed Systems, Proceedings of the,* 1994, pp. 151–157.

[51] N. Haller, *The S/KEY one-time password system,* IETF RFC 1760, Feb. 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1760.txt

[52] “Secure Hash Standard (SHS),” NIST FIPS 180-3, Oct. 2008.

[53] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, “AMOEBA: Robust Location Privacy Scheme for VANET,” *Selected Areas in Communications, IEEE Journal on,* vol. 25, no.8, pp. 1569–1589, Oct. 2007.

[54] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux, “Mix-Zones for Location Privacy in Vehicular Networks,” in *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS),* 2007.

[55] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, “Overhaul of IEEE 802.11 modeling and simulation in ns-2,” 2008. [Online]. Available: http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php

[56] The Pairing-Based Cryptography Library. [Online]. Available: http://crypto.stanford.edu/pbc/

[57] J. A. Solinas, “Generalized Mersenne numbers,” Centre for Applied Cryptographic Research (CACR), Waterloo, ON, Canada, Tech. Rep. 99-39, 1999.

[58] M. Nakagami, “The m-distribution, a general formula for intensity distribution of rapid fading,” in *Statistical Methods in Radio Wave Propagation: Proceedings of a Symposium held June 18-20, 1958,* W. C. Hoffman Ed. Oxford, UK: Permagon Press, 1960, pp. 3–36.