

**CERTIFICATE REVOCATION LIST DISTRIBUTION IN
VEHICULAR AD HOC NETWORKS**

A Dissertation
Presented to
The Academic Faculty

By

Michael E. Nowatkowski

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

Georgia Institute of Technology

May 2010

COPYRIGHT (C) 2010 BY MICHAEL E. NOWATKOWSKI

**CERTIFICATE REVOCATION LIST DISTRIBUTION IN
VEHICULAR AD HOC NETWORKS**

Approved by:

Dr. Henry L. Owen, III, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. John A. Copeland
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. George F. Riley
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Raheem A. Beyah
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Michael P. Hunter
School of Civil and Environmental
Engineering
Georgia Institute of Technology

Date Approved: March 30, 2010

To my family, who helped me through all the hard times.

ACKNOWLEDGEMENTS

I want to begin by expressing my deepest thanks to Dr. Henry Owen for his guidance and mentoring during this entire process. His ability to push without being pushy was invaluable to my completing all the requirements in the time I was allotted. I would also like to extend my thanks to the members of my committee, Dr. Copeland, Dr. Riley, Dr. Beyah, and Dr. Hunter. I appreciate your time and thoughtful comments.

I also need to thank many of my fellow students that assisted me over the past three years with my work and made lasting friendships: Chris Lee for spending many days exposing me to interesting research topics; Ying Xia, Kevin Fairbanks, Joe Benin and Selcuk Uluagac for all of the coding help; and Yusun Chang for all of the discussions about wireless protocols.

Many thanks to Revathi Balakrishnan for her help with GTNetS, and even more thanks to Dr. Riley, Josh Pelkey, Mathieu Lacage, and many, many others on the ns-3-users list for their help with ns-3.

Thank you to the Army and especially to the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, New York for giving me this incredible opportunity.

I would like to thank my wife Susie and my three children, Morgan, Thomas, and Caroline, for being so understanding during the past three years. They were incredibly supportive of me, allowing me to work while they kept everything going. I could not have accomplished this without their love and understanding.

Lastly, I must thank my parents for making me understand the value of education and perseverance. They have always been here for me, encouraging me to succeed in everything I have done.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	x
SUMMARY	xii
<u>CHAPTER</u>	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Contributions	2
1.3 Roadmap for Dissertation	3
2 VEHICULAR AD HOC NETWORKS	4
2.1 Introduction	4
2.2 Privacy	9
2.3 Public Key Infrastructure	10
3 CERTIFICATE REVOCATION LIST	14
3.1 CRL	14
3.2 Pseudonyms	17
3.3 Revocation Rate	22
3.4 CRL Size	25
3.5 Certificate Authority Regions	27
3.6 The WAVE CRL	29

4	VANET EFFECTS ON CRL DISTRIBUTION	33
4.1	Introduction	33
4.2	Encoding Methods	33
4.3	Available Channel Capacity	38
5	CRL DISTRIBUTION METHODS	40
5.1	Introduction	40
5.2	Current Methods of CRL Distribution (Related Work)	41
6	NEW METHODS OF DISTRIBUTION	49
6.1	Most Pieces Broadcast	49
6.2	Generation per Channel	53
6.3	CRL Distribution Flow Chart	54
7	SIMULATION METHOD	56
7.1	Introduction	56
7.2	ns-3	56
7.3	Assumptions	56
7.4	Common Simulation Settings	57
7.5	Mobility Models	59
7.6	Verification and Validation	65
7.7	Design of Experiments	66
8	SIMULATION RESULTS	70
8.1	Evaluation Criteria	70
8.2	Statistical Analysis	71
8.3	Factor Experiment Results	73

8.4 Trace Mobility Model Results	78
8.5 Comparison of Results	84
9 CONCLUSION	85
9.1 Recommendations	85
9.2 Future Work	86
APPENDIX A NS-2 MOBILITY TRACE CONVERTER	89
REFERENCES	93
VITA	101

LIST OF TABLES

	Page
Table 1. EDCA parameters in DSRC, number of slots.	7
Table 2. Size of elliptic curve elements, in bytes.	11
Table 3. Annual revocation rate triggers, 2005.	25
Table 4. Required pieces to download for a 2000-piece file with no coding methods used.	34
Table 5. Common simulation settings.	57
Table 6. Vehicle trace parameters.	62
Table 7. Experiment factors.	67
Table 8. Number of pieces required for CRL download based on various piece sizes.	68
Table 9. Most effective levels per factor.	73

LIST OF FIGURES

	Page
Figure 1. Elements of a VANET.	5
Figure 2. DSRC channels. [62]	6
Figure 3. WAVE sync interval. [16]	6
Figure 4. Number of pseudonyms valid over a one year period for an average of two hours driven per day, with year-long lifetime or week-long lifetime.	20
Figure 5. Number of valid pseudonyms in an OBU per day.	21
Figure 6. Number of hourly revocations over varying pseudonym lifetimes.	27
Figure 7. WAVECRL elements.	30
Figure 8. Size of hourly CRL over varying pseudonym lifetimes.	31
Figure 9. Example of network coding generations.	37
Figure 10. Number of packets transmitted and throughput as a function of packet length for a single SCH interval.	39
Figure 11. Download time for a 1 megabyte file with different packet lengths.	39
Figure 12. CRL distribution in a VANET.	40
Figure 13. V2I and V2V integration in VANET.	44
Figure 14. Example of most pieces broadcast.	52
Figure 15. Flowchart for the simulation models.	55
Figure 16. Bouncing box RSU and OBU start positions with 50 OBUs.	60
Figure 17. Position traces from ns-3 output for the Switzerland maps.	63
Figure 18: Comparison of simulation model to Code Torrent output.	66
Figure 19. Effect of piece selection on download time.	74
Figure 20. Effect of piece size on download time.	75
Figure 21. Effect of GPC channel selection method on download time.	76

Figure 22. Effect of number of GPC channels on download time and PDR.	77
Figure 23. Completion times for OBUs from Enge-Oberstrass with 520 OBUs.	78
Figure 24. Packet deliver ratio and normalized packet overhead for OBUs from Enge-Oberstrass with 520 OBUs.	79
Figure 25. Number of transmitted and received pieces from Enge-Oberstrass with 520 OBUs.	79
Figure 26. Completion times for OBUs from Enge-Oberstrass with 1705 OBUs.	80
Figure 27. Packet deliver ratio and normalized packet overhead for OBUs from Enge-Oberstrass with 1705 OBUs.	81
Figure 28. Number of transmitted and received pieces from Enge-Oberstrass with 1705 OBUs.	81
Figure 29. Completion times for OBUs from Hurgem-Jona with 1275 OBUs.	82
Figure 30. Packet deliver ratio and normalized packet overhead for OBUs from Hurgem-Jona with 1275 OBUs.	83
Figure 31. Number of transmitted and received pieces from Hurgem-Jona with 1275 OBUs.	83
Figure 32. Comparison of completed OBUs during vehicle traces.	84
Figure 33. Trace file header information.	89
Figure 34. Sample from trace file hw-ct-hurgem-jona-1day.high.0.adj.mov.	90
Figure 35. Comparison of vehicle locations from the original trace file, the current ns-2 trace reader, and the improved ns-2 trace reader.	91
Figure 36: Mobility trace from the output of ns-3 for a single node.	92

LIST OF SYMBOLS AND ACRONYMS

\bar{D}	Difference Between Sample Means
S^2	Sample Variance
\bar{Y}	Sample Mean
$t_{1-\alpha/2, v}$	Student-t distribution
AC	Access Category
ACID	Application Class Identifier
ACM	Application Context Mark
AIFS	Arbitration Inter-Frame Space
CA	Certificate Authority
CCH	Control Channel
CRL	Certificate Revocation List
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CWmax	Contention Window Maximum Value
CWmin	Contention Window Minimum Value
DSRC	Dedicated Short Range Communication
EDCA	Enhanced Distributed Channel Access
ECDSA	Elliptic Curve Digital Signature Algorithm
EPFL	<i>Ecole Polytechnique Federale de Lausanne</i>
GPC	Generation per Channel
IPV6	Internet Protocol version 6
MAC	Medium Access Control
MPB	Most Pieces Broadcast
NPO	Normalized Packet Overhead
OBU	On Board Unit

PDR	Packet Delivery Ratio
PKI	Public Key Infrastructure
RSU	Road Side Unit
SCH	Service Channel
TNB	Top N Broadcast
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad Hoc Network
WAVE	Wireless Access in Vehicular Environment
WBSS	WAVE Basic Service Set
WSM	WAVE Short Message
WSMP	WAVE Short Message Protocol

SUMMARY

The objective of this research is to investigate improved methods for distributing certificate revocation lists (CRLs) in vehicular ad hoc networks (VANETs). VANETs are a subset of mobile ad hoc networks composed of network-equipped vehicles and infrastructure points, which will allow vehicles to communicate with other vehicles and with roadside infrastructure points. While sharing some of the same limitations of mobile ad hoc networks, such as lack of infrastructure and limited communications range, VANETs have several dissimilarities that make them a much different research area. The main differences include the size of the network, the speed of the vehicles, and the network security concerns. Confidentiality, authenticity, integrity, and availability are some of the standard goals of network security. While confidentiality and authenticity at times seem in opposition to each other, VANET researchers have developed many methods for enhancing confidentiality while at the same time providing authenticity. The method agreed upon for confidentiality and authenticity by most researchers and the IEEE 1609 working group is a public key infrastructure (PKI) system. An important part of any PKI system is the revocation of certificates. The revocation process, as well as the distribution of revocation information, is an open research problem for VANETs. This research develops new methods of CRL distribution and compares them to existing methods proposed by other researchers. The new methods show improved performance in various vehicle traffic densities.

CHAPTER 1

INTRODUCTION

Network communications equipment will be installed in vehicles and roadside infrastructure points. The primary uses identified for Vehicular Ad Hoc Networks (VANETs) are safety-related messages, transportation efficiency, and entertainment content [1]. The Vehicle Safety Communications-Applications consortium identified some potentially life-saving warnings, including: emergency electronic brake light, pre-crash sensing, cooperative forward collision warning, left turn assistant, lane-change warning, traffic signal violation warning, curve speed warning, and stop sign movement assistant [2]. For security and safety reasons, messages must be authenticated to ensure that a legitimate member of the VANET sent the message. This is especially critical for safety-related messages. When a user sends out erroneous messages, whether intentionally or unintentionally, other members of the VANET should ignore those messages to protect their safety. Public key certificates are used for authentication to prevent attackers from causing harm. Certificate revocation lists (CRLs), which contain the identification numbers of certificates that should be ignored, are distributed to all members of the VANET. To protect the members of the VANET, these lists should be distributed as quickly and efficiently as possible without over-burdening the network. This research develops two methods for distributing CRLs more quickly and efficiently than other currently proposed methods.

1.1 Motivation

Improving driving safety is seen as the most important reason for enabling the capability of vehicle to vehicle communication. In 2008, there were 37,261 fatalities and over 2.3 million injuries caused by more than 5.8 million vehicle accidents [3]. For 2009,

33,963 fatalities occurred [4]. The estimated economic impact for vehicle crashes in 2000 is estimated at \$230 billion, of which \$21 billion was paid using tax money, or about \$200 per household [5]. After examination of the causes for vehicle accidents in 2004, [6] estimates that vehicle-to-vehicle safety messages could have prevented or reduced the seriousness of about 66%, close to 4 million, of the crashes that year.

Transportation efficiency applications for VANET include re-routing traffic to avoid congestion, enhanced route guidance and navigation, green light optimal speed advisory, lane merging assistant, and tolling. Information and entertainment application for VANET include mobile-Internet, point-of-interest notification, fuel consumption management, and vehicle diagnostics [1, 7].

The messages exchanged in the VANET must be accurate and reliable for these applications to improve the safety, efficiency, and convenience of driving. Protecting members of the VANET from misbehaving nodes — nodes which send out erroneous messages — is paramount. Identifying misbehaving nodes and revoking their ability to send messages will contribute to the security of the VANET.

1.2 Contributions

This work contributes several novel items to the VANET research community, as well as to the ad hoc network community. The primary contribution is the development, simulation, and analysis of two methods for delivering CRLs in a VANET environment. These two methods are Most Pieces Broadcast (MPB), and Generation Per Channel (GPC). These methods may be applicable for the distribution of large files in other forms of ad hoc networks as well. Additional contributions include examining the lifetime of pseudonyms and their effect on CRL size, developing VANET modules in ns-3 for other researchers in the community to continue to develop, and providing a synopsis of the current state of the art for CRL distribution in VANETs. After conducting an in-depth literature search, no other researchers are considering the multi-channel nature of

dedicated short range communication (DSRC) for the distribution of CRLs. This research uses the multiple channels of DSRC to enable methods that greatly improve CRL distribution. These methods are compared to other proposed methods in simulation using realistic vehicle mobility traces.

1.3 Roadmap for Dissertation

This dissertation is organized into nine chapters plus an appendix. Chapter 2 introduces vehicular ad hoc networks, specifically discussing the physical and medium access control (MAC) layers, and privacy and security issues. Chapter 3 discusses certificate revocation lists. The use of certificates is central to this discussion, so they are discussed in the context of VANETs. Chapter 4 looks at the relationship between VANETs and the CRL, examining the parameters involved with sending large files over a highly mobile network. Chapter 5 discusses the specifics of distributing the CRL in a VANET from a practical point of view and looks at previous research in this area. Chapter 6 introduces the methods developed for examination in this body of work. Chapter 7 details the simulation parameters and modeling techniques used to generate data for analysis followed by the analysis of the data in Chapter 8. The concluding remarks are in Chapter 9. Details about incorporating the mobility model into ns-3 are included as Appendix A.

CHAPTER 2

VEHICULAR AD HOC NETWORKS

Vehicular ad hoc networks are a subset of mobile ad hoc networks (MANETs). While sharing some of the same limitations, such as lack of infrastructure and limited communications range, they have several dissimilarities that make VANETs a much different research area; therefore, research done on MANETs is not completely applicable to VANETs. VANETs are hosted on vehicles and fixed infrastructure points, so power and space for radio, storage, and processing units is not an issue, and the number of vehicles and their speed makes scalability difficult [8]. Hartenstein and Laberteaux in [1] discuss many practical difficulties in VANET research, including lack of communication coordination, dynamic network topology due to mobility and radio propagation limitations, as well as balancing security and privacy. Confidentiality, in the forms of privacy and anonymity, becomes a very important issue. In fact, IEEE Standard 1609.2 [9] specifies that “anonymity for end-users” is a required security service.

2.1 Introduction

While consumer VANETs have not yet been deployed, the idea has been discussed by research groups, government agencies, and vehicle manufacturers for several years. In the United States, the FCC designated a 75 MHz band in the 5.9 GHz range for this purpose in 1999 [10]. Several other countries have also apportioned frequencies for VANETs. Currently, test beds in the United States and Europe [11-13] are fielding prototypes and testing some of the initial protocols. The physical layer specification is called by different names, including Dedicated Short Range Communication (DSRC) and Wireless Access in Vehicular Environments (WAVE).

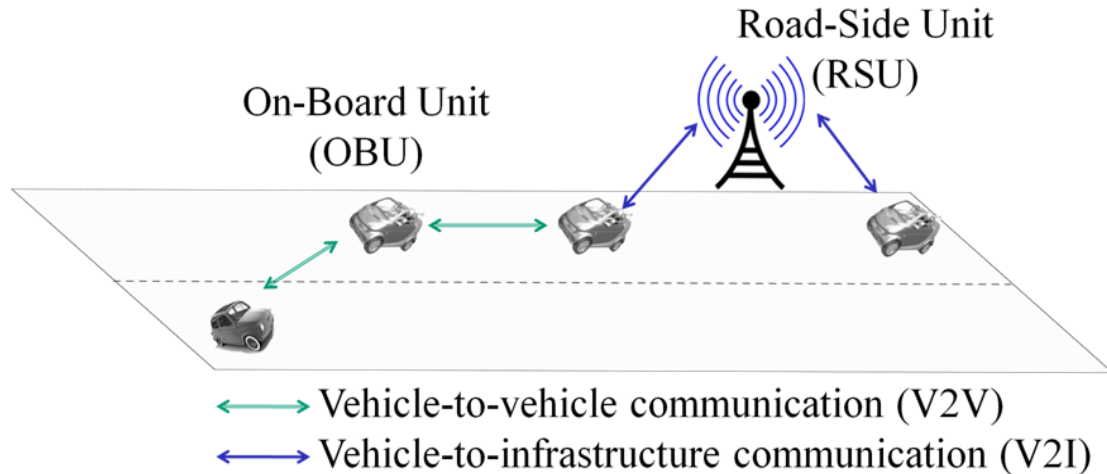


Figure 1. Elements of a VANET.

2.1.1 IEEE 1609 Standards

The IEEE P1609 working group has developed and issued a series of Trial-Use Standards for Wireless Access in Vehicular Environments (WAVE). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication are accomplished using Dedicated Short Range Communication (DSRC), which is covered in the IEEE Standard P802.11p and the IEEE Standard 1609 series. This standard has four main parts, covering the application layer [14], security services [9], multichannel operation [15], and network services [16]. IEEE 1609 uses IEEE P802.11p and DSRC as the WAVE protocols. Two devices are defined in the 1609 standard: a roadside unit (RSU) and an on-board unit (OBU) [14]. Figure 1 shows the elements of a VANET.

2.1.2 Physical Layer Standard

IEEE 802.11p [17] is a draft IEEE standard for vehicular communication as an amendment to the IEEE 802.11 standard [18]. The 802.11p standard is specifically for wireless access in vehicular environments and covers many aspects of the physical and MAC layer protocols for this case. Two different classes of channels are described for use in DSRC/WAVE. The first channel class is the control channel, referred to as CCH,

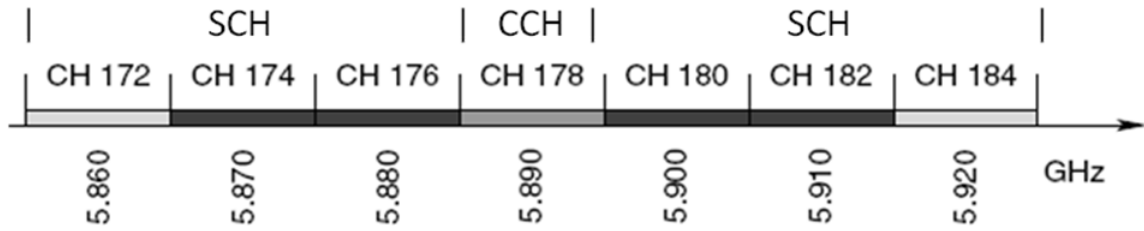


Figure 2. DSRC channels. [62]

which is a single channel "reserved for short, high-priority application and system control messages [15]." OBUs transmit beacons at regular intervals on the CCH, usually 10 beacons per second. The beacons broadcast information about the location, speed, and direction of the vehicles. This information is used for safety-related and network-related uses. The other channel class is the service channel, or SCH, which has six different 10 MHz channels that support a wider range of applications and data transfer. The channel layout for DSRC is shown in Figure 2.

Each node in the VANET must monitor the CCH during time periods designated as control channel intervals. Each CCH interval and SCH interval is 50 milliseconds in duration. The time period for an entire CCH Interval and SCH Interval is 100 milliseconds in duration and is called a Sync Interval (see Figure 3). Between CCH intervals, nodes may choose to participate on a SCH for applications such as file downloads, navigation updates, or other applications.

A guard interval is placed at the beginning of each CCH and SCH interval to assist in channel synchronization for all the nodes in the VANET. This guard interval allows for possible "variations in channel interval time and timing inaccuracies [16]." Annex D of IEEE 1609.4 specifies the parameters for the sync intervals and the guard

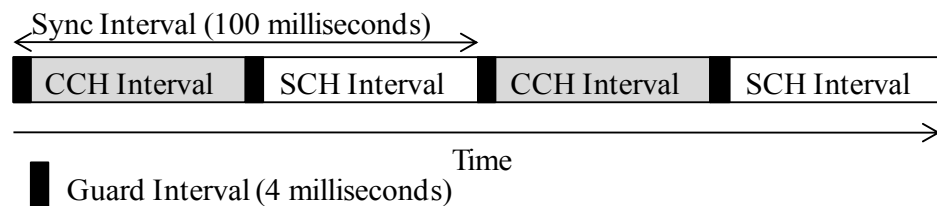


Figure 3. WAVE sync interval. [16]

interval. The guard interval is the "SyncTolerance" value plus the "MaxChSwitchTime" value. Both of these values are 2 milliseconds by default, for a total guard interval time of 4 milliseconds. This leaves 46 milliseconds for transmissions in the CCH and SCH intervals. A medium busy signal is set during the guard interval to prevent stations from transmitting during this time. The medium busy signal also prevents stations from transmitting immediately at the end of the guard interval, forcing the stations into a random contention window selection at the start of each interval.

2.1.3 Medium Access Control (MAC) Layer Standard

Packet collisions and poor radio reception are the primary causes for nodes not receiving data sent over a wireless medium. DSRC uses carrier sense multiple access/collision avoidance (CSMA/CA) to reduce the number of collisions and to allow fair access to the medium. Each node using CSMA/CA must first sense the medium to

Table 1. EDCA parameters in DSRC, number of slots.
Slot time is 16 microseconds, aCWmin = 15, aCWmax = 511.

Control Channel (CCH)						
ACI	AC	CWmin		CWmax	AIFS	
1	Background	aCWmin		aCWmax	9	
0	Best Effort	$\frac{(aCWmin+1)}{2}$	-1	aCWmin	6	
2	Video	$\frac{(aCWmin+1)}{4}$	-1	$\frac{(aCWmin+1)}{2}$	-1	3
3	Voice	$\frac{(aCWmin+1)}{4}$	-1	$\frac{(aCWmin+1)}{2}$	-1	2
Service Channel (SCH)						
ACI	AC	CWmin		CWmax	AIFS	
1	Background	aCWmin		aCWmax	7	
0	Best Effort	aCWmin		aCWmax	3	
2	Video	$\frac{(aCWmin+1)}{2}$	-1	aCWmin	2	
3	Voice	$\frac{(aCWmin+1)}{4}$	-1	$\frac{(aCWmin+1)}{2}$	-1	2

determine if the medium is idle or busy. When the medium is idle, nodes wait for a fixed arbitration inter-frame space (AIFS) time plus a random time between zero and the minimum contention window value (CW_{min}) before sending data. Transmission prioritization in DSRC is scheduled using Enhanced Distributed Channel Access (EDCA) based on the IEEE Standard 802.11e, described in [18, 19]. EDCA specifies four different access categories (ACs) for different priorities of data. The CCH and SCH have slightly different parameters for the different ACs. Table 1 shows the values of the contention window timers and the AIFS timers for the CCH and SCH over the different ACs. The slot time, as described in [16], is 16 microseconds.

Broadcast mode sends data to the network broadcast address instead of to a specific destination address. The data is not sent to a specific destination but rather to every node that is within radio range of the sender. The ready-to-send (RTS) and clear-to-send (CTS) handshake used for point-to-point data transmission is not used in broadcast mode. The RTS/CTS handshake is an attempt to reduce the number of collisions resulting from the hidden terminal problem. Acknowledgements (ACKs) are also not sent by receiving nodes when broadcast mode is used. Therefore, since there is no RTS, CTS, or ACK when using broadcast mode, channel overhead is reduced since fewer transmissions are needed to complete a data transfer. This allows for a potentially greater throughput of data. However, there is no confirmation that the data was successfully received by any destination, and there is a higher risk of collisions due to hidden terminals [20].

2.1.4 Network Layer Standards

WAVE supports two different network layer protocols, IPv6 and the WAVE short message protocol (WSMP). IPv6 traffic is not allowed on the CCH, but WSMP traffic is allowed on both the CCH and the SCHs. The WSMP does not use a MAC address or IP address to identify the source or destination. Instead, WSMP uses an Application Class

Identifier (ACID) and an Application Context Mark (ACM) to identify the application class and the instance of the application class, respectively [9]. This helps to increase the level of user anonymity since the MAC address and IP address could be used to identify nodes and their presence in the VANET. A WAVE Short Message (WSM) is a message format used for sending messages using the WSMP. WSMP also serves as the transport layer protocol, replacing TCP and UDP for these messages. WSMs may be sent on either the CCH or the SCHs. According to [9], a certificate revocation list (CRL) shall be transmitted as a WSM. The WSM header is 22 bytes in length [15], compared to 40 bytes for the IPv6 header plus 8 more bytes for the UDP header.

2.2 Privacy

The members of the VANET will be made up of publicly and privately-owned vehicles. Private citizens using WAVE applications will have some expectation of privacy while operating their vehicles. IEEE 1609.2 already specifies that privacy is a required service for WAVE applications. The standard defines the term anonymity in the context of privacy, stating that "broadcast transmissions from a vehicle operated by a private citizen should not leak information that can be used to identify that vehicle to unauthorized recipients [9]." More specifically, the ability to link two or more messages from the same sender must become more difficult as the time and distance between messages increases; however, the ability to authenticate the sender of the message as a valid member of the VANET must also be accomplished. The requirements for anonymity and authentication are in the standard, but the methods to provide both at the same time are not covered. The rest of this chapter explains more about the need for privacy, and some suggested methods from other researchers for providing anonymous authentication.

2.3 Public Key Infrastructure

The main components of a public key infrastructure are the users, the certificates, and the certificate authority (CA). Private keys are used to cryptographically sign messages that can be authenticated using the matching public key. Public key certificates are used for authentication to prevent attackers from causing harm. Cryptographically signed messages also provide message integrity; any changes to the message will cause signature verification to fail. Certificates normally have a time period for which they are valid, defined by a start time and an end time, or simply a lifetime [21].

The encryption algorithm specified for use in VANETs by IEEE Standard 1609.2 is elliptic curve encryption, specifically, the elliptic curve digital signature algorithm (ECDSA). Both 224-bit and 256-bit key sizes are allowed in the standard. OBUs use 224-bit keys, while CAs and RSUs use 256-bit keys.

2.3.1 Certificates

Certificates are “data structures that bind public key values to subjects [21].” In other words, a certificate is proof that a public key belongs to a certain user. The certificate authority (CA) generates certificates upon a request from an individual user. The CA is a trusted source that cryptographically signs a user’s public key, thus creating a certificate for the user. The user must trust at least one CA in order to validate certificates; thus, a user that places trust in the CA can also trust that objects signed by the CA are trustworthy.

Researchers have discussed several issues pertaining to the trade-offs between privacy (confidentiality) and authenticity [22]. Pseudonyms have been proposed as a method to handle the opposing requirements of non-anonymous authentication and end-user anonymity. A pseudonym is a short-lifetime certificate that does not contain identity-linking information. Users request pseudonyms from a CA using a longer-lifetime certificate, such as the electronic chassis number or electronic license plate

Table 2. Size of elliptic curve elements, in bytes.

Type	CA and RSU ECDSA-256	OBU ECDSA-224
Public Key	33	29
Private Key	32	28
Signature	64	56
Certificate	135	125
Signed Message Overhead using Certificate	237	219
Certificate ID	10	8
Signed Message Overhead using Certificate ID	110	108
Pseudonym Size	Not Applicable	153

suggested by [8]. The CA that generated the pseudonym holds the linking information in escrow in case of a legal necessity for proving the identity of the pseudonym owner. Changing pseudonyms periodically greatly increases end-user anonymity while still maintaining a reliable means of authentication. 1609.2 includes the specification for the use of non-anonymous authentication using certificates and elliptic curve digital signature algorithms (ECDSA), certificate revocation lists, and end-user anonymity, as described in Annex D.3.3 of the standard.

Every certificate issued by a CA must have a certificate identification number to identify the certificate. The CA generates the certificate identification number by calculating the SHA-256 hash of the certificate. The resulting size of the certificate identification number can be either 64 or 80 bits according to IEEE 1609.2 [9].

Using 224-bit ECDSA for OBUs, the current size of an OBU signing certificate is 125 bytes, 29 bytes of which are the OBU public key. The size of the private key associated with an end-user certificate is 28 bytes. For every pseudonym stored, the certificate and private key must be stored; therefore, 153 bytes of memory are required in the OBU to store the certificate and the private key associated with that certificate. Table 2 summarizes these numbers. The size of the pseudonyms is important because an OBU will change pseudonyms frequently to prevent others from tracking the vehicle's location.

The OBU must store enough pseudonyms to change pseudonyms about every minute while driving, according to Raya, Papadimitratos, and Hubaux in [23]. This equates to about 43,800 pseudonyms per year for an average of two hours of driving per day. Haas, Hu, and Laberteaux in [24] recommend changing pseudonyms every 10 minutes, and driving 15 hours per week. This equates to 4,660 pseudonyms per year, but they recommend storing five years of pseudonyms for a total of about 25,000 pseudonyms per OBU.

2.3.2 Threat Model to Privacy in VANET

One threat to privacy in VANETs is tracking a vehicle based on its radio transmissions. Vehicles will broadcast beacons, safety messages, and other application messages regularly. Any other vehicle in range is capable of storing these messages due to the broadcast nature of WAVE. The beacons and safety messages will require cryptographic signing of the message to prove authenticity and membership in the VANET. The identification sent with signed messages, known as the certificate, is enough to link messages sent by the same vehicle [25-27]; thus, while certificates provide a means for authentication, they do not provide privacy when the same certificate is used for a prolonged period, this the need for pseudonyms in VANETs.

2.3.3 Trade-offs: Security versus Privacy

While frequent changes of pseudonyms will help to protect the identity of the user, it also makes security more challenging. The ability to identify misbehaving users becomes much more challenging since the identity of the misbehaving user will change rapidly; thus it is difficult to identify a series of erroneous messages as originating from the same user.

There is also the difficulty of merely changing pseudonyms so that the change is not able to be tracked. Unless the change is coordinated in some way with surrounding

nodes, the change of a single vehicle's pseudonym can be detected. "Mix zones" [25] have been proposed as a way to conceal the pseudonym change. The mix zone method requires groups of vehicles to coordinate when the pseudonym change will take place. A simplified way to accomplish the pseudonym change would be to synchronize the pseudonym change with the change from CCH to SCH (or between SCH and CCH), synchronizing the change at every minute.

CHAPTER 3

CERTIFICATE REVOCATION LIST

3.1 CRL

When a node's certificate is identified for revocation, the currently used certificate must be revoked along with every pseudonym stored in that OBU. This assumes that whatever caused the current certificate to be revoked will cause future uses of certificates by the same node to also trigger a revocation. Examples that would cause this event include a malfunction in the vehicle's sensors causing erroneous warning messages to other vehicles, or malicious activity by a given vehicle. By revoking all of the pseudonyms, further damage is avoided. The information regarding which certificates are no longer valid, i.e., revoked, is sent out in a certificate revocation list (CRL). The size of the CRL is directly proportional to the revocation rate, the number of nodes in the system, and, for VANETs, the number of pseudonyms used by each vehicle. In 1609.2, the CRL is referred to as the WAVECRL. Actual WAVECRL sizes have been discussed briefly in the literature [28, 29]. Several authors have discussed issues with managing pseudonyms, certificate life-time, and certificate revocation methods, such as in [8, 25, 28, 30-34]. Eichler, in [35], examines revocation in an ad hoc network, but only looks at 100,000 nodes with an assumed 10% annual revocation rate. Also, his work does not take into account VANET conditions.

The literature makes the assertion that the security of the VANET is improved if participants receive timely revocation information, most notably by distributing the CRL as quickly as possible. The common theme among discussed methods to reduce distribution time is to reduce the size of the CRL, since smaller files can be distributed more quickly. Methods for reducing the CRL file size include limiting the cases where revocation is needed and using fewer pseudonyms per vehicle

A certificate revocation list (CRL) is a list of certificate identification numbers that are no longer valid prior to the expiration date of the certificate. The lists are generated and issued by either the actual CA or an entity authorized by the CA. The CRL is cryptographically signed by the CA or authorized entity, so the communication channel and storage medium do not need to be secure since any modification to the CRL during transmission or by other nodes will result in signature validation failure. The CRL is published publicly at a time interval specified by the particular revocation policy. This time interval may be regular, such as hourly, weekly, or monthly, or it may be based on measures other than time, such as a certain number of revocations. The information contained in a CRL includes the expiration date of the CRL, the next time the CRL will be published, and the list of revoked certificates. Each user maintains the CRL and checks the list as part of the message verification process.

In a situation where the rate of revocation is very low, the full and complete CRL will be small and will not change often. Conversely, in a situation where the rate of revocation is high, the full and complete CRL will be large and will change often. The cost, in terms of network resources, of transmitting the full and complete CRL in the second situation will be much higher than in the first situation. A method for reducing the size of CRL information involves sending periodic updates instead of the entire list. Some of the variations of this method are Delta CRL and Over-Issued CRL, discussed in [36, 37]. Arnes completed a very detailed study of CRL policies in [36]. Partitioned CRLs and Bloom filters are another way of reducing the size of the CRL information distributed.

3.1.1 Base CRL

The basic method for distributing CRL information is to distribute the entire list at a specified interval. This method requires the greatest amount of transmitted information to ensure that nodes have the most recent CRL. The last full and complete CRL is

referred to as the base CRL [21, 36]. The base CRL contains a list of every revoked certificate from a single CA.

3.1.2 **Delta CRL**

To reduce the high cost of sending out a new base CRL at every update period, the CRL issuer may transmit only the changes to the base CRL. This method is referred to as Delta CRL. The base CRL is issued at a regular interval that is much greater than the Delta CRL issuance interval. This greatly reduces the amount of revocation information that must be sent. For example, a base CRL could be sent the first day of each month, with daily Delta CRLs that contained only the changes from the base CRL. The size of the Delta CRL increases over time as more changes are added from the base CRL.

3.1.3 **Over-Issued CRL**

Over-Issued CRL is similar to Delta CRL, but instead of issuing only at regular intervals, updates are issued multiple times with over-lapping effective periods. This means that a new CRL will be issued before the normal CRL publication time interval is reached. A variation of this method has been used recently by Haas, Hu, and Laberteaux [24] where every revocation is treated as an update to the base CRL.

3.1.4 **Partitioned CRL**

Partitioned CRLs are presented in [9] as a different method for reducing the size of the CRL. This is a method of organizing certificates into groups to establish a hierarchy of certificates to speed up the distribution and searching of CRLs. The groups are established by the CA and designated in the certificate series field. The certificate series is checked first before searching the rest of the CRL. This could be useful for allowing a single CA to partition a large geographic region of coverage into smaller, geographically different sub-regions.

3.1.5 Compressed CRL (Bloom Filter)

A different way of reducing the size of the CRL involves using types of compression techniques. One method for compressing the CRL information using Bloom filters was introduced by Raya, et al. in [38] and further explored in [25, 32]. In this method, each certificate that is revoked is hashed to a fixed number of bits several times. The resulting hash value for each revoked certificate forms a type of signature. The signatures of several revoked certificates can be combined into a single bit sequence that serves as the Bloom filter. Each time a certificate is received, the same hashes are performed and the resulting value is checked against the Bloom filter. If the signature matches a pattern in the Bloom filter, that means the certificate has been revoked with high probability. Storing CRL information in this manner compresses the size of the CRL considerably since a fixed-length Bloom filter is distributed instead of distributing 8 to 14 bytes for every certificate that is revoked. In [24], Haas, Hu, and Laberteaux examine this method in much more detail. There is a small probability of a false positive occurring when using this method due to hash collisions, which increases as more certificates are added to the Bloom filter. [24] suggests testing a new pseudonym against the currently-possessed Bloom filter to see if the new pseudonym tests positive (revoked) using the Bloom filter. If the pseudonym does test positive, the user should discard the pseudonym and try a different one.

3.2 Pseudonyms

For security and safety reasons, messages must be authenticated to ensure that they were sent by a legitimate member of the VANET. This is especially critical for safety-related messages. Public key certificates are used for authentication to prevent attackers from causing harm. Certificates normally have a time period for which they are valid, defined by a start time and an end time, or simply a “lifetime.”

Because of the opposing requirements of non-anonymous authentication and end-user anonymity, pseudonyms have been proposed as a method to handle these requirements. A pseudonym is a short-lifetime certificate that does not contain identity-linking information. Pseudonyms are requested from a certificate authority (CA) using a longer lifetime certificate. The linking information is held in escrow by the CA that generated the pseudonym in case of legal necessity for proving the identity of the pseudonym user. Changing pseudonyms periodically greatly increases end-user anonymity while still maintaining a reliable means of authentication. While several researchers have proposed novel ideas for generating pseudonyms when needed, their ideas depend on an established and dense infrastructure to support pseudonym distribution [25, 30, 31, 34]. They suggest that OBUs can request new pseudonyms from RSUs whenever new pseudonyms are needed.

Since pseudonyms would ideally have a very short lifetime, on the order of minutes or hours, vehicles must either store large quantities of pseudonyms or make frequent requests for small quantities of pseudonyms. Until sufficient infrastructure is available to support spontaneous pseudonym delivery, users will likely have to store large quantities of pseudonyms in the OBU to preserve anonymity rather than relying on the network for distribution. Distributing pseudonyms from RSUs will increase the amount of network traffic, reducing the capacity for other information to be transmitted on the VANET.

3.2.1 Need for Pseudonyms

Vehicles need pseudonyms to protect their privacy, especially their location information. If the vehicle used the same certificate all the time, anyone could track the vehicle simply by watching where that certificate is used in the network. This is also why vehicles need new pseudonyms when entering a new CA region. Researchers have

examined this situation in some detail, going so far as to recommend changing pseudonyms in "mix zones" [25, 39].

3.2.2 Quantity of Pseudonyms

Raya and Hubaux in [8] propose changing pseudonyms about every minute while driving. They estimate an average driving time of two hours per day. This yields about 43,800 pseudonyms for a year (120×365), which would require almost 7 megabytes of storage memory using 157 bytes per pseudonym. With a limited amount of infrastructure, real-time distribution of new pseudonyms through RSUs is not realistic. A consumer cannot be expected to return to a predetermined location every few days to reload their certificate store. Therefore, a large number of pseudonyms will need to be stored in the OBU for the convenience of the consumer.

X.509 certificates include validity fields for "not before" and "not after" [21] to specify their lifetime. The current trial-use version of 1609.2 specifies a format for the WAVE Certificate that includes a four byte certificate expiration date, but does not include a field for the beginning of the validity period. This means that certificates are valid as soon as they are generated by the CA. We propose adding four bytes to the WAVE Certificate for a "valid after" field to enable the generation of limited lifetime pseudonyms.

An assumption that has been made in some previous papers is that pseudonyms issued by a CA at vehicle registration time are valid for an entire year. However, these pseudonyms do not necessarily have to be valid for an entire year. Instead, a lifetime could be specified for individual pseudonyms using the proposed "valid after" and expiration fields in the WAVE Certificate. Pseudonym lifetimes from one minute to one year could be specified at the time of generation by the CA.

For example, if a driver requests enough pseudonyms for two hours of driving every day of the year, the CA will generate 120 pseudonyms for each day of the year for

Week number	1	2	3	...	50	51	52
Year lifetime	43,800						
Week lifetime	840	840	840	...	840	840	840

Figure 4. Number of pseudonyms valid over a one year period for an average of two hours driven per day, with year-long lifetime or week-long lifetime.

a total of 43,800 pseudonyms. If the pseudonyms all had a lifetime of one year, they would all be valid at any time during the entire year. However, if the CA generated pseudonyms with lifetimes of one week, the driver would have 840 valid pseudonyms to use anytime during each week, as shown in Figure 4. At the end of week 1, any of the 840 pseudonyms that were not used would expire and be deleted from the OBU. Thus, during each week, only 840 pseudonyms are considered valid due to their “valid after” and expiration field values. This also means that only 840 certificates would need to be on the CRL for each vehicle, instead of 43,800 for a vehicle with year-lifetime pseudonyms.

This has the benefit of limited lifetime pseudonyms without the requirement for extensive infrastructure or the added burden on the VANET to generate and transmit spontaneous pseudonyms. At each pseudonym refill, the OBU is loaded with a year of limited lifetime pseudonyms to preclude the driver from having to return to a pseudonym distribution point at frequent intervals.

When the lifetime of the pseudonym is less than one day, 1440 pseudonyms must be generated and stored to cover each minute of the 24 hour period since the owner will not know at the time of pseudonym generation which two hours of the day the pseudonyms will be used. This would also require the generation and storage of $1440 \times 365 = 525,600$ pseudonyms for the year, which is not very practical.

The private key and certificate are deleted from the OBU upon expiration and when a new pseudonym is selected for use. This means that the total number of pseudonyms remaining in the OBU decreases over time. The number of valid

pseudonyms with a lifetime of one year stored in an OBU is expressed in (1), and shown graphically in Figure 5 on the top.

P_{OBU}	Number of pseudonyms stored in the OBU
d	Number of days since the last annual pseudonym refill
P_{used}	Number of pseudonyms used per minute of driving
M_{driven}	Number of minutes per day for which the vehicle has requested pseudonyms

$$P_{OBU} = (365 - d) \cdot P_{used} \cdot M_{driven} \quad (1)$$

Shorter pseudonym lifetimes are available when the “valid after” field is added to the WAVE Certificate; therefore, a general form of (1), where LT is the pseudonym lifetime in days, is shown in (2), and graphically in Figure 5 on the bottom. The

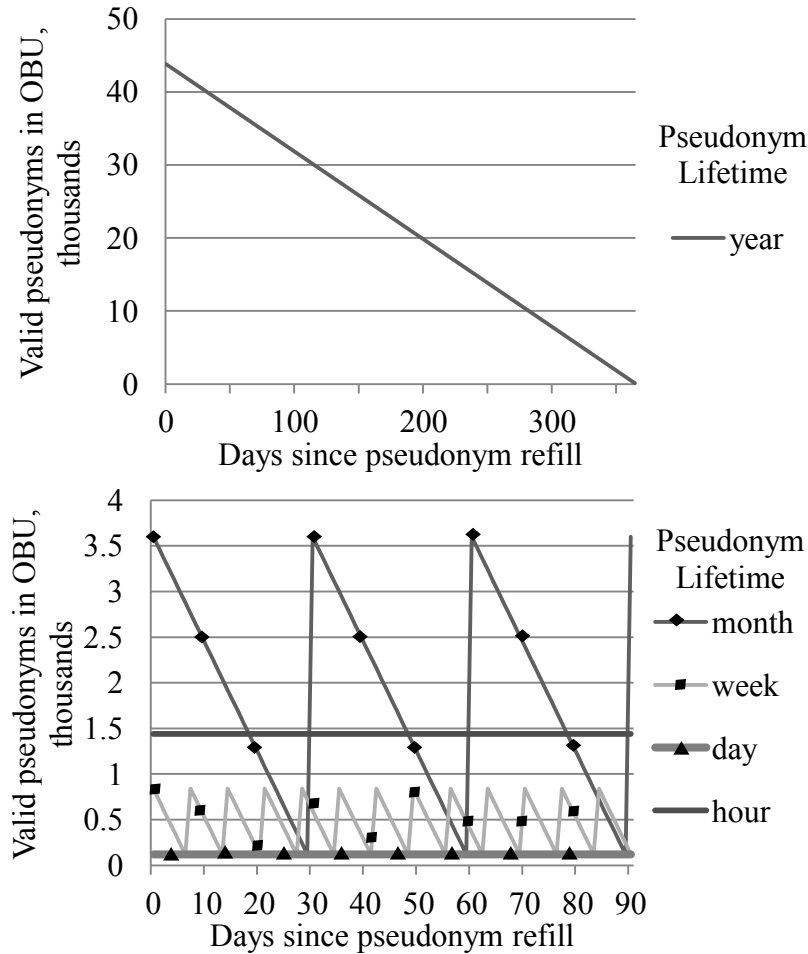


Figure 5. Number of valid pseudonyms in an OBU per day. Lifetime is year-long (top), month-long, week-long, day-long, or hour-long (bottom), provisioned for two hours of driving per day.

immediate reduction in the number of valid pseudonyms present in the OBU is clearly visible.

$$P_{OBU} = (LT - d \bmod(LT)) \cdot P_{used} \cdot M_{driven} \quad (2)$$

3.2.3 Obtaining New Pseudonyms

Researchers have suggested several methods for obtaining pseudonyms, such as yearly refills at the Department of Motor Vehicles, periodic refills at locations such as a gas station, referred to as an "info-fueling station [40]," and pseudonym on demand [39], where pseudonyms are requested and delivered while driving. The latter two methods require the support of RSUs to generate and distribute certificates.

3.3 Revocation Rate

The process of invalidating the certificate is called certificate revocation. Revocation occurs when a certificate needs to be invalidated before the expiration date of the certificate. Revocation informs a user attempting to validate a public key that the private key associated with the public key is no longer valid. When a vehicle's messages become untrustworthy in some fashion, whether by malicious acts, private key compromise, or equipment failure, the certificate that vehicle uses to verify its identity and trustworthiness must not be allowed to be considered valid anymore. Other reasons for revocation may include changing of vehicle ownership from car sales, thefts, or rentals. While changes to vehicle ownership may not actually trigger a revocation when VANETs are implemented, this study does look at that possibility in order to show the extent to which certificate revocation may affect the VANET. We will refer to vehicles that have certificates requiring revocation as "misbehaving." This term encompasses all reasons for revocation.

3.3.1 Malicious Activity

This category includes users that intend to influence the behavior of other vehicles in the VANET, either directly or indirectly. Direct measures are events such as sending fake location messages, false safety messages that cause other vehicles to brake suddenly, or traffic information that is incorrect, causing vehicles to alter their travel routes. The rate of malicious activity represents the greatest degree of uncertainty. With the wide variety of applications made possible by VANETs, it is highly likely that someone will attempt to take advantage of the system and its users. In [22], Parno and Perrig describe several categories of such adversaries, including greedy drivers, snoops, pranksters, industrial insiders, and malicious attackers. A large range of activities are described in which a user manipulates the network, whether it is to clear the road ahead, track vehicles, disable information flow, load malware, or inflict physical harm. When a node can be isolated as the source of bad information or other problems in the network, its certificates should be revoked for the safety and betterment of the rest.

3.3.2 Private Key/Certificate Compromise

Private key compromise is when the owner of the private key suspects that someone else is also in possession of that private key. Whoever possesses the private key can request certificates and sign messages as if they were the actual entity with whom the private key is associated. Whenever a private key is compromised, all certificates associated with that private must be revoked and no new certificates may be issued for that private key. Depending on the actual implementation of VANET security, people with specific knowledge may be able to access and possibly corrupt or copy material in the OBU. Physical security of vehicles equipped with VANET technology cannot be maintained to always prevent physical access by malicious entities. Although trusted computing platforms are targeted for use in VANET applications, it is not unreasonable to assume there will be vulnerabilities in the implementations [41].

3.3.3 Equipment Malfunctions

This category includes hardware failures of sensors, radio equipment, trusted platforms, location determination equipment (such as the global positioning satellite device), etc. that cause erroneous messages to be sent out into the VANET. Software errors may also contribute to this rate. Electronic devices are imperfect, so malfunctions are possible, causing erroneous or faulty messages to be sent from an otherwise innocent source. This rate is expected to be low, but no actual numbers have been determined yet. Until the device can be repaired, the certificates associated with it should be revoked so that other nodes in the network discard the misinformation.

3.3.4 Change of Ownership

While specific reasons for revocation are not addressed in the IEEE 1609 standards, it is likely that a change in vehicle ownership will trigger the issuance of new certificates; thus, the previous certificates must be invalidated prior to their normal expiration. Change of ownership events represent the different causes for possession of the vehicle to change, which may or may not prompt a change in certificates and pseudonyms. The decision as to whether or not this category is included in the revocation process has not been discussed in previous literature. When a car changes ownership, both the seller and the purchaser may be concerned that there could be an error in determining responsibility in the event one of the car's certificates is associated with a malicious event. Thus, valid certificates remaining in the OBU should be revoked when a change in ownership occurs to ensure that unused certificates do not fall into the wrong hands. Simple deletion of the keys stored in the OBU may not be sufficient assurance that the certificates will not be used elsewhere. To ensure the certificates cannot be considered valid by any entity, they must all be revoked. Several events where vehicles change hands include: rentals, theft, sales, leases, and accidents. Because of their short duration, rentals are not considered a change in ownership.

3.4 CRL Size

The size of the CRL depends on the number of vehicles, size of the CA region, frequency of CRL issuance, the rate at which certificates are revoked, the number of pseudonyms each vehicle possesses, and the pseudonym lifetime. In the following section, we try to determine the number of revocations for the entire United States as a starting point for calculating CRL file sizes.

Portions of the change of ownership data are available for the United States for different years. We used data from 2005, since it is the most recent year for which data is available for all of the categories examined. According to [42], published annually by the FBI, about 1.2 million car thefts occurred in the US in 2005. The National Transportation Statistics, 2007 [43] shows new and used car sales and leases to be about 61 million in 2005. According to [43], the number of accidents in 2005 that caused injury was 1.8 million. The total number of registered vehicles in the United States in 2005 was reported as 243 million by [44].

The number of ownership changes is 26.3% of the total number of vehicles operating in the United States. Adding in 5% for revocations due to malfunctions, malicious activity, and compromise results in a 31.3% annual revocation rate. While data is not available for malicious activity, compromises, or malfunctions, estimates of 5% and 10% have been used in other papers for revocation rates [28, 35, 36, 38]. Table 3

Table 3. Annual revocation rate triggers, 2005.

Type	Number (millions)	Percentage
Total vehicles	243	100%
Malfunctions, malicious activity, and compromise	12.2	5%
Thefts	1.2	0.5%
Sales, New	13.5	5.6%
Sales, Used	44.1	18.1%
Leases	3.4	1.4%
Accidents	1.8	0.7%
Total Revocations	76.2	31.3%

shows the annual revocation triggers for the entire United States.

The expected number of pseudonyms at the time of revocation is one half of the vehicle's issued pseudonyms. The annual number of revocations can be found using (2) to find the number of valid pseudonyms remaining per OBU, P_{OBU} , with the number of days, d , set to 365/2, and lifetime, LT , set to 365. This value is then used in (3) to find the total number of annual revocations. If every vehicle in the United States was provisioned with 43,800 pseudonyms for an average of two hours of driving per day, it is expected that at the rates specified in Table 3 about 1.37×10^{12} pseudonyms would require revocation over the year.

R	Total number of certificate revocations
N	Total vehicles in the system = 243 million
OCP	Ownership changes using pseudonyms = 20.8%
MMC	Revocations due to maliciousness, malfunctions, and compromises = 5%
OCI	Ownership changes with one certificate = 5.6%
P_{OBU}	Valid pseudonyms in the OBU at revocation time

$$R = N \cdot (OCP + MMC) \cdot P_{OBU} + N \cdot OCI \quad (3)$$

Figure 6 shows the number of revocations per hour when pseudonyms are requested for two hours of driving per day, with different pseudonym lifetimes. These charts also show the dramatic reduction in the number of revocations due to using limited lifetime pseudonyms instead of year-long lifetime pseudonyms.

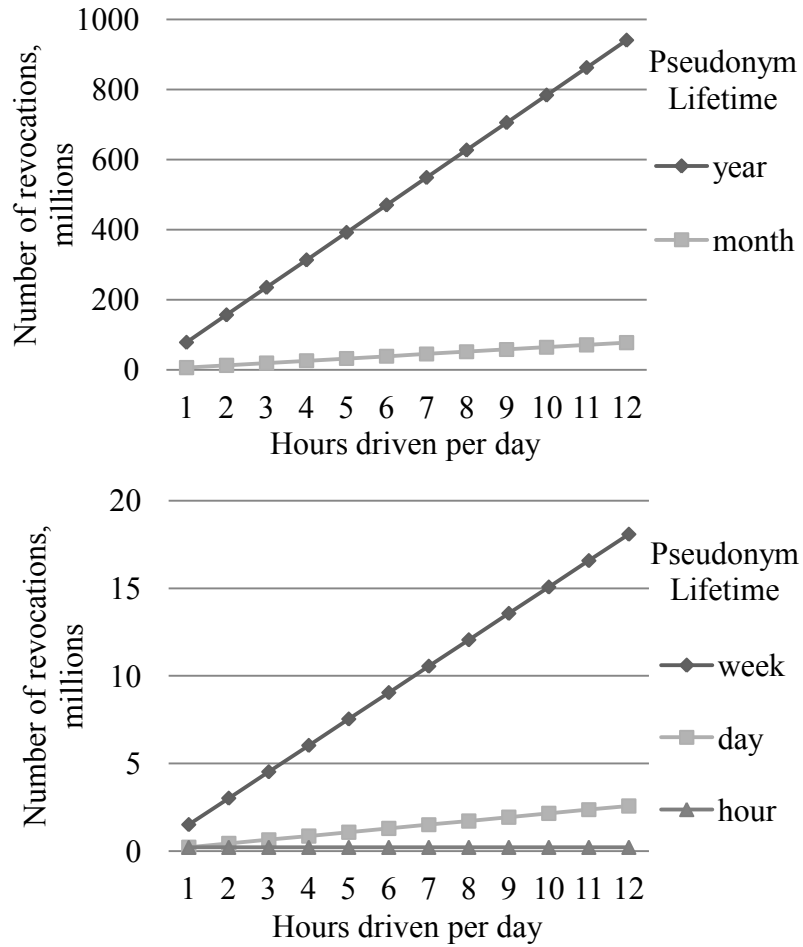


Figure 6. Number of hourly revocations over varying pseudonym lifetimes.

3.5 Certificate Authority Regions

Another factor that influences the size of the CRL is the number of vehicles in the CA region. The CA issues all pseudonyms for a specific geographic region. When vehicles move from one geographic region to another, the vehicle must acquire new pseudonyms for that region to protect the privacy of the "foreign" vehicle. A hierarchy of CA responsibility is established so that a long-term certificate from one CA can be used to acquire pseudonyms from a different CA, as long as the pseudonym-issuing CA trusts the long-term certificate-issuing CA, i.e., has their public key. New pseudonyms should

be acquired rather than using out-of-region pseudonyms to preserve privacy. If an out-of-region pseudonym is used, it will be easily recognizable as such [25].

There is a trade-off between the size of the CA region and size of the CRL, as well as the management complexity of the entire PKI system for VANETs. The least complicated region to manage would be a single large area, such as the entire United States, with a single CA responsible for every certificate and pseudonym. This would also result in the largest CRL since every revocation would appear on the CRL. This relationship is shown in (4) and (5).

$$\text{CRL Size} \propto \text{Size of Region} \quad (4)$$

$$\text{Management Complexity} \propto 1/\text{Size of Region} \quad (5)$$

Bellur in [28] gives some analysis on region size and provides some techniques for managing transitions from one region to another, including maintaining multiple sets of pseudonyms. Thus, a vehicle would have a set of pseudonyms for their "home" region plus additional sets of pseudonyms for regions adjacent to their "home" region. He also explains the proportionality between the CRL size and the size of the region in more detail.

Another possibility for adjusting the size of the CRL with large regions is to use the partitioned CRL method of distribution, along with the "series" field of the WAVECRL. The partitioned CRL would be used to distribute regional CRLs, designated by the series field. Thus, when a vehicle plans a destination in a different region, the vehicle can request the CRL for that region. Another method involves distributing a region's CRL to adjacent regions and vehicles can decide to store or ignore CRL pieces based on the series. This would require an additional 4 bytes, or less if a hash value or cardinal direction bit is used, of overhead per piece in the header to designate the series of the CRL piece.

3.6 The WAVE CRL

The CRL structure defined in [9] has four main parts: version, signer, unsigned_CRL, and signature. These fields verify the authenticity and integrity of the CRL by describing the version and which CA created it, and also provide the wanted information: revoked certificates. The CRL version, as specified in the current release of [9], is always 1. The signer field is information about the CA issuing and signing the CRL. The unsigned_CRL, which contains the useful information, is composed of additional fields: type, crl_series, ca_id, crl_serial, start_period, issue_date, next_crl, and entries. The series, start period, issue date, and next CRL all help identify the CRL so that the user knows what list they have received as well as when the next update can be expected. As discussed in sections 3.1.4 and 3.5, the series field can be used to designate a geographical or logical partition of the CRL. Entries can contain either the ID of a revoked certificate only, or both the ID and the expiry date. Including the certificate expiry date helps to reduce the number of stored revocations over time. Each WAVECRL may contain up to a maximum of $2^{64}-1$ entries. Finally, the CA issuing the CRL signs the unsigned_crl portion and appends the ecdsa_signature.

Using the fields described above and the lengths found in Annex D.3.3 of the 1609.2 standard for other structures, the size of a CRL was determined to be 230 bytes plus 14 bytes per certificate. The certificate ID is found by generating the SHA-256 hash of the certificate and then taking the low-order 10 bytes of the hash output. The expiry date is an additional 4 bytes, resulting in 14 bytes per revoked certificate. This information is summarized in Figure 7, which is formatted according to the structures in Annex C of 1609.2.

Length (bytes)	Field		
1	version = 1		
1	signer	type = certificate	
135		CA certificate (see 1609.2 C.2 for details)	
1	unsigned_crl	type	
4		crl_series	
8		ca_id	
4		crl_serial	
4		start_period	
4		issue_date	
4		next_crl	
10		entries =	id
4		id_and_expiry	expiry
32		signature	ecdsa_signature
32	s		

Figure 7. WAVECRL elements.

The actual size of a CRL can be highly variable, depending on a few factors. Clearly, the size of the CRL directly corresponds to the number of pseudonyms needing revocation, which is dependent on the revocation rate. Based on the earlier assumption that a pseudonym is removed from the OBU upon selection of a new pseudonym, the number of pseudonyms in the OBU needing revocation will decrease over time. Assuming a uniform distribution for the revocation rate, one-half of the valid pseudonyms in an OBU will need to be added to the CRL, on average.

The CRL will be generated at a pre-determined time interval, e.g. every hour. Using the number of revocations from Figure 6, the WAVECRL size can be calculated by multiplying the number of revocations by the size of each revocation, 14 bytes. The WAVECRL elements in Figure 7 add an additional 230 bytes to the final size. The resulting WAVECRL sizes are shown in Figure 8 for different values of driving time and pseudonym lifetime [29]. For example, if enough pseudonyms were generated for an average of two hours of driving per day with a lifetime of one week, the hourly CRL size is 42.2 megabytes per hour. Since these are the new revocations each hour, the CRL size will be additive over the day; thus, the total daily CRL size for previous example would be 1012 megabytes (24 hours times 42.2 megabytes per hour). If delta-CRL is used to

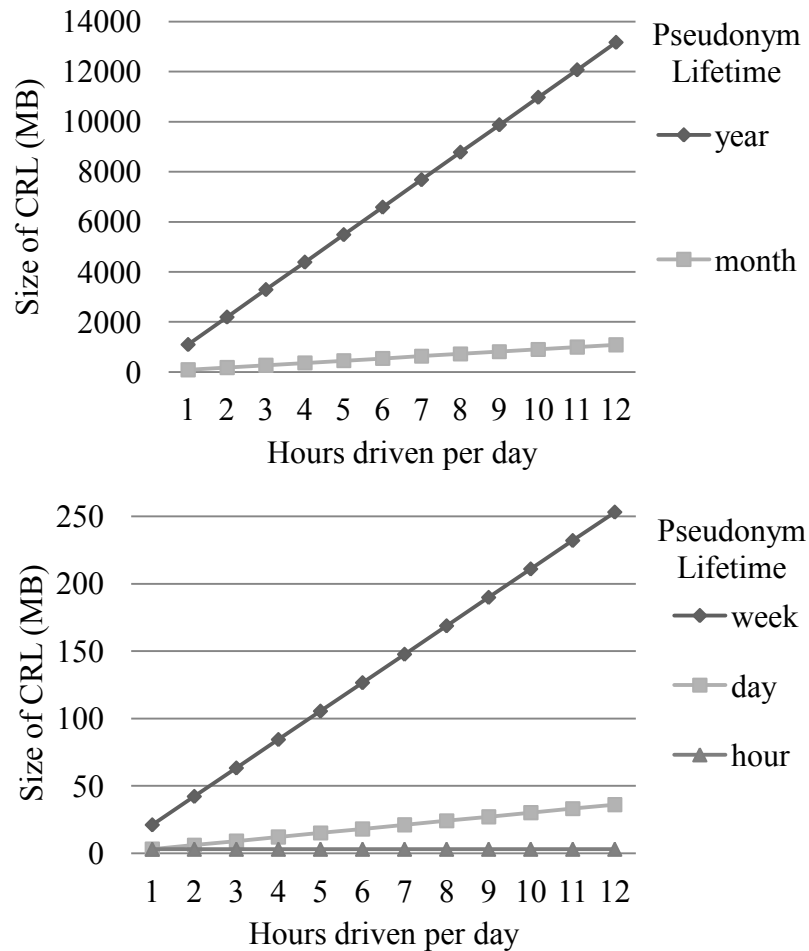


Figure 8. Size of hourly CRL over varying pseudonym lifetimes.

send out hourly updates with the base-CRL sent out once per day, a summation of the hourly CRLs would be about 12 gigabytes of data over the day, i.e., each hour's Delta-CRL size is the hour number times the base-CRL size, assuming uniform revocations throughout the day (thus the base-CRL size is equal to one hour's Delta-CRL size). However, since Delta-CRL sends all changes since the base-CRL was sent, only the base-CRL and the most recent Delta-CRL need to be stored in the OBU. In this example, the base-CRL is about 42.2 megabytes, and the last Delta-CRL of the day is about $23 \times 42.2 = 971$ megabytes. Assuming only yearly consumer visits to a pseudonym update facility, Figure 8 compares the file sizes of CRLs. The year line represents the case where our proposed "valid after" field is not used. All other lines (month, week, day, hour) assume the use of our new "valid after" field. By comparing to the CRL file sizes associated with the year line, significant reduction in CRL file sizes is shown when the proposed "valid after" field is used.

The previous revocation numbers and CRL file size are based on the entire United States as a single CA region. Dividing the United States into smaller regions will also divide the number of revocations and the CRL file size by the same factor. While using more CA regions increases the complexity of pseudonym and CRL management due to vehicles crossing regional boundaries, this method does very effectively reduce the CRL file size, which will contribute to quicker distribution of the CRL.

CHAPTER 4

VANET EFFECTS ON CRL DISTRIBUTION

4.1 Introduction

Several factors influence the distribution of the CRL in the VANET, including the encoding method used and the available channel capacity. We look at these factors now to see how they impact the selection of methods and the expected outcome.

4.2 Encoding Methods

The CRL file may be a large file, depending on the number of vehicles in the region of the certificate authority, the revocation list distribution protocol, and the revocation rate. CRL file sizes in excess of 1 megabyte will be common. The CRL can be sent out in broadcast mode to the members of the VANET using a distribution method where each RSU transmits the entire CRL over and over again. However, rather than sending out the entire file as a stream of packets where every packet is needed to reconstruct the file, coding techniques can be used to break the file into packet-sized pieces using encoding to generate redundant pieces. This will accomplish two things: a single missing piece will not prevent an OBU from recovering the CRL, and OBUs can forward individual pieces even if they do not possess the entire file.

When no coding method is used, every piece must be downloaded. Coding techniques reduce the impact of the piece problem, also referred to as the coupon collection problem, which exists in file sharing techniques that simply split a file into multiple pieces, such as BitTorrent™ [45]. To download a copy of the original file using BitTorrent™, the destination node must download a copy of every piece of the file from peers. The piece problem occurs when there is difficulty downloading one or more pieces due to the lack of availability of some pieces at other peer nodes. The missing

pieces prevent the destination node from completing the download. While BitTorrent™ works well in a wired environment, it does not work as well in a wireless environment due to the limited distances over which the physical layer is able to reach, and because some nodes join and leave the network quickly.

Based on information theory principles in [46], the number of pieces that must be downloaded to guarantee every piece of a file is received is found as

$$pieces\ downloaded = pieces\ in\ file * \ln\left(\frac{pieces\ in\ file}{1 - \% \ guarantee}\right). \quad (6)$$

The cost of not coding is shown in Table 4. This table shows the number of pieces required for download of a 2000-piece file at varying percentage levels of guarantee that all 2000 pieces of the file are received. This means that if pieces are broadcast using a random selection process and that every piece must be received to reassemble the file, on average almost 20,000 broadcasts must be done to have a 90% guarantee that all 2000 pieces were selected for broadcast.

Some VANET file-sharing models discuss the use of Network Coding or Erasure Coding as a means of making data dissemination throughout a network more efficient and timely [47-51]. Network coding and Erasure coding techniques mitigate the piece problem by coding the file in a manner such that the file can be reconstructed from a subset of the pieces generated. These coding techniques are very effective in network conditions like those in a VANET, where nodes join and leave frequently, and nodes are in contact for short periods of time.

Table 4. Required pieces to download for a 2000-piece file with no coding methods used.

% guarantee	# required to download
90.00%	19,807
95.00%	21,193
99.00%	24,412
99.90%	29,017
99.99%	33,622

Two parameters that are necessary to describe the coding techniques are coding rate and coding overhead. The coding rate describes the percentage of additional pieces encoded. The coding overhead describes the percentage of additional pieces needed to reconstruct the file. Thus, a 200% coding rate means that twice the number of pieces are generated from a file. A coding overhead of 5% means that 105% of the number of file pieces are necessary to reconstruct the file. For example, if a file can be broken into 2000 pieces, a 200% coding rate generates 4000 pieces. A coding overhead of 5% means that 2100 pieces (of the 4000) must be received to reconstruct the file [52].

Two predominant coding techniques are discussed by other researchers [47-49, 52-57], network coding, and a type of erasure coding called raptor coding. The primary difference between the two techniques according to Fujimura, et al. in [49] is where the coding is accomplished. Erasure coding entails all encoding done by a single source. Original source-encoded pieces are shared in order to re-create the original file. When using erasure coding, a node will simply send out the same piece it received without making any changes. Network coding entails coding at each node, combining the file pieces possessed at that node into new pieces that contain aggregated data combined from existing pieces. These new pieces are shared with other nodes. When a node receives a network coded piece, it is checked for linear independence with currently possessed pieces, requiring greater processing capability at every OBU. Erasure coding has less overhead, both in packet overhead to carry the coding information, and in processing overhead to reconstruct the file. Raptor coding is a more specific type of erasure coding that is better suited for file distribution in VANETs. Raptor codes are rateless erasure codes capable of producing a large number of redundant file pieces [52, 58].

4.2.1 Erasure Coding

Using erasure coding, CRL pieces are all generated by the CA. Each piece may be signed individually by the CA, which adds 110 bytes of overhead per piece, as shown

in Table 2. Only 110 bytes of overhead is required since only the CA certificate ID needs to be sent with the pieces since every node must possess the CA certificate. This increases the security of sending pieces since each piece is signed by the CA. Piece generation by the CA also makes the most efficient use of processing time. This means that OBUs only have to check the signature of each piece. No processing is necessary to pass the piece on to a neighbor. Since the content of the piece is not confidential, there is no need to encrypt pieces, so time is spent only on signature verification. Decoding the pieces is $O(k \log(\frac{1}{\epsilon}))$, where k is the number of pieces and ϵ is the coding overhead [52].

4.2.2 Network Coding

Network coding, as introduced by Ahlswede et al. in [53], makes very efficient use of the available channel capacity by encoding data in such a way that all existing pieces are used to generate new pieces to forward. This is especially applicable to multicast situations, such as the distribution of files from one source to many nodes. The result of this is that "new" pieces are generated by each node by forming random linear combinations of existing data pieces and forwarding them to other nodes. The cost of processing new pieces is $O(n)$ where n is the number of pieces. According to [57], the encoding may take more time than the transmission of the piece. Decoding can be a lengthy process, as explained by Kötter and Kschischang in [59], on the order of $O(n^2)$ where n is a measure of the vector dimensions used to encode the data based on the amount of data transmitted. Another source, [60], states that decoding is $O(n^3)$. Thus, as the size of the CRL increases, the complexity of decoding the file increases rapidly. The encoding vector also grows as the file size increases, by about one byte per piece sent [57]. Thus, for a file consisting of 1000 1-kilobyte pieces, an encoding vector overhead of 1000 bytes per piece sent is needed, doubling the size of the packet.

Although network coding presents great benefits for the use of channel capacity, the process of using network coding to distribute CRLs in a VANET is very challenging

for two reasons: possible injection of false pieces, and a non-zero error rate on the channel. Both situations lead to severely corrupted CRL files. Since every node can generate new pieces that are supposed to be linear combinations of other pieces, false pieces could be injected by either a malicious or malfunctioning node. The false pieces would be difficult to detect until they were decoded. This problem is compounded by the fact that receiving nodes will use these corrupted pieces to combine with their existing pieces, passing along more corrupted pieces. Noise on the channel could also corrupt broadcast pieces, although the layers below the application where the pieces are used should be able to detect corrupted packets. Noise affects all packets, but if a corrupted piece is accepted in a network coding scenario, not only is that piece corrupted, but also all pieces that use the corrupted piece to linearly combine to form new pieces.

Because of the high computation cost and the security issues, using network coding is not as well suited as erasure coding for distributing CRLs in VANETs.

4.2.3 Network Coding Generation Concept

One feature of network coding that may prove useful in other methods is the concept of generations. Generations are parts of the total file that are encoded separately, that is, each generation must be downloaded to re-create the entire file. This concept can be extended such that the CRL can be broken into smaller, stand-alone parts of the entire CRL. This would allow large CRLs, which are basically just a listing of certificate IDs, to be broken into smaller parts and used independently until the rest of the parts can be

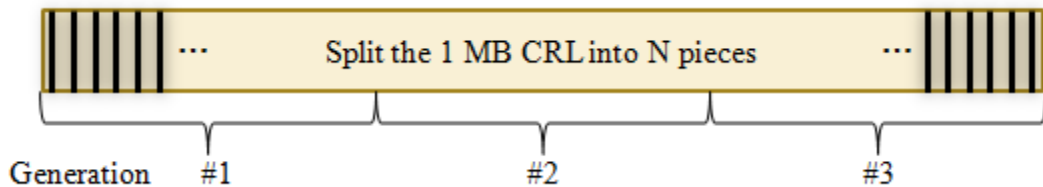


Figure 9. Example of network coding generations.

downloaded; thus, a vehicle could access portions of the CRL faster than waiting for the entire CRL. The concept of generations is shown in Figure 9 [61].

4.3 Available Channel Capacity

In [62], Eichler evaluates DSRC for a single broadcaster to determine how many packets can be sent during a single control channel interval. We present updated information here based on parameters from IEEE 1609.4 [16] and IEEE 802.11p [17] for the service channel interval.

Given that the CCH and SCH interval is 50 milliseconds minus the guard interval of 4 milliseconds, each interval is only 46 milliseconds in duration for transmission of data. The number of packets that can be sent during a single CCH or SCH interval can be determined by the interval time divided by the amount of time it takes to send a single packet over the channel. The data rate for the broadcast channel is 3 million bits per second.

A beacon packet is 100 bytes of payload plus 72 bytes of additional headers, so it takes 459 microseconds to transmit a single beacon packet. Using the values in Table 1 for "Best Effort" access class, we determine that on average 75 beacons can be transmitted during the CCH interval after adding an additional 152 microseconds per beacon for the AIFS and the average contention window timers.

The length of a data packet determines the number of packets that can be transmitted during the SCH interval. The number of packets that can be transmitted during a single SCH is shown in Figure 10 for several different packet lengths. The throughput drops at certain packet sizes due to having to receive a complete packet to be counted toward the throughput, so all packet numbers are rounded down to integer values.

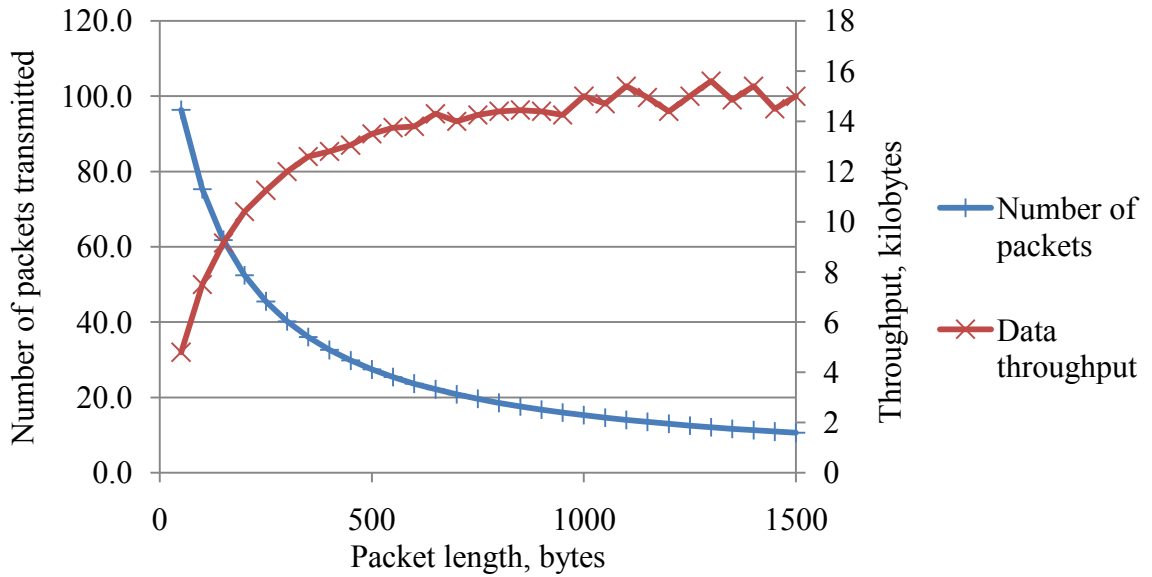


Figure 10. Number of packets transmitted and throughput as a function of packet length for a single SCH interval.

This becomes an upper limit for the number of packets that can be transmitted during a single interval. Collisions will reduce the number of packets successfully received. Based on the best case of the upper limit of packets received, an estimate of the download time for a 1 megabyte CRL file is presented in Figure 11.

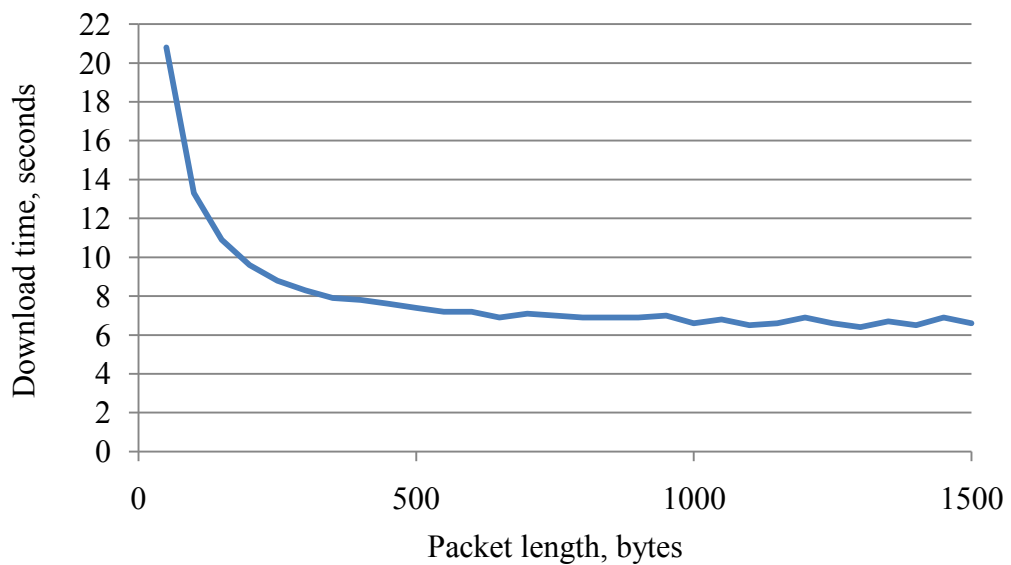


Figure 11. Download time for a 1 megabyte file with different packet lengths.

CHAPTER 5

CRL DISTRIBUTION METHODS

5.1 Introduction

Distribution of the CRL in a VANET is a multi-step process. Figure 12 is used to illustrate the process. The first step is the certificate authority (CA) generating a list of OBU certificates that require revocation (1). Once the CRL is formatted and cryptographically signed by the CA according to the specification in [9], encoding methods are used to split the CRL file into multiple pieces (2). The CA also signs each piece to protect the pieces from being modified. The next step is for the CA to distribute the CRL pieces to the RSUs (3). The RSUs receive all of the CRL file pieces from the CA and determine the authenticity of the pieces by verifying the signature on each piece. After the pieces have been verified, they can be broadcast to the OBUs within radio range of the RSU. V2V communications are used to pass the file to those vehicles not within range of the RSU. The OBUs will gather pieces of the CRL until enough of the file pieces are received to reassemble the file, at which time the CA signature is used to verify the file authenticity [63].

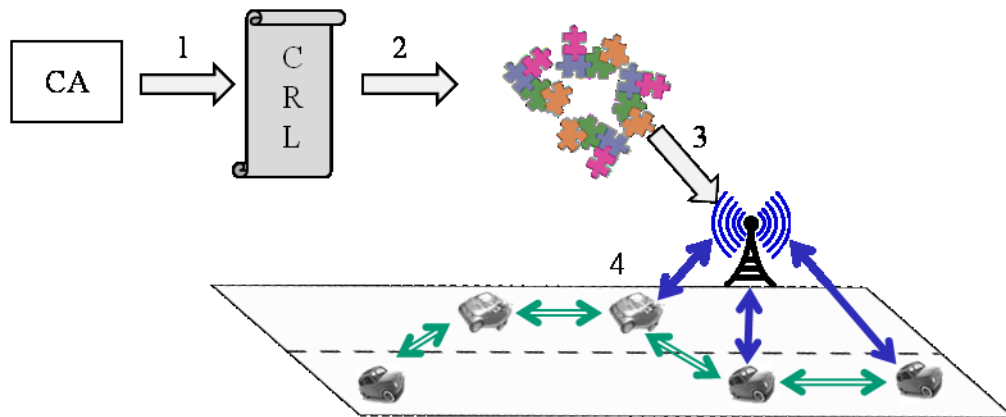


Figure 12. CRL distribution in a VANET.

Taking advantage of the coding techniques discussed in Chapter 4.2, only a certain number of pieces must be stored to reconstruct the file, rather than needing every piece of the file when coding techniques are not used. Another benefit of using raptor codes is that the CA can pre-code the entire CRL into pieces before distributing the file to the RSUs. This uses the higher processing capacity of the CA to generate the pieces.

5.2 Current Methods of CRL Distribution (Related Work)

We have studied methods of certificate revocation and file distribution in detail through literature surveys and discussions with the authors of those papers. Some research in the area of distributing CRLs in VANETs has been done previously by other research groups. Two groups have proposed solutions to the CRL distribution problem. The first group developed solutions that depend on extensive infrastructure for their methods to work. The second group looked at using both RSUs and other vehicles to distribute the CRL.

5.2.1 RSU-only (EPFL)

A group from *Ecole Polytechnique Federale de Lausanne* (EPFL) in Switzerland, led by Dr. Jean-Pierre Hubaux, accomplished a great deal of work on VANET security, such as certificate revocation techniques and privacy-protection techniques [13, 23, 25, 32, 38, 64-66]. While they discussed concepts for CRL distribution, their approaches have not considered high vehicle traffic densities, the large-scale scope of the VANET, or alternative reasons for revocation; thus, they looked at a simplified model for evaluating their methods. They have also not considered V2V communications in their methods, until recently in [47].

EPFL's main contribution to the CRL solution is their proposed three-part scheme consisting of two methods that require RSUs and a third method used outside of RSU range for ignoring messages from suspected misbehaving nodes, first introduced in [38].

These methods include deleting certificates directly from the vehicle hardware using messages sent from RSUs, using Bloom filters to generate compressed revocation lists, and vehicles cooperatively determining possible malicious or malfunctioning nodes [23, 25, 32, 64-66]. These three methods are employed together to cover several situations for revoking certificates or ignoring misbehaving nodes. This protocol set covers situations where the nodes will not always be guaranteed access to the CA; therefore, the off-line method is proposed to cover those times when the CA is not reachable, or more accurately, when the CA is not able to reach the nodes. Assumptions in this protocol are that the vehicles already have several hardware components installed, including a trusted component for securely storing and processing cryptographic material, and that RSUs are readily available.

5.2.1.1 Revocation of the Trusted Component

The first on-line revocation method described by the group from EPFL in [32] is Revocation of the Trusted Component (RTC). The trusted component is a part of the OBU that stores the valid certificates for a vehicle, signs messages, and performs encryption and decryption functions. RTC is a series of messages sent from the CA through an RSU to the misbehaving vehicle's trusted component. These messages serve to effectively delete all valid certificates from the vehicle, preventing the OBU from signing messages. This method assumes the presence of infrastructure to send the messages to the trusted component. It also assumes that the trusted component will follow the directions to delete the certificates; however, this is not necessarily a valid assumption, since the trusted component may have been compromised. Several methods for compromising trusted components are discussed in [41]. To ensure that messages from this OBU are not considered valid once the certificates have been revoked, revocation information must also be distributed via CRLs.

5.2.1.2 Revocation Using Compressed CRL

The second on-line revocation method discussed in [32] is Revocation using Compressed Certificate Revocation Lists (RC2RL). This method sends out certificate revocation lists that are compressed using Bloom filters to make the lists smaller. This method reduces the size of the CRL by using about half the number of bytes to specify the certificate ID for revocation. This shortens the already hashed value so that the number of false positives increases.

RTC prevents the misbehaving node from sending out signed messages by directly removing the certificates from the trusted component, while RC2RL notifies other nodes in the VANET that the certificates have been revoked. It is necessary to notify nodes in the event that RTC failed to delete the certificates in the misbehaving trusted component. These methods are both dependent on vehicles being able to communicate with the CA through RSUs.

5.2.1.3 Local Eviction of Attackers by Voting Evaluators

The off-line method from EPFL is called Local Eviction of Attackers by Voting Evaluators (LEAVE). This is more of a warning method rather than an actual revocation method. The Misbehavior Detecting System (MDS) is a part of the LEAVE protocol. MDS functions as a type of intrusion detection system to recognize patterns of misbehavior using on-board sensors to verify information sent by nodes. Once MDS identifies a misbehaving vehicle, LEAVE warns other vehicles in close proximity about the misbehaving node. This results in a temporary suspension of accepting messages from the accused misbehaving node. Once the vehicles are able to contact a CA, the CA may initiate actual revocation of the misbehaving node's certificates using RTC and RC2RL.

5.2.2 V2V Methods

Combining V2I with V2V communication enables a new set of CRL distribution methods to be developed. Figure 13 illustrates the concept of integrated V2I and V2V communication. The exchange begins with a vehicle entering the broadcast range of the RSU (1). As the vehicle travels past the RSU (2) it receives and stores several CRL pieces. Once the vehicle drives beyond radio range of the RSU it will eventually encounter another vehicle (3). Each vehicle may have some CRL pieces. As the vehicles come into radio range of each other (4) they broadcast pieces of the CRL. Any new

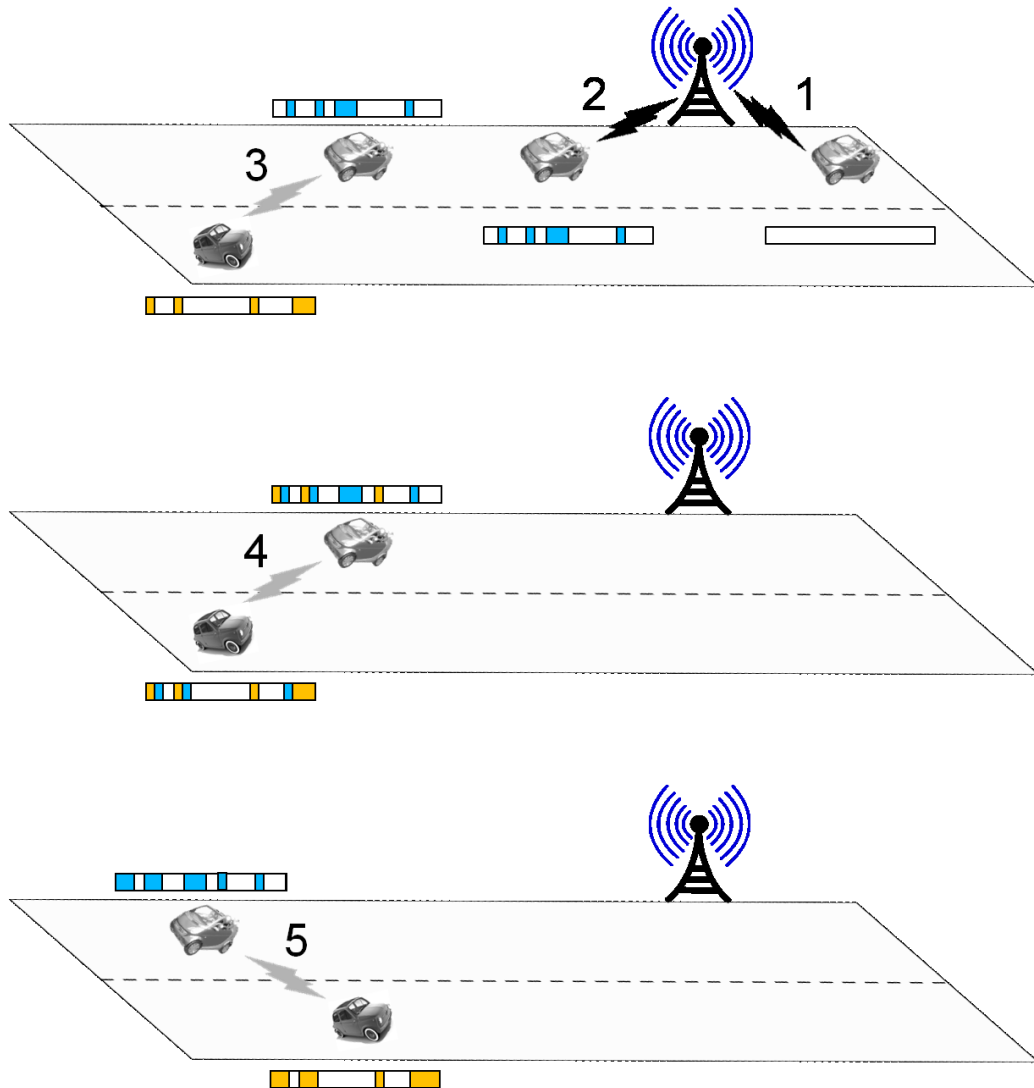


Figure 13. V2I and V2V integration in VANET.

pieces are stored by the receiver (5). Duplicate pieces received by the OBU are discarded.

5.2.2.1 EPFL V2V CRL Distribution

The one paper by EPFL that did discuss using V2V communications is a paper dealing specifically with distribution methods of CRLs [47]. Their assumptions were that there was not a dense infrastructure, RSU-to-RSU communication was not required, and CRLs were regionally focused. Their scheme has three basic elements: small CRLs due to only sending information about a certain region; breaking the CRL into small pieces; and using low-rate broadcast of CRL pieces from RSUs. Fountain codes and Erasure codes are mentioned as the method for encoding the CRL into small pieces.

5.2.2.2 Epidemic CRL Update Distribution

Laberteaux, Haas, and Hu proposed an epidemic distribution method in [50] that is most closely related to this proposal, yet they do not go into the implementation details other than stating that CRL updates will be distributed using V2V communications. They discuss the difficulty of picking the proper level of detail and accuracy for simulations to study the problem. A simple "infection" model was used in their study for proof of concept. Vehicles are considered to receive the CRL update whenever they are within 100 meters of another vehicle for 2 seconds, so no form of radio propagation or network congestion was involved. The simulation study was massive in size, accounting for 260,000 vehicles over approximately 354km x 263km of area surrounding Zurich, Switzerland [67, 68]. Only CRL updates are distributed in this model, not the entire CRL. The study did show that by using V2V communications with a single RSU, the CRL update was distributed to over 99% of the vehicles at the end of the 9000 second simulation. This compares to a completion rate of about 92% using 325 RSUs with no V2V communication and only a 0.1 second contact requirement.

This group also proposed a method of linking certificates through the use of a secret key [24]. This method distributes a single secret key that is used by the OBU to generate hashes of every pseudonym the revoked vehicle owns. While this does greatly reduce the size of the CRL, it does associate every pseudonym to the owner. This violates the premise of maintaining privacy in the VANET. By associating every pseudonym, the owner's previous locations can be traced by examining stored communications. Just because the certificates are revoked does not mean that the reasonable expectation of privacy should be waived. This is especially true in the case of an equipment malfunction. Just because a vehicle has equipment problems, the owner of that vehicle should not forfeit their privacy. Distributing the Bloom filter instead of the secret keys would alleviate the problem of revealing all of the pseudonyms and also relieve the OBU of having to conduct multiple hashes to generate the Bloom filter.

5.2.2.3 Car Torrent

A group from the University of California, Los Angeles, led by Dr. Mario Gerla, conducted research involving vehicular peer-to-peer networking. Their methods were designed for general-purpose file sharing rather than CRL distribution. For the purposes of general file sharing in VANETs, Lee, Seung-Hoon, et al. advocate Car Torrent, a peer-to-peer file-swarming protocol similar to BitTorrent™ in which users not in contact with an RSU can still download parts of files from their peers [69]. They use a file-swarming protocol based upon SPAWN, which is previous work by one of the co-authors [70]. Car Torrent uses k-hop limited-scope broadcasting, known as gossiping, and a piece selection strategy to optimally download files from peers. As the vehicle density increases, the overhead due to the gossiping and the routing algorithms quickly increases the delay of file transfers. Also, since there is no coding involved in this method, every piece of the file must be downloaded to recover the original file.

5.2.2.4 Code Torrent

Because of the high delays associated with Car Torrent, another method, called Code Torrent, was developed by Dr. Gerla's research group. Unlike the multiple hops used by Car Torrent, Code Torrent approaches data dissemination from a single-hop perspective by only allowing peers to share with their immediate neighbors, thereby eliminating the need for a routing mechanism [55]. In the Code Torrent scheme, participating nodes broadcast information to their immediate neighbors regarding the files they have and can share. All nodes listen promiscuously to the broadcasts sent by their neighbors. If a node receives information that one of its neighbors has a file that is of particular interest to it, then it will broadcast a request for that file to all of its neighbors. Any neighbor node that receives the request and that has all or part of the requested file to share then responds with a coded frame containing parts of the requested file. The interested node will continue to request coded frames from its neighbors until it has enough linearly independent pieces to recover the entire file. The authors refer to the percentage of nodes interested in downloading the file as the file's popularity. They used a 40% popularity in their study.

Code Torrent uses network coding to construct the coded frames. Each coded frame contains a random selection of various parts of the requested file and the encoding vector that the sender used to encode it, as well as the file identification and transaction identification. Dr. Gerla's group compares the performance of Code Torrent to Car Torrent in [55]. Their results show that as the number of nodes increases, like in a metropolitan area, Car Torrent performance gradually degrades because of the gossip messages and the underlying routing protocol. On the other hand, Code Torrent delay to obtain files decreases as mobility and congestion increase. However, the number of vehicles classified as interested in downloading the file in the Code Torrent study was only a maximum of 80 vehicles in a 5.76 square kilometer area, which is roughly 14 vehicles per square kilometer. Code Torrent performs better in congestion situations than

Car Torrent does, but still suffers from all nodes attempting to access the medium. This results in nodes waiting until the medium is idle for several slot times before broadcasting. The other condition that occurs is the hidden terminal problem, which results in increased transmission collisions. We anticipate that there is a vehicle density at which Code Torrent will no longer pass files effectively.

CHAPTER 6

NEW METHODS OF DISTRIBUTION

While the original intent of Code Torrent was for sharing files that only a subset of the network was interested in downloading, the method may also work very well for files such as the CRL that every node in the VANET wants to download. The CRL will have a popularity of 100% since every node will want the most recent CRL to protect them from malicious users and malfunctioning equipment, as well as to increase the overall security and safety of the VANET. While Code Torrent makes improvements over other previous methods of V2V file sharing by using broadcast mode to avoid routing difficulties and network coding to mitigate the piece problem, it still suffers under heavy load. This is due to the method having every OBU broadcast requests and relevant coded frames. Normal traffic on a bi-directional multi-lane highway can exceed 100 vehicles in a single 300-meter DSRC radio range, which is roughly 300 vehicles per square kilometer [71]. This normal traffic condition is much greater than the 14 vehicles per square kilometer simulated in the Code Torrent study. To reduce the contention for the wireless channel, we developed two methods to attempt to reduce the number of OBUs contending for the channel, Most Pieces Broadcast and Generation per Channel. Reducing the number of OBUs contending for the channel will increase the number of CRL pieces successfully received by the OBUs in the VANET [56]. This research combines aspects of the approaches from EPFL, Laberteaux, Haas and Hu, and UCLA to develop a better solution for distributing CRLs in a VANET environment.

6.1 Most Pieces Broadcast

The best situation for reducing contention for the wireless medium is to limit the number of broadcasting nodes to a single node. At this point there is no contention for

the channel, so the throughput will be the highest possible within the constraints of the channel capacity. The Most Pieces Broadcast (MPB) method, introduced in [56], creates a situation where only the node with the most number of CRL file pieces is selected to broadcast within a given radio broadcast range. The hidden terminal problem still exists for those OBUs that are within radio range of more than one selected node, so collisions still occur, but contention for the channel is reduced significantly. MPB will work in both V2I and V2V conditions with or without the presence of RSUs. Using MPB, RSUs will be selected as the node with the most pieces, since the RSU has every piece of the CRL file, resulting in an "RSU-only" broadcast scenario in the vicinity of the RSU. The RSUs are still needed to disseminate the CRL initially before V2V distribution can be used so that some vehicles have pieces to share.

MPB takes advantage of the CCH interval in DSRC to accomplish the process of selecting the node with the most CRL pieces. During the CCH interval, nodes exchange beacon packets. MPB adds information to the beacon packet to identify the CRL and the number of pieces that the node possesses. Each CRL is uniquely identified by the pair of fields containing the CA identifier and the CRL serial. Methods for reducing the number of bytes required for the CA identifier and CRL serial, such as hashing or using only the lower two bytes of each field, could be used to reduce the number of bytes added to the beacons. The number of pieces can be represented with 2 bytes, allowing up to 65,535 pieces to be accounted for. This requires a total of 6 bytes added to the beacon to advertise the number of CRL pieces possessed by each node for a specific CRL.

This piece-count information is used by receiving OBUs to increment a counter that contains the number of OBUs that possess more pieces than the OBU receiving the beacon. OBUs reset the counter that keeps track of other nodes that have more pieces to zero at the beginning of every CCH interval. RSUs also send out a beacon that includes the piece-count. If an RSU is within radio range of the OBU, the RSU will always increment the OBU counter since the RSU will always have the complete CRL. If the

OBU counter is zero, that means the OBU has the most number of pieces within its listening range, so it will become the broadcaster during the following SCH interval. During the SCH interval, only OBUs that have been "selected" will broadcast CRL pieces. If an OBU counter is greater than zero, this means that other OBUs within its radio range have more CRL pieces, so it will remain silent and listen for broadcast pieces during the SCH interval. Listening OBUs will store any new pieces they receive during the SCH interval and update their piece-count to send out with their next beacon on the following CCH interval.

If no pieces are received for a time period equal to the wait time calculated in (7), the silent OBU will begin to broadcast. This occurs when an OBU that incremented the waiting OBU's counter is listening to another OBU in its range.

$cwMin$	The $cwMin$ value from Table 1, measured in slots
$AIFS$	The AIFS value from Table 1, measured in slots
T_{Slot}	Slot time, 16 microseconds
Ctr_{CCH}	Counter value from the CCH Interval of nodes with more CRL pieces
T_{Packet}	Time to transmit the packet at the given data rate
Len	Length of packet in bytes
$Data Rate$	Data Rate of transmitter in bits per second

$$Wait Time = 2 \times (cwMin + AIFS) \times T_{Slot} \times Ctr_{CCH} + T_{Packet} \quad (7)$$

$$T_{Packet} = \frac{(Len_{Hdr} + Len_{Piece} + Len_{Tracker}) \times 8}{Data Rate} \quad (8)$$

Based on using the "Best Effort" access class, the wait time is 576 microseconds per counter value, plus 2,741.3 microseconds for T_{Packet} , based on a 22-byte WSMP header, a 1000-byte piece size, and 6 bytes of CRL piece-tracker information. This results in a 3,317.3 microsecond wait time for an OBU that has a counter value of one.

MPB is illustrated with the help of Figure 14. For this example, ACI 0, best effort, is used to calculate the CWmin and AIFS values from the previous paragraph as 576 microseconds. The numbers inside the diamonds designate the node number and the number of CRL file pieces possessed by that node. Node 1 receives beacons from nodes 3, 5, 6 and 7 during the CCH interval, all of which have more CRL pieces than node 1; therefore, node 1's counter is incremented to 4. Node 6 receives a beacon from node 7, so node 6 will not broadcast at the beginning of the SCH interval. Node 5 receives a beacon from node 6, so it will not broadcast at the beginning of the SCH interval. During the SCH interval, node 6 will receive pieces from node 7 so node 6 will remain silent for the duration of the SCH interval. Node 5 will not receive any pieces since node 6, as well as nodes 1 and 3, is silent. Therefore, node 5 will wait 576 microseconds plus the time required to transmit a packet and then begin to broadcast pieces. Nodes 6 and 1 have the opportunity to receive pieces from both node 7 and node 5 at this point, although the possibility of receiving many collisions also exists.

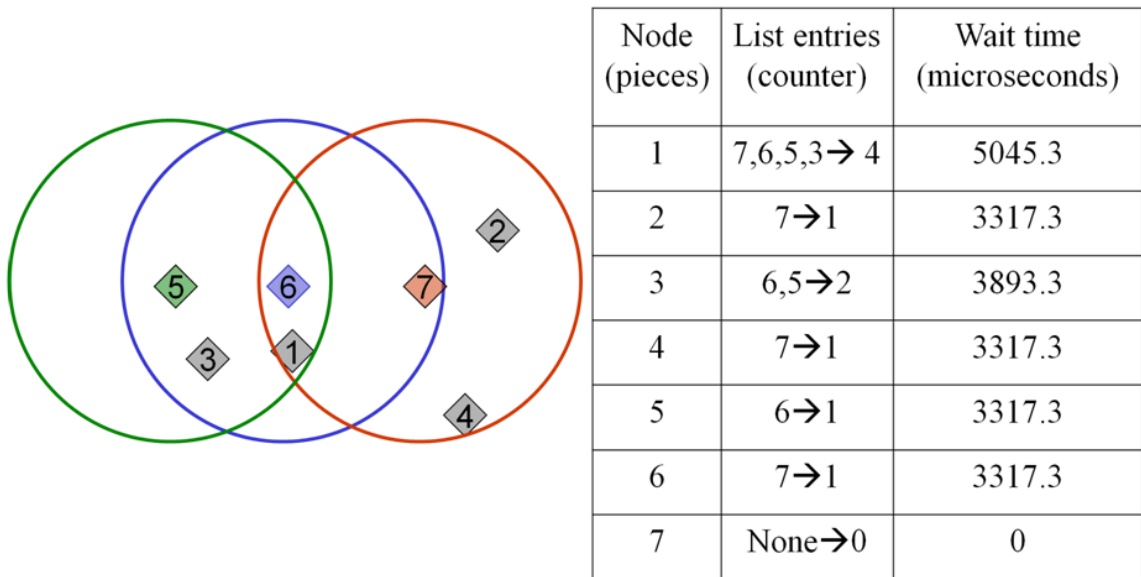


Figure 14. Example of most pieces broadcast.

The basic MPB algorithm for CRL distribution is as follows:

1. broadcast a beacon during the CCH interval with the CRL file description and number of pieces possessed;
2. keep a count of other nodes that possess more pieces than self (if an RSU is present it will always be the node with the most pieces);
3. wait for a fraction of the SCH interval equal to 576 microseconds times the count of nodes with more pieces plus the time required to transmit a packet;
4. if no pieces are received, then broadcast pieces for remainder of SCH interval, or else remain silent for remainder of SCH interval; //node is receiving pieces
5. return to 1

6.2 Generation per Channel

To allow for more nodes to participate by broadcasting their pieces, while still reducing the overall number of nodes contending for the medium, we developed a method that takes advantage of the multiple DSRC service channels. Generation per Channel uses multiple service channels to distribute CRL pieces in a fashion similar to network coding generations, where the CRL is split into multiple independent parts. Additional service channels are used to accomplish sending out these independent generations on different service channels. Two, four, and six service channels were used in this research. Announcements during the CCH interval inform nodes about which generations are available on which SCHs. This method has the benefits of pure Code Torrent, but the number of nodes contending for the medium is divided by the number of SCHs used. GPC could also use the MPB method on multiple SCHs, but this was not analyzed in this research. This would require each node to determine which service channel they will use in the following SCH interval and include that information in the beacon. Only the counter for the SCH the OBU will use in the following SCH interval would be incremented in this case.

An added benefit of this method is that the multiple CRL parts can be used independently, allowing for the use of a partial CRL once a generation is fully recovered

on a single SCH. A drawback of this method is that nodes must spend some time on multiple SCH to receive the entire CRL.

Two variations of GPC were evaluated. The first method has each OBU randomly select a different service channel each SCH interval. This means that each OBU downloads CRL generation pieces from every channel with equal probability, resulting in download completion for each generation at approximately the same time, on average. The second method has each OBU stay on a particular service channel each SCH interval until the entire CRL generation is downloaded. This results in staggered CRL generation downloads, but the overall download time is expected to be similar to the random channel selection method. The "stay on a channel" method has the advantage of being able to use a partial CRL if the CRL is encoded to allow a generation to be used independently of other generations. In both cases, the CCH interval and SCH interval are still alternately used as in MPB. The basic GPC algorithm for CRL distribution is as follows:

1. the CRL is split into two to six generations (each generation is a different part of the file);
2. service announcements are broadcast on the CCH to make nodes aware of which generation is on which channel;
3. each generation is distributed on a different SCH;
4. nodes go to different SCHs to get the parts of the file (could use either Code Torrent or MPB)

6.3 CRL Distribution Flow Chart

The flow chart in Figure 15 shows the steps needed for the methods introduced in this chapter. These steps were all examined closely during the model verification phase of the study. See section 7.6.1 for more details of this process.

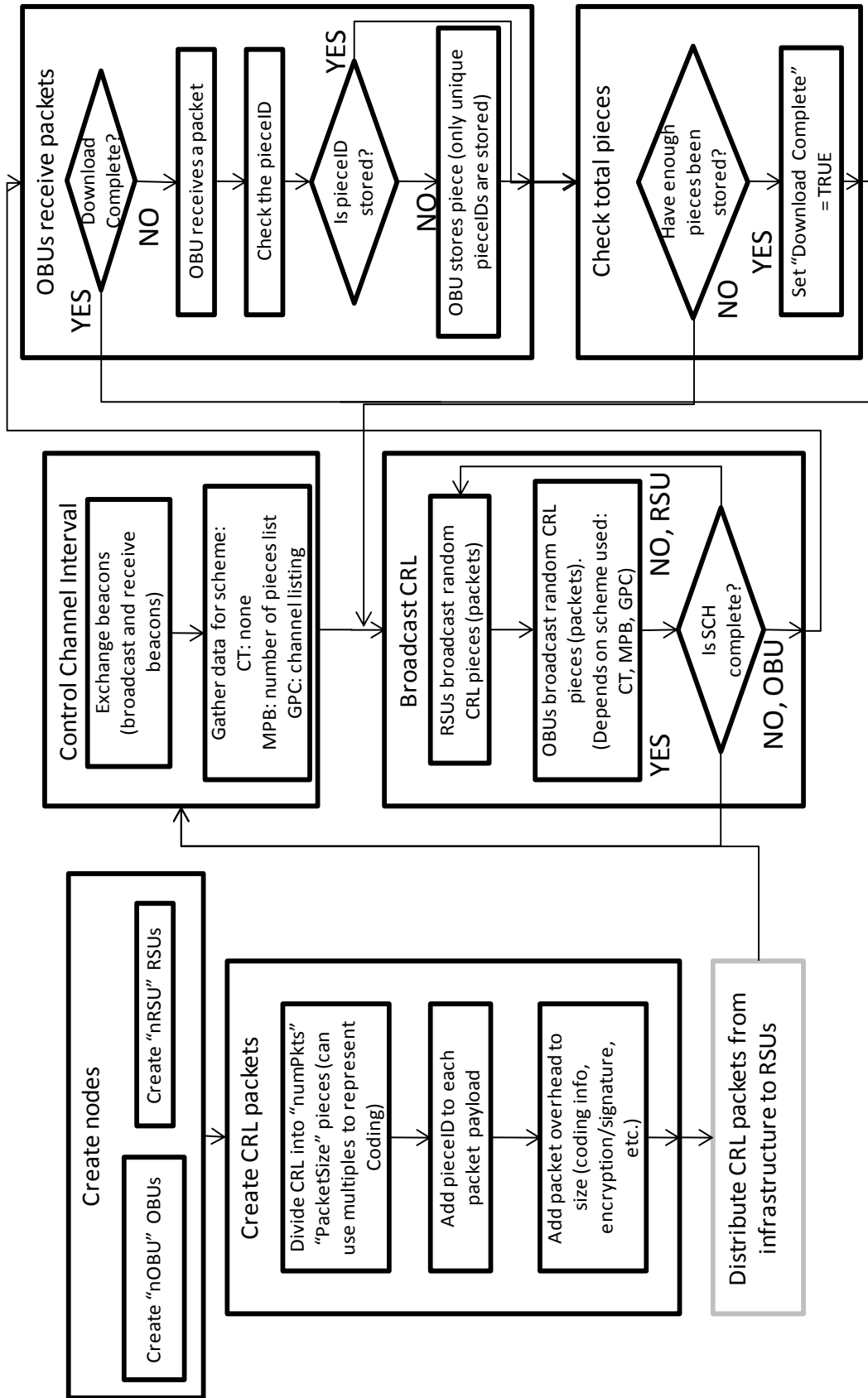


Figure 15. Flowchart for the simulation models.

CHAPTER 7

SIMULATION METHOD

7.1 Introduction

We compared the CRL distribution methods developed in Chapter 6, Most Pieces Broadcast and Generation per Channel, to the existing CRL distribution methods, RSU-only and Code Torrent. We created simulation models using the ns-3 network simulator [72]. The existing methods represent the extreme cases of having no OBUs broadcast in the RSU-only method and every OBU broadcast in the Code Torrent method. The primary comparison criterion was the number of OBUs that successfully downloaded the CRL. Other evaluation criteria used were the measure the number of packets transmitted and received during the distribution process.

7.2 ns-3

We chose to use the ns-3 network simulator [72] for this research. ns-3 is an open-source network simulator that was developed through NSF funding to update and modernize the ns-2 network simulator. Although the product is still in development, the software has many features that make it desirable to use, including the active development of new protocols and the use of a single programming language for model development and simulation control.

7.3 Assumptions

Several assumptions about the behavior of the nodes were made for this simulation. We followed the behaviors and specifications from the IEEE 802.11p and 1609 standards for the RSUs and OBUs. RSUs and OBUs have the same characteristics in the simulation except that RSUs are stationary and OBUs are mobile. The standard does require that all uses of DSRC be compatible with OBUs that have a single

transmitter/receiver unit. This study follows that requirement in that each OBU can only monitor one DSRC channel at any given time, either the control channel or one of the six service channels. This does not apply to RSUs, which can have multiple transmitter/receiver units installed, allowing them to send and receive on all service channels during the same SCH interval.

7.4 Common Simulation Settings

Two different mobility scenarios were used in this research, a random mobility model and a simulated vehicle trace. Simulation settings common to both mobility scenarios include the communication parameters for the infrastructure and mobile nodes and the CRL file properties. The following sections explain the common simulation settings used for both mobility models.

Table 5. Common simulation settings.

Parameter	Value
EnergyDetectionThreshold	-96.0
CcaModelThreshold	-99.0
TxGain	4.5
RxGain	4.5
TxPowerLevels	1
TxPowerEnd	16
TxPowerStart	16
RxNoiseFigure	4
CRL file size	1,000,000 bytes
Piece tracking overhead	6 bytes
Beacon size	100 bytes
Coding overhead	5%
CCH interval, SCH interval	50 milliseconds
Guard interval	4 milliseconds
Broadcast data rate	3 megabits per second
Channel rate	3 megabits per second
Slot time	16 microseconds
SCH cwMin for AC 0	15 slot times, 240 microseconds
SCH AIFSN for AC 0	3 time slots, 48 microseconds
Delay time per node with more pieces	$2*(cwMin+AIFSN)*Slot\ time$ 576 microseconds + packet transmit time

7.4.1 Physical and MAC Layer Settings

The simulation uses the physical and MAC layer parameters from 802.11p. All nodes alternate between the control channel and a service channel every 50 milliseconds in synchronization. Table 5 summarizes the physical and MAC layer parameters used for the simulation. RSUs and OBUs have the same physical and MAC layer settings.

The radio broadcast range was set using parameters in ns-3 so that a maximum reception range of about 300 meters was achieved. Initial values were taken from research done by Schmidt-Eisenlohr, et al. [73, 74] for work they did in ns-2. We ran several preliminary experiments to refine these values for use in ns-3.

7.4.2 Infrastructure (RSUs)

RSUs were designed as stationary, wireless nodes. Each RSU starts the simulation with the complete CRL, which means that every piece of the CRL is possessed by the RSUs. This includes the RSUs possessing every complete generation of the CRL file during the Generation per Channel cases. Also, in the GPC cases, RSUs can broadcast CRL file pieces on all service channels during the same SCH interval, effectively broadcasting a different generation on each service channel.

7.4.3 Vehicles (OBUs)

OBUs were designed in the model as mobile, wireless nodes. OBUs are given a mobility model dependent on the scenario being evaluated. All OBUs start the simulation with the zero CRL file pieces.

7.4.4 Replications and Pseudo-random Numbers

Care was taken to use pseudo-random numbers correctly in this simulation study. For every model, the same random number sub-stream was used for each common replication. This allows us to use certain statistical analysis tools for correlated output.

The sub-stream value was set at run-time for each replication using the following command,

```
NS_GLOBAL_VALUE="RngRun=$RngRunValue" ./waf --run  
"scratch/sim_model_name",
```

where "\$RngRunValue" is the sub-stream value used for that specific replication.

We ran several replications of each model, synchronizing the sub-stream values so that the first replication of each model used sub-stream value 1, the second replication of each model used sub-stream value 2, etc.

7.4.5 CRL File Parameters

A one megabyte CRL was distributed using a fixed piece size per scenario. Piece sizes correspond to packet payload size in the simulation. Piece sizes of 500, 1000, and 1500 bytes were evaluated. For example, with 500-byte pieces, there are a total of 2000 file pieces in the original file. A coding rate of 200% was used, generating 4000 pieces from the CA. A coding overhead of 5% was used, so each OBU must receive 2100 unique file pieces to download the complete CRL. CRL pieces are tracked by a piece number added to the packet payload. A total of 6 bytes is added as overhead to each packet payload to represent the CA identifier, CRL serial, and the piece number.

7.5 Mobility Models

Härri, Filali and Bonnet [75] provide a taxonomy and a detailed summary of existing mobility and network simulator projects. Two different mobility models were used in this study. The first is based on random, two-dimensional motion. The second is based on realistic vehicular mobility traces. While random motion of nodes does not provide accurate vehicle movement, it does simulate the interaction of nodes and the basic effectiveness of the CRL distribution methods. The random-mobility model is used for verification and validation of the model and to establish parameters for the second mobility model. Due to the high number of OBUs in the trace files and the length of time

to simulate these models, we decided to limit the number of scenarios to only the most competitive set of parameters for each of the two methods we developed. Once the most competitive levels were determined for the various experimental factors based on the random mobility models, we used realistic simulated vehicle traces to compare the effectiveness of the four different methods: RSU-only, Code Torrent, Most Pieces Broadcast, and Generation per Channel.

7.5.1 Random Mobility, aka "Bouncing Box"

Two different random mobility scenarios were examined for the purpose of verification and validation of the model, one with five RSUs, each 450 meters apart with overlapping radio range, and one with two RSUs 1650 meters apart. The boundaries are at 0 and 2250 on the x-axis, and 0 and 500 on the y-axis. The positions of the RSUs are the same for every replication. The OBUs starting position is randomly assigned inside the boundaries for each replication. A representative starting location set for both the two-RSU case and the five-RSU is shown in Figure 16. The ns-3

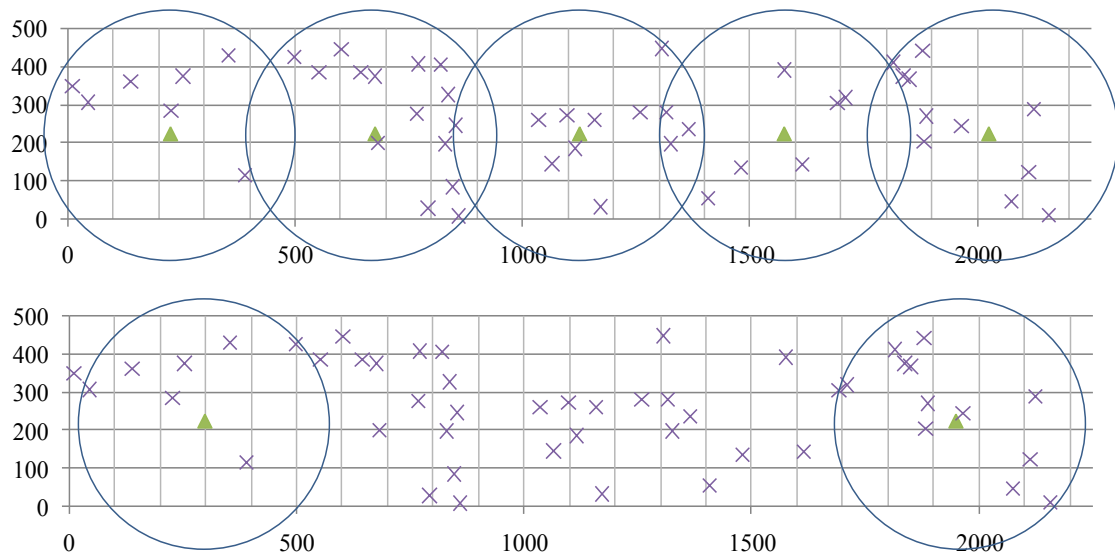


Figure 16. Bouncing box RSU and OBU start positions with 50 OBUs. The circles represent a 300 meter radio-range radius. Top diagram shows scenario with 5 RSUs, bottom diagram shows scenario with 2 RSUs.

"RandomDirection2dMobilityModel" is used so that OBUs move in a straight line until they hit the boundary of the "box" at which point they pick a new heading at random and travel in a straight line until they hit the boundary again. We refer to this mobility model as the "bouncing box." This movement continues for the entire simulation. Each OBU in the bouncing box had a constant velocity of 26 meters per second, which is roughly 60 miles per hour. RSUs start the simulation with every CRL piece while OBUs start the simulation with zero CRL pieces. These scenarios represent different infrastructure density levels. The five-RSU scenario should result in only the RSUs broadcasting when using MPB as explained in chapter 6.1. The same total area was used for both scenarios.

7.5.2 Simulated Vehicle Traces

The second mobility model uses more accurate vehicle movements to study the effectiveness of the CRL distribution methods. Vehicle traces produced by a microscopic vehicular traffic simulator were used to control the OBU mobility. We used a set of traces based on road maps in Switzerland [67], first used by Naumov, Baumann and Gross [68, 76], and later by Laberteaux, Hu and Haas in their "epidemic" paper [50]. Each trace is 300 seconds in duration. The traces are formatted for use with the ns-2 network simulator, so we adapted a program to read the mobility trace files so that the trace files are usable with ns-3. More information about the traces and the trace reader is found in Appendix A.

Table 6. Vehicle trace parameters.
Length in meters, velocity in meters per second.

Region	Number of OBUs	Length, X	Length, Y	Min Velocity	Max Velocity	Average Velocity
Enge-Oberstrass (medium)	520	4000	7000	8.62	32.61	19.75
Enge-Oberstrass (high)	1705	4000	7000	3.3	32.62	18.09
Hurgen-Jona (high)	1275	8500	4500	2.44	32.44	13.68

Figure 17 shows the mobility traces of different vehicle densities in both city and highway scenarios. Two different traces were used, one of a city environment, Enge-Oberstrass, and one of a mixed highway and city environment, Hurgen-Jona. Both of these traces were used in the simulation experiments to compare the four CRL distribution methods (RSU-Only, Code Torrent, Most Pieces Broadcast, and Generation per Channel). The number of nodes and velocity information is provided in Table 6 for these traces.

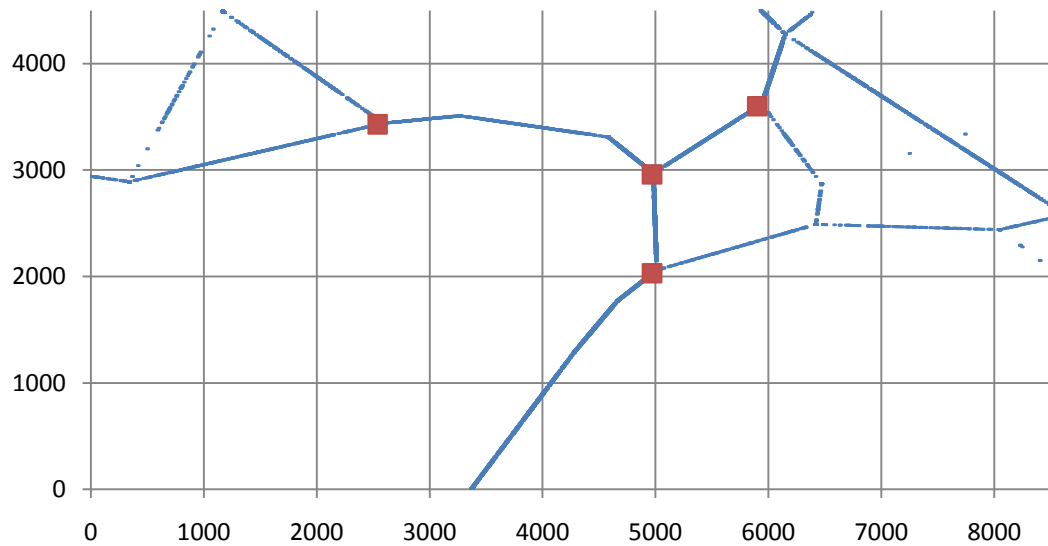
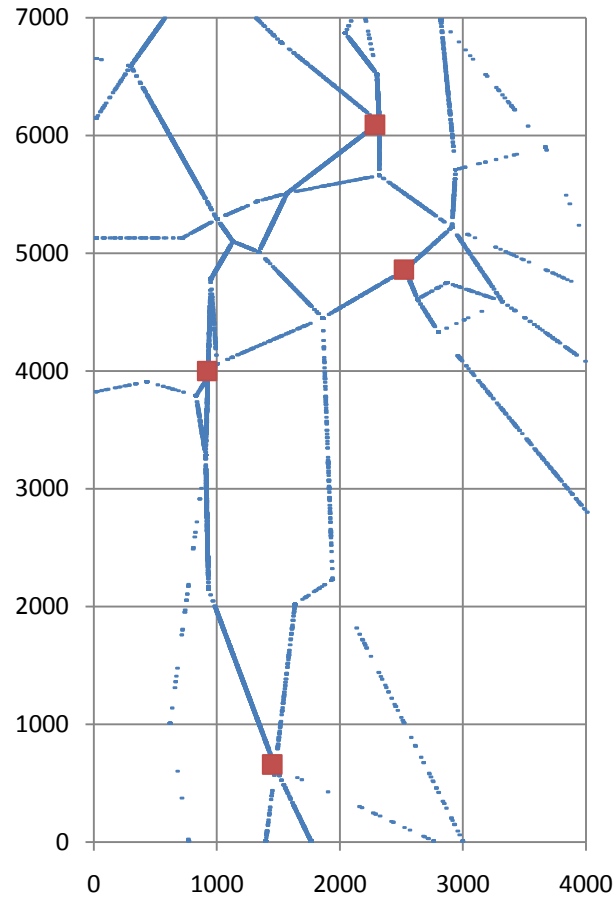


Figure 17. Position traces from ns-3 output for the Switzerland maps.
 The squares are the locations of RSUs used in the simulation. Distances are in meters.
 Top trace, City, Enge-Oberstrass, 520 and 1705 OBUs.
 Bottom trace, Highway/City, Hurgens-Jona, 1250 OBUs

7.5.3 Comparison of Mobility Models

There are several differences between the bouncing box model and the simulated vehicle trace. In the trace, the vehicles are confined to the roads, so while there are fewer OBUs per total area, the OBUs are concentrated in very small areas. This results in an even higher OBU density than in the bouncing box model. Another significant difference is the amount of contact with neighbors. In the bouncing box, contact was usually of short duration since the nodes were moving in random directions at 26 meters per second, but two nodes may come into contact several times throughout the simulation due to the confined area. In the trace models, since vehicles are confined to the roads, contact time between vehicles moving in the same direction will be greater than in the bouncing box model; however, once contact is broken between neighbors, it is unlikely those two neighbors will come into contact again. These contact patterns were not examined in this study.

In the bouncing box model, the simulation was run until every node downloaded the complete CRL, so completion times are used as the primary criterion. In the trace mobility models, since they were of a fixed-time duration of 300 seconds, the total number of OBUs downloading the complete CRL is used as the primary criterion for evaluation for the CRL distribution methods.

7.6 Verification and Validation

Verification and validation is the process of making sure the simulation is working correctly, both logically and realistically. The random mobility models were used to verify that the model behaved correctly and to validate the results based on other researcher's results.

7.6.1 Verification

Model verification deals with the logic of the model -- "building the model *right* [77]." Extensive verification of the model was done throughout writing the code for the simulation. Debug statements and examination of the output were the main form of verification. This process was completed for all four methods (RSU-only, Code Torrent, Most Pieces Broadcast, and Generation per Channel). Each method was run for five replications to ensure that independent runs were being generated through the use of increasing the random number sub-stream for each replication. Functionality of node generation, piece generation, channel switching, OBU mobility, piece downloading, and piece broadcast were all verified.

7.6.2 Validation

Model validation deals with how realistically the model portrays the system under study -- "building the *right* model [77]." This means that the simulation model correctly produces results that are similar to the results of the known system. In our case, there is no existing system in place to compare the model other than the results published in [55] for the results of a Code Torrent simulation. These results are shown below in Figure 18. Their test cases were for a file popularity of 40%, which means that only 40% of the nodes in their network were actively downloading the file. We simulated our model with 40, 60, and 80 OBUs with a velocity of 20 meters per second and 26 meters per second so we could compare the results. While the mobility models were different in each study,

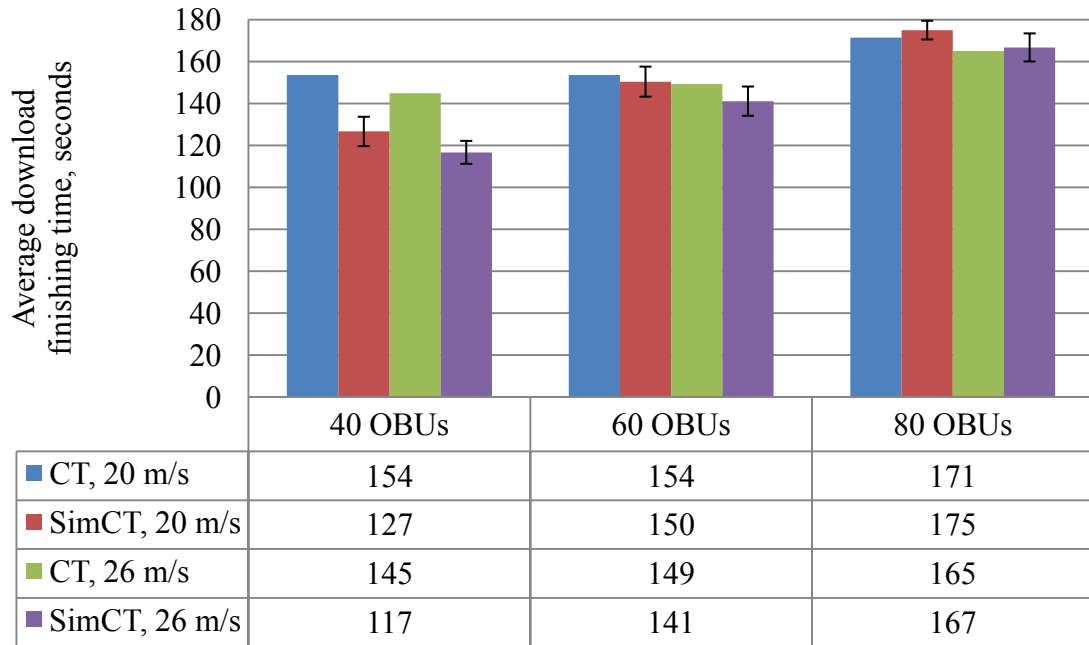


Figure 18: Comparison of simulation model to Code Torrent output.

there should be at least an order of magnitude similarity between the models. Also, the Code Torrent simulation in [55] did not use alternating control channels and service channels. To account for this difference, we doubled the download times from the Code Torrent results in [55] to compensate for our 50% duty cycle.

The simulation developed for this study does show similar values compared to the results in [55] for 60 and 80 OBUs. The error bars shown in Figure 18 are the 95% confidence interval for the average value of the download time, clearly showing that the Code Torrent values fall within the range of the confidence interval.

7.7 Design of Experiments

Using the definitions in [77], a *factor* is a parameter or variable that may affect the outcome of a model. Each factor can have several different values, or *levels*, at which it can be evaluated. In this study, four different factors were explored to determine the most effective levels for the factors of MPB and GPC. These factors are piece selection method, piece size, GPC channel selection method, and number of GPC channels, shown

Table 7. Experiment factors.

Factor name	Levels
Piece Selection	random, random permutation
Packet Size	500, 1000, 1500
GPC Channel Selection	Random, Stay on
Number of GPC Channels	2, 4, 6

in Table 7 with the different levels we explored. The four factors were evaluated at the different levels while holding other factors constant to generate four different experiments using the bouncing box mobility model with two RSUs. Several OBU densities were used to examine whether the factor was sensitive to vehicle density. Each case was run with 10, 25, 50, 100, 250, and 500 OBUs. The level for each factor that resulted in the lowest time for all OBUs to complete the CRL download was selected as the level to use in the trace mobility simulation runs. This level is referred to as the *most effective level*.

7.7.1 Randomness of Piece Selection

The first experiment conducted helped to determine the method for selecting which CRL file piece to broadcast. This factor applied to all four distribution methods, and both RSUs and OBUs. During each SCH interval, nodes broadcast CRL file pieces based on the method of CRL distribution. The selection of which piece to broadcast was accomplished using two different methods, a random selection method and a type of random permutation where a piece is not selected again until all other pieces have been selected. We anticipated that the random permutation method will result in lower download times since more worthwhile pieces are broadcast more frequently. OBUs have far fewer pieces to select for broadcast, so the methods where every OBU broadcasts pieces is expected to have higher download times.

7.7.2 Piece Size

The second experiment evaluated three different piece sizes for all nodes to use. The size of the pieces was also the size of the payload of the packets used in the simulation. Piece sizes of 500, 1000, and 1500 bytes were tested. While smaller pieces require less time to broadcast than larger pieces, there is a constant packet overhead that reduces the overall throughput when using small packets. However, larger packets are more susceptible to collisions, noise, radio fading, and other factors that reduce the number of packets received.

While the piece size could have remained fixed at 500 bytes and multiple pieces sent in each packet, using larger pieces is effectively the same. Instead of sending 3 500-byte pieces in a single 1500-byte packet, a single 1500-byte piece was sent in a 1500-byte packet. The number of pieces needed is dependent on the piece size and the file size. We used a fixed file size of 1,000,000 bytes for the CRL. The number of pieces needed based on a coding overhead of 5% for different piece sizes is shown in Table 8. 4000 file pieces were generated by the CA regardless of the piece size used.

7.7.3 GPC Channel Selection

The third experiment examined the way the GPC method selects from which service channel to download pieces. The GPC method requires the OBUs to spend time on different service channels to download the generations of the CRL file. As discussed in chapter 6.2, GPC can either randomly select a service channel each interval, or stay on a particular service channel until the generation is fully downloaded. We anticipated that

Table 8. Number of pieces required for CRL download based on various piece sizes.

Piece Size	Number of Pieces Needed
500	2100
1000	1050
1500	700

the vehicle density will influence this level. When the number of vehicles is less dense, the random channel selection will enable a better distribution of pieces. However, as the vehicle density increases, more vehicles are available per channel with which to exchange pieces.

7.7.4 Number of GPC Channels

The next experiment determines how many service channels to use for the GPC method. The number of generations of the CRL file is equal to the number of service channels used. With six service channels available in DSRC, we evaluated the performance of two, four, and six service channels to determine which level downloaded the entire CRL in the least amount of time. There is a trade-off between reducing the number of nodes contending for the channel and having to spend time on more channels. The number of OBUs contending for the same service channel is reduced by a factor of the number of service channels used in this method. If RSUs are available to the OBU, it could potentially receive the entire generation during the time it takes to drive through the RSUs radio footprint if the OBU stayed on the same service channel.

7.7.5 Comparison of Methods Using the Trace Mobility Model

The culminating experiment compares the number of nodes that successfully download the CRL during a 300 second time period using three different trace mobility models. All four methods, RSU-only, Code Torrent, MPB and GPC, were simulated on the three different trace mobility models described in chapter 7.5.2. The levels from the first four experiments were used to reduce the number of combinations of factors and levels since each replication of the 1275 and 1705 OBU traces required in excess of 12 hours of computing time. Ten replications of each of the four methods for each of the three trace models were run for a total of 120 replications.

CHAPTER 8

SIMULATION RESULTS

The results of the simulation are discussed in this chapter, both for the bouncing box scenario and the three mobility trace scenarios.

8.1 Evaluation Criteria

8.1.1 CRL Download Time

For the bouncing box models, the simulation was run until all n OBUs completed the download of the CRL file. The time of download completion for each OBU was recorded. The time for the final OBU to complete is treated as the completion time for the simulation run. When compared to another method, lower completion times indicate that the method performed better than the other method.

For the trace mobility models, the simulation was run for 300 seconds of simulation clock time, which was the length of the mobility trace. The completion time of each OBU receiving the complete CRL was recorded. At the end of the 300 seconds the total number of OBUs completing the download was recorded. When compared to another method, a higher number of OBUs completing the CRL download indicate that the method performed better than the other method.

The times and number of OBUs complete are used as evaluation criteria. In each case, statistical analysis is required to determine if the differences between the methods are statistically significant.

8.1.2 Number of Pieces Transmitted and Received

The total number of pieces transmitted by the physical layer and received by the application layer during the service channel intervals by the RSUs and the OBUs was

recorded. The number of pieces received by each OBU until the OBU completed the CRL file download was also recorded. These data points are used to evaluate two different evaluation criteria, the Packet Delivery Ratio (PDR), and the Normalized Packet Overhead (NPO) [78].

PDR is defined as

$$\frac{\text{total number of pieces received during the simulation}}{\text{total number of pieces transmitted during the simulation}}. \quad (9)$$

A PDR value greater than one is possible in the case where a single broadcaster sends pieces to several listening nodes, as in the RSU-only model. Higher values of PDR indicate that the method is achieving a higher effective throughput on the medium. Very low numbers indicate a high number of dropped packets due to collisions in the medium, or it could indicate a low ratio of receivers to transmitters. The PDR value for the RSU-only scenario can be used as a rough estimate for vehicle density since the RSUs are the only broadcasters and the OBUs are the only receivers.

NPO is defined as

$$\frac{\text{number of CRL pieces downloaded}}{\text{number of CRL pieces needed to recover the file}}. \quad (10)$$

An NPO value of one indicates that every piece received was used to recover the CRL file. Numbers greater than one indicate that duplicate pieces were received. The NPO value is an average of all of the individual OBU pieces downloaded to receive the CRL.

8.2 Statistical Analysis

Statistical analysis was completed to compare the results of each method to the results of the other methods. This was accomplished using the methods described in [77, 79-81] for correlated sampling, or common random numbers. The same random number seed was used throughout all simulation runs, and the run number was advanced to use the next sub-stream of random numbers, per [82]. This synchronizes the random

numbers to reduce variance between different methods of CRL distribution. In particular, the mobility of the OBUs is the same for every method on each of the independent replications with the same run number. The paired-t comparison requires correlated sampling and independent replications.

Each experiment is run for several replications to find several samples (Y) for the experimental criteria, namely time or number of OBUs completed, and pieces received and transmitted. The next step is comparing each of the experiments to each other. Since correlated sampling is used, the difference between results for each method is found and then averaged, resulting in the average difference between the methods, \bar{D} (11), and the variance of the difference, S_D^2 (12). Here R is the total number of replications and r designates an individual replication.

$$\bar{D} = \frac{1}{R} \sum_{r=1}^R (Y_{r1} - Y_{r2}) \quad (11)$$

$$S_D^2 = \frac{1}{R-1} \sum_{r=1}^R (D_r - \bar{D})^2 \quad (12)$$

The null hypothesis, H_0 , is that the two means of the different methods are the same; thus the difference of the means would be zero, as shown in (13). The alternative hypothesis, H_a , where the means are different, indicates that difference between the methods is statistically significant at the specified confidence interval, $(1 - \alpha)$, shown in (14). A confidence interval of 95% was used throughout this study.

$$H_0: \bar{D} = 0, \quad (13)$$

$$H_a: \bar{D} \neq 0 \quad (14)$$

The number of samples and the sample variance of the two compared experiments are used to find the half-width of the difference between the means using (15). The value of 0.05 is used for α , resulting in a 95% confidence interval for the half-width. The number of samples (simulation runs) for each experiment (R) is used to determine the degrees of freedom for the student-t distribution.

$$half - width = t_{1-\frac{\alpha}{2}, R-1} \cdot \sqrt{\frac{S_D^2}{R}} \quad (15)$$

The half width is then used with the difference of the sample means to determine if the experiments have a significant statistical difference. If $\bar{D} \pm half - width$ contains zero, then the null hypothesis cannot be rejected, indicating that there is not sufficient evidence that the experiments produced different means, i.e., that they are not different.

8.3 Factor Experiment Results

The results from the experiments to determine the most effective level of the four factors are discussed in detail below. Table 9 summarizes the results of the best factor levels as determined by the earlier simulation results.

Table 9. Most effective levels per factor.

Factor name	Selected factor level
Piece Selection	Random permutation
Packet Size	1000
GPC Channel Selection	Random
Number of GPC Channels	6

8.3.1 Randomness of Piece Selection

The piece selection method that is most effective across the different vehicle densities is the random permutation method. While the methods perform essentially the same, there is some statistical difference in favor of the random permutation method. Specifically, the random permutation method outperforms the random method at the 500 OBU density level for both Code Torrent and MPB. The average download times are shown in Figure 19.

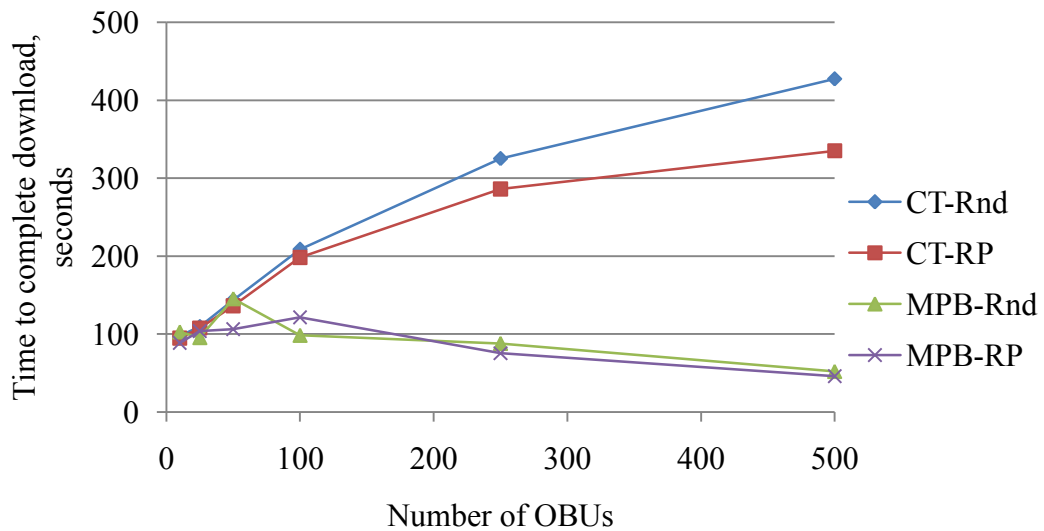


Figure 19. Effect of piece selection on download time. "Rnd" is the random selection method, "RP" is the random permutation selection method.

8.3.2 Piece Size

The piece size that is the most effective across the different vehicle densities is the 1000-byte piece size. While the difference between the 1000-byte piece and the other sizes is not statistically significant at every density, the 1000-byte piece size does perform at least as good as the other packet sizes at every density level for both Code Torrent and MPB except at the 500 OBU case of MPB, where the 500-byte piece size statistically outperforms the other piece sizes. The difference 500-1000 and 500-1500 at that level is only 2 seconds and 7 seconds respectively. The average download times are shown in Figure 20.

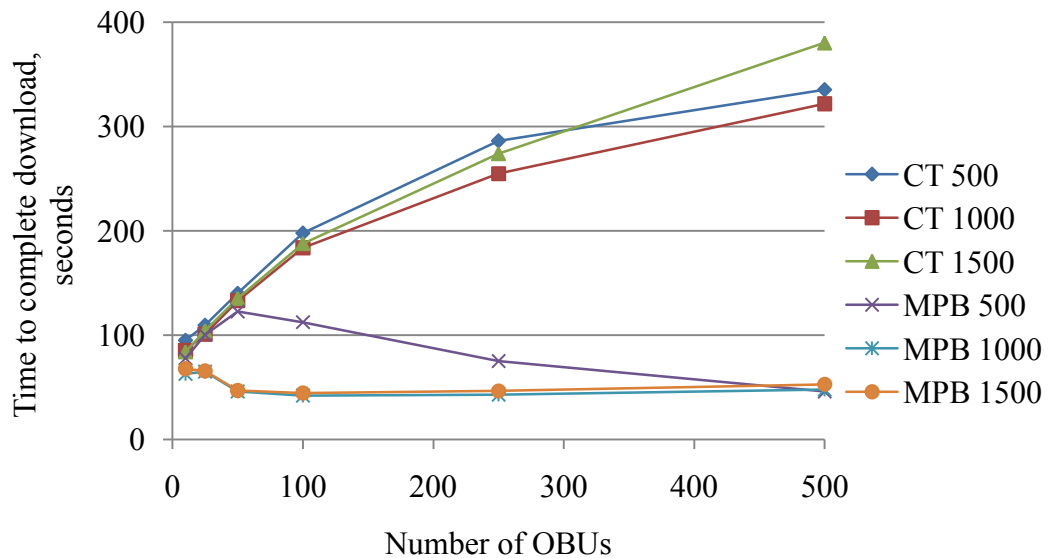


Figure 20. Effect of piece size on download time.

8.3.3 GPC Channel Selection

The GPC channel selection method that is most effective across the different vehicle densities is the random channel selection method. There is significant statistical difference in favor of the random channel selection method for all OBU densities except for the 10 and 25 OBU densities, where there is no statistical difference between the methods. The average download times are shown in Figure 21. There is no significant statistical difference in the PDR values for this factor

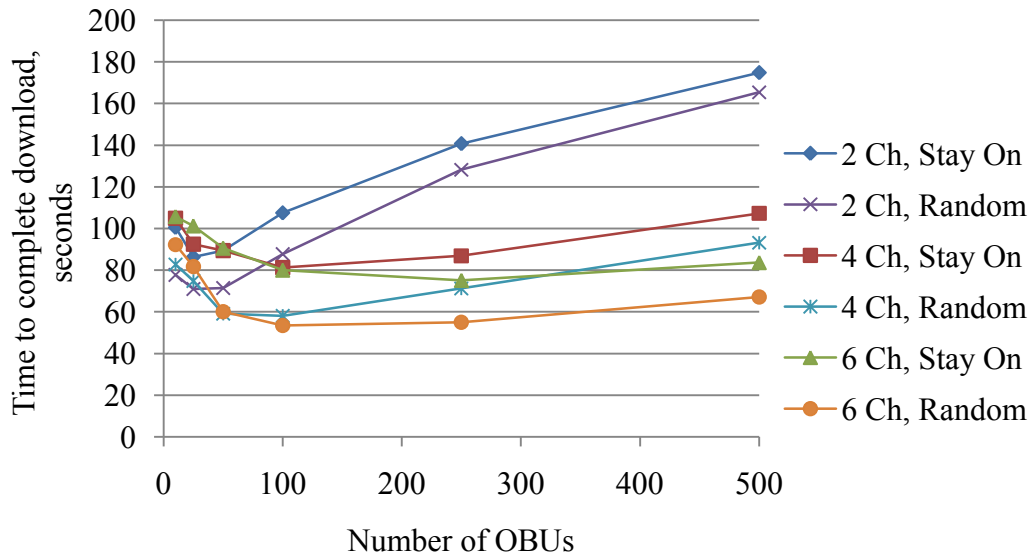


Figure 21. Effect of GPC channel selection method on download time.

8.3.4 Number of GPC Channels

The number of GPC channels that is most effective across the different vehicle densities is six. There is strong statistical difference in favor of the 6-channel model for all OBU densities except for the 10 and 25 OBU densities, where there is no significant difference between the methods. The average download times are shown in Figure 22 at the top. There is statistical difference in the PDR values for this factor. For lower OBU densities, the 2-channel model has the highest PDR, but as the OBU density increases above 250 OBUs, the 6-channel model has the highest PDR.

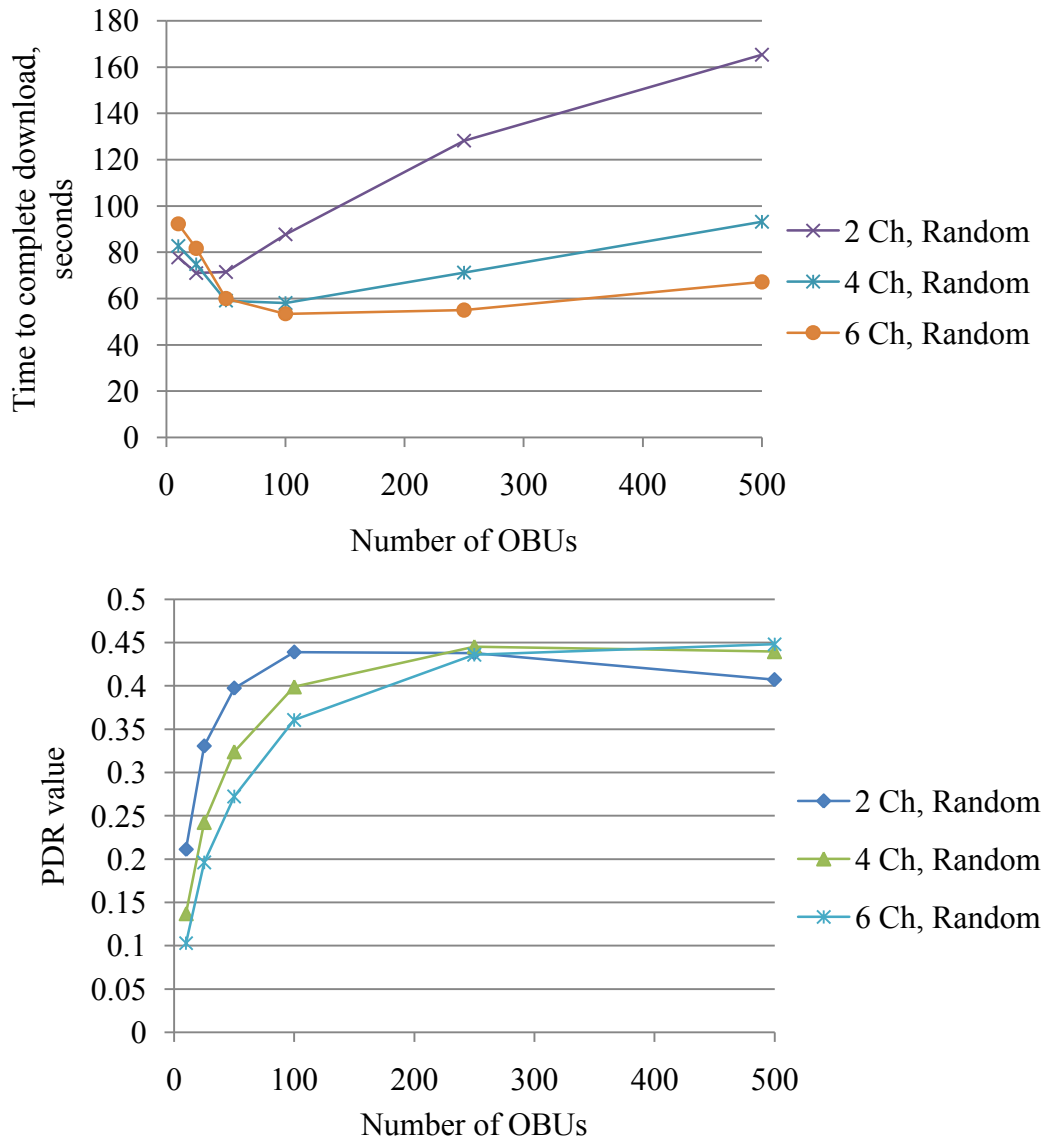


Figure 22. Effect of number of GPC channels on download time and PDR.

8.4 Trace Mobility Model Results

Using the most effective levels for the piece selection method, piece size, GPC channel selection, and the number of service channels for GPC, the four CRL distribution models were each simulated using three different simulated vehicle traces. The total number of OBUs receiving the complete CRL over the 300 second trace is shown, as is the packet delivery ratio (PDR) of pieces received divided by the pieces transmitted, and the normalized packet overhead (NPO) of number of pieces received divided by the number of pieces needed. Higher numbers are better for OBU count and PDR, lower numbers are better for NPO.

8.4.1 Results from the 520 OBU Network

The results from 10 replications using the trace file Enge-Oberstrass with four RSUs and 520 OBUs show that MPB distributes the CRL to the most number of OBUs, outperforming GPC6 by 18.7 ± 1.1 . MPB had the highest PDR aside from the RSU-only case and 26% fewer packet transmissions than Code Torrent and 78% fewer than GPC6. Figure 23, Figure 24, and Figure 25 show the results from this experiment.

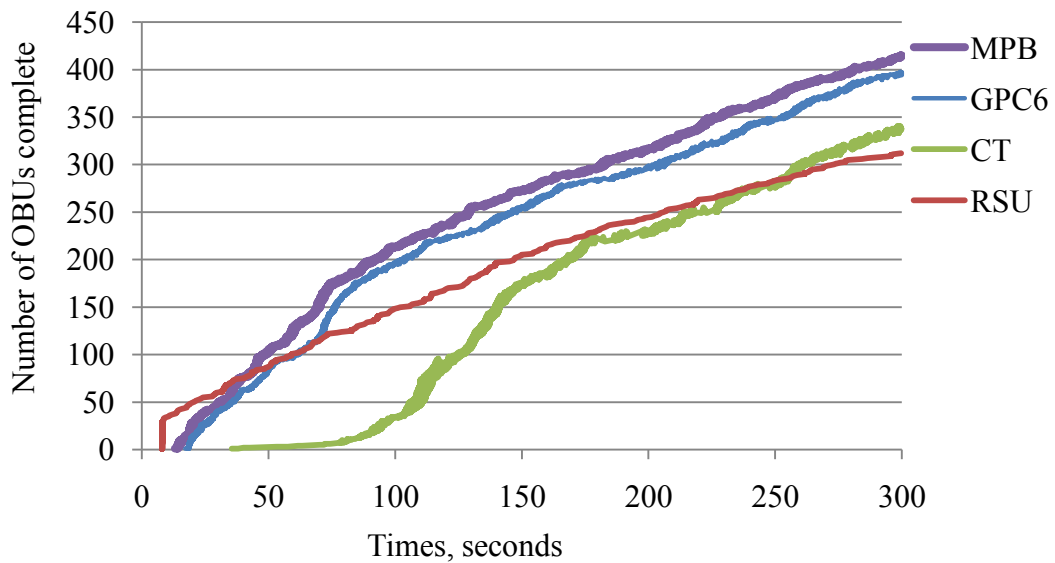


Figure 23. Completion times for OBUs from Enge-Oberstrass with 520 OBUs.

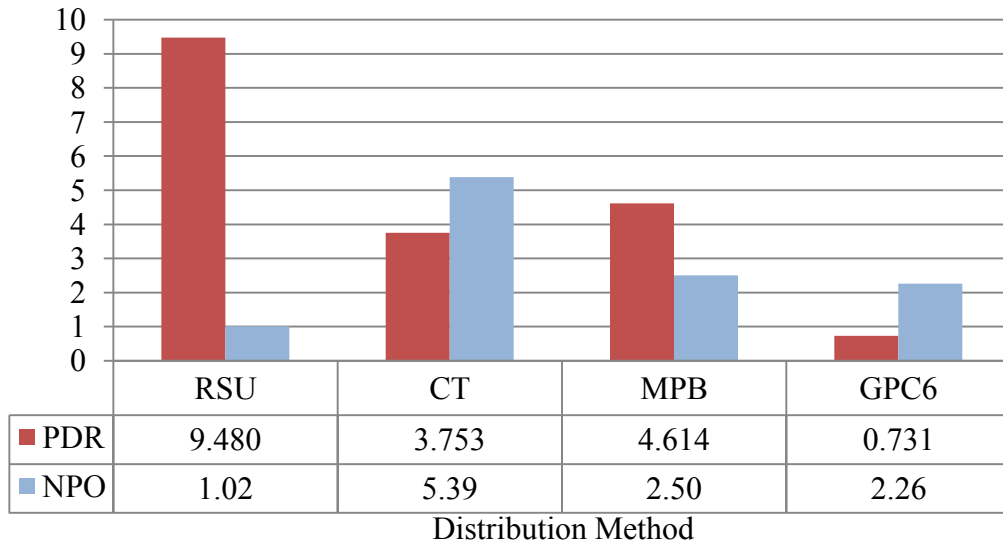


Figure 24. Packet deliver ratio and normalized packet overhead for OBUs from Enge-Oberstrass with 520 OBUs.

The number of OBUs that downloaded partial CRLs using the GPC6 method is 11.2 ± 1.6 . This means that on average another 11 OBUs would have access to some parts of the CRL that were downloaded as independent generations, offering some protection to those OBUs.

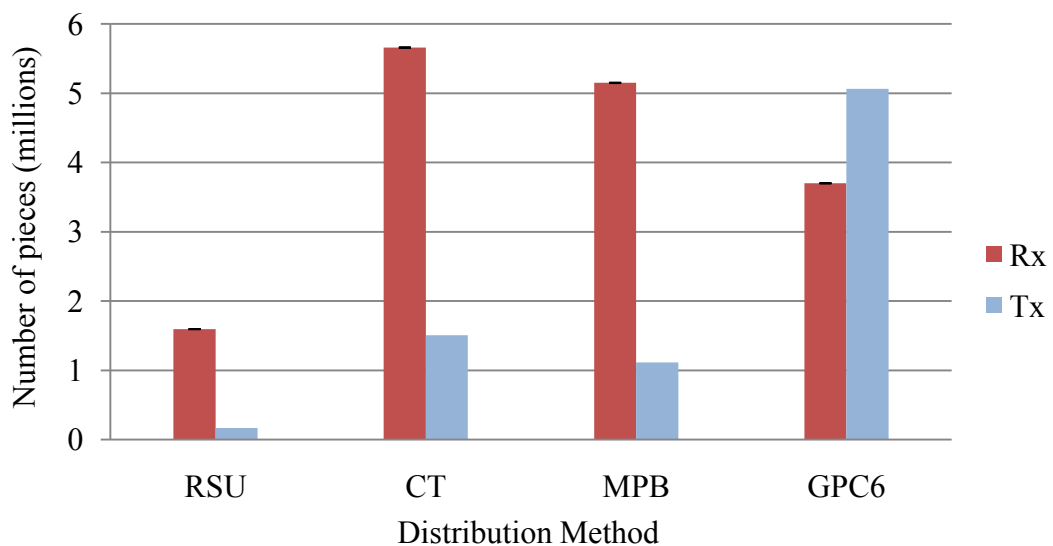


Figure 25. Number of transmitted and received pieces from Enge-Oberstrass with 520 OBUs.

8.4.2 Results from the 1705 OBU Network

The results from 10 replications using the trace file Enge-Oberstrass with four RSUs and 1705 OBUs show that MPB and GPC6 perform almost the same; however, MPB does distribute the CRL to 5.0 ± 1.9 more OBUs during the simulation. MPB also has the highest PDR aside from the RSU-only case. MPB had 32% fewer packet transmissions than Code Torrent and 80% fewer than GPC6. Figure 26, Figure 27, and Figure 28 show the results from this experiment. RSU-only outperformed CT by 199.9 ± 7.0 .

GPC6 and MPB continue to perform well at the increased vehicle density levels since the number of OBUs contending for the channel is reduced; however, Code Torrent shows that it does not scale well to higher vehicle densities, resulting in a much lower number of OBUs completing the CRL download. This occurs when OBUs that only have a small number of pieces contend for the channel with OBUs that have a large number of pieces. This has the effect of reducing the number of new pieces available for distribution during a service channel interval. The high Normalized Packet Overhead (NPO) value for CT shows quantitatively that multiple duplicate pieces are received

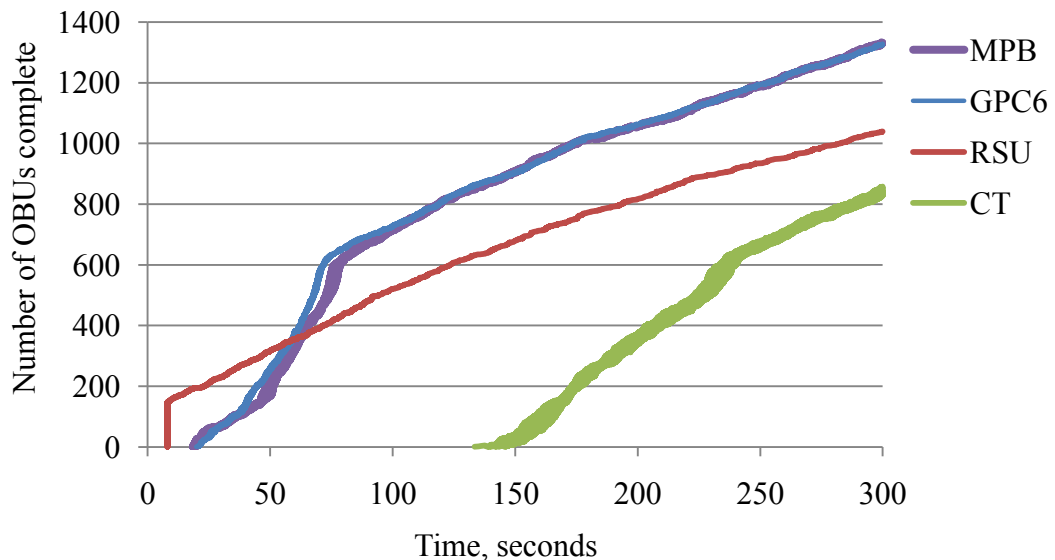


Figure 26. Completion times for OBUs from Enge-Oberstrass with 1705 OBUs.

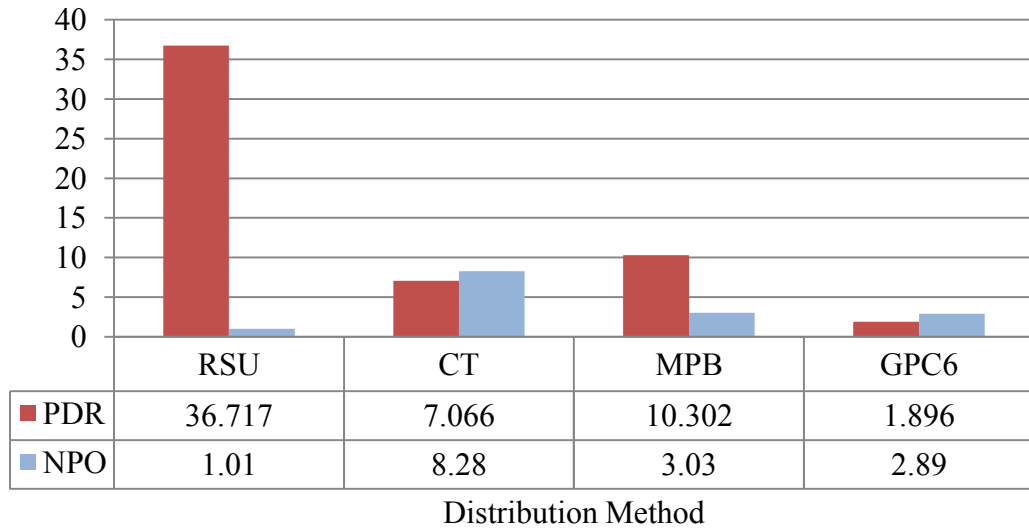


Figure 27. Packet deliver ratio and normalized packet overhead for OBUs from Enge-Oberstrass with 1705 OBUs.

during the CRL download process.

The number of OBUs that downloaded partial CRLs using the GPC6 method is 33.4 ± 4.7 . This means that on average another 33 OBUs would have access to some parts of the CRL that were downloaded as independent generations, offering some protection to those OBUs.

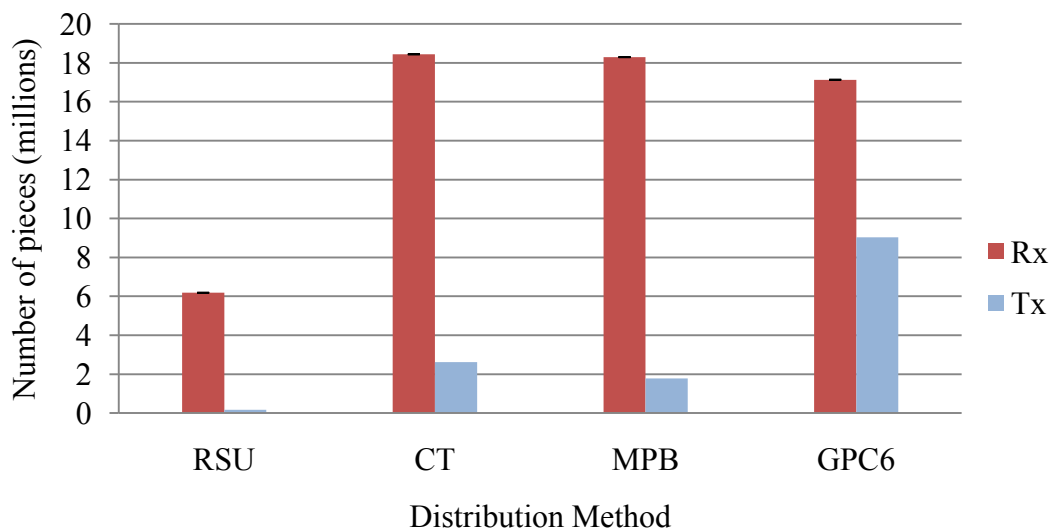


Figure 28. Number of transmitted and received pieces from Enge-Oberstrass with 1705 OBUs.

8.4.3 Results from the 1275 OBU Network

The results from 10 replications using the trace file Hurgen-Jona with four RSUs and 1275 OBUs show that MPB and GPC6 perform very similarly; however, MPB does distribute the CRL to 17.8 ± 1.8 more OBUs during the simulation. MPB also has the highest PDR aside from the RSU-only case. MPB had 33% fewer packet transmissions than Code Torrent and 83% fewer than GPC6. This is the first OBU density-level where Code Torrent performed worse than RSU-only, confirming our expectation that at a certain vehicle density Code Torrent would no longer perform well because of the high number of OBUs contending for the channel to participate in the piece exchange. At this vehicle density, RSU-only outperforms Code Torrent by a difference of 69.5 ± 5.9 OBUs receiving the CRL. Figure 29, Figure 30, and Figure 31 show the results from this experiment.

The number of OBUs that downloaded partial CRLs using the GPC method is 15.2 ± 3.2 . This means that on average another 15 OBUs would have access to some parts of the CRL that were downloaded as independent generations, offering some protection to those OBUs.

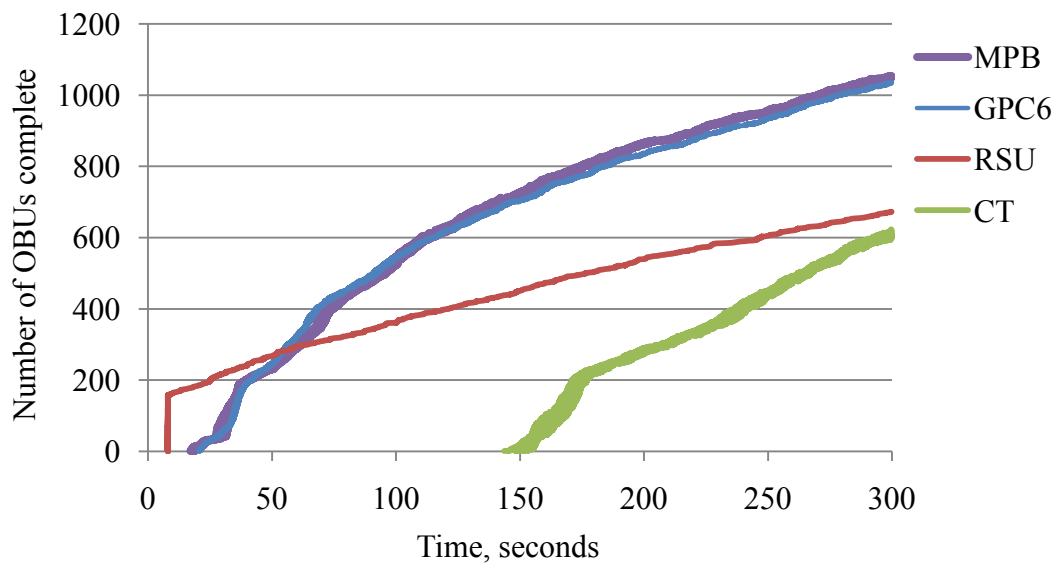


Figure 29. Completion times for OBUs from Hurgen-Jona with 1275 OBUs.

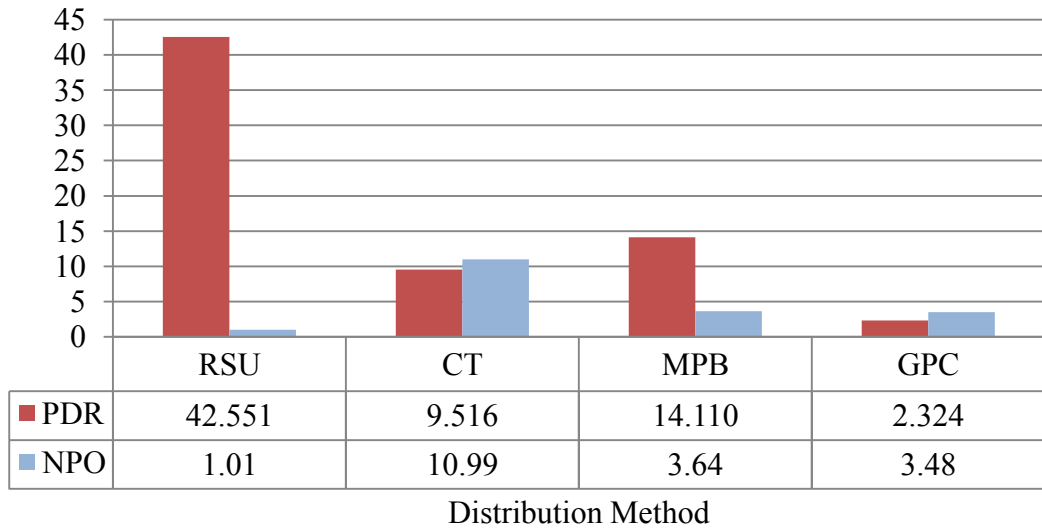


Figure 30. Packet deliver ratio and normalized packet overhead for OBU's from Hurgen-Jona with 1275 OBU's.

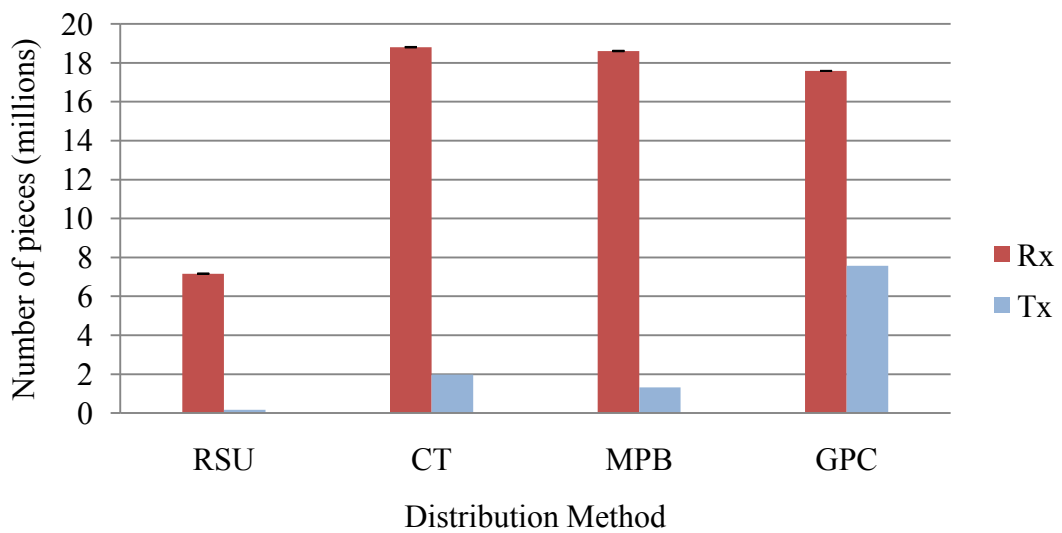


Figure 31. Number of transmitted and received pieces from Hurgen-Jona with 1275 OBU's.

8.5 Comparison of Results

Overall, MPB distributed the CRL to the most number of OBUs with the least amount of network resources. MPB consistently distributed the CRL to the highest number of OBUs during each vehicle trace simulation, as shown in the top of Figure 32. GPC-6 exhibited completion levels similar to MPB, but at a higher resource requirement, using additional service channels and generating higher levels of network traffic. Code Torrent did not scale well to higher vehicle density levels due to increasing levels of channel contention and the reduced number of new pieces available for distribution.

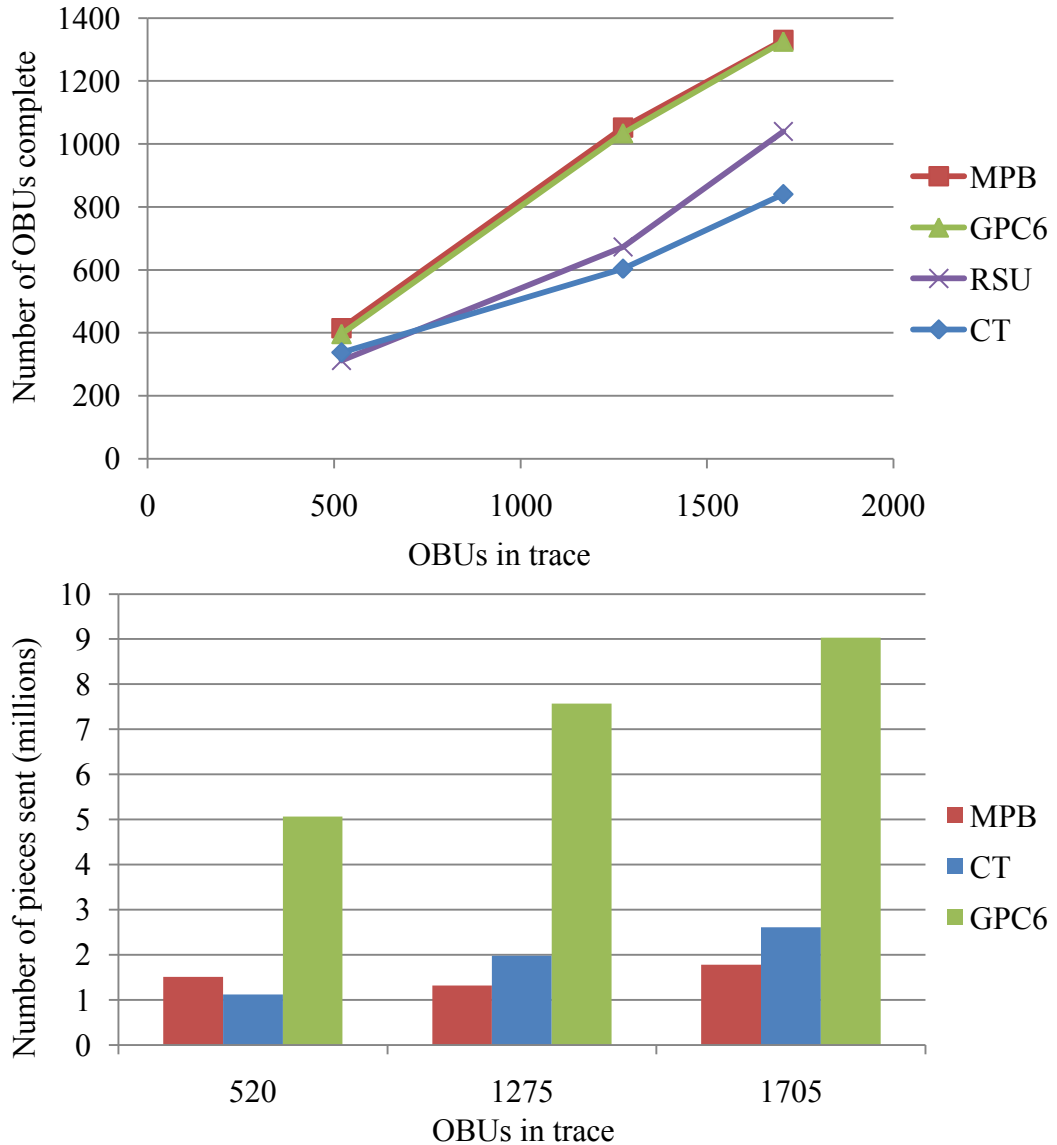


Figure 32. Comparison of completed OBUs during vehicle traces.

CHAPTER 9

CONCLUSION

9.1 Recommendations

The purpose of this research was to investigate improved CRL distribution methods for VANETs. The two CRL distribution methods developed in this research, Most Pieces Broadcast and Generation per Channel, both outperform the existing methods of RSU-only and Code Torrent for the total number of OBUs that receive the CRL file during a fixed time period. MPB and GPC outperform Code Torrent because the number of OBUs attempting to broadcast pieces during the service channel interval is significantly reduced. MPB and GPC outperform RSU-only because vehicle to vehicle communication is used so that OBUs can exchange CRL pieces outside of RSU radio range.

MPB requires only a single service channel, while the GPC evaluated in this study requires the use of all six service channels to achieve the higher download rate. Use of all six service channels may not be an option once VANETs are implemented. Thus, our recommendation is for MPB as the better of the two methods developed, since the amount of network traffic is reduced while still distributing the CRL effectively to a majority of the VANET users.

Coding methods should be used to distribute large files, such as the CRL, in a VANET environment. Raptor coding has several advantages over network coding in VANETs, including reduced overhead for tracking pieces and increased security of pieces through authentication and integrity checks since each piece is cryptographically signed by the CA.

The concept of splitting a large CRL into stand-alone "generations" as was done in the GPC method is effective in providing some parts of the CRL without having to

download the entire file. This concept becomes more important as the size of the CRL grows larger. While this does not directly reduce the overall size of the CRL, it does reduce the amount of information that must be received before being able to use some of the information.

Another recommendation is to adopt the "Valid After" field to limit the lifetime of pseudonyms. This allows an OBU to store pseudonyms for future use that are not immediately valid.

9.2 Future Work

The two main methods of reducing the download time we discussed were reducing the contention on the channel and reducing the size of the CRL. While this research focused more on reducing the channel contention, there are some ways to reduce the CRL size that would also be interesting to examine.

9.2.1 Method for Reducing Channel Contention

GPC and Code Torrent both transmitted several times more pieces than were received. One of the primary reasons for this is collisions due to several OBUs trying to broadcast during the same contention window slot time. One way to reduce the number of OBUs attempting to broadcast would be to limit the rate at which they transmit pieces. This can be accomplished by assigning a probabilistic forwarding rate for the OBUs in the case where every node broadcasts. This could be applied to the Code Torrent model and the GPC model, which uses Code Torrent on the separate service channels. [49] looked at this briefly, varying the forwarding rate of the nodes. They saw that as node density increased, the forwarding rate needed to decrease to allow data to flow better. This could be done by adding a probability for each OBU to send pieces. There may be some ratio of nodes per channel that produces the best throughput, so the forwarding rate could be based on this ratio. This could possibly even work to scale the broadcast rate

higher in cases of low vehicle density. [60, 83] consider probabilistic forwarding in more detail. There may be also be a way to link the vehicle density to the number of service channels used. As the vehicle density increases, more channels can be used. Combining probabilistic forwarding with the number of channels used may lead to the least amount of congestion since it would reduce the number of broadcasters to the lowest values.

Another way to reduce DSRC channel contention is to examine other means of communication to distribute the CRL, such as AM/FM radio, WiMax, cellular, or satellite. Vehicles already have several receivers available that could be used in addition to DSRC. Since obtaining the CRL is almost purely a matter of file download, the lack of a transmitter on the alternate bands does not inhibit CRL distribution. Also, since the CRL is signed by the CA to guarantee file integrity, the broadcast medium does not have to be secure.

9.2.2 Method for Reducing the CRL Size

Further research in the trade-offs between CRL size and management complexity due to CA region size is needed. One way to reduce the CRL size is to distribute regional CRLs. The use of specific generations or service channels for different CRL regions may aide in reducing the management complexity of regional CRLs. The generations could have priorities based on proximity to where the revocation was triggered. This format would also be portable from one region to another since the generation distributed as lower priority in one region could be issued identically but with a different priority in a different region. This would allow for possibly intelligent download based on the vehicle destination, e.g., if a vehicle were traveling south the owner may not be as concerned about revocation information for northern destinations. A channel priority guide could be used in every region, such as channel 172 always having the highest priority CRL generation, with four other service channels distributing CRL generations for each of the

four cardinal directions one region away. This would also help reduce the size of the immediately needed CRL.

9.2.3 Other Factors

While not directly reducing channel contention, one way for MPB to achieve a higher CRL completion rate might involve having slightly more OBUs broadcast during the service channel intervals. GPC was successful at distributing the CRL on several service channels, possibly due to having more broadcasters with a greater variety of pieces to exchange. Instead of having only the one OBU with the most pieces broadcast, it would be interesting to evaluate if allowing the N OBUs with the most pieces broadcast, adapting MPB to "Top N Broadcast."

The current versions of the WAVE standards specify a 50% duty cycle between the control channel interval and the service channel interval. We noticed that only a small fraction of the control channel interval was used to communicate beacons. It would be interesting to study if more time could be given to the service channels to increase the amount of data transferred during a complete synch interval.

APPENDIX A

NS-2 MOBILITY TRACE CONVERTER

The mobility trace files used in this simulation were from a project by Naumov, Baumann and Gross, in 2006 [68]. The traces were developed using the Multi-agent Microscopic Traffic Simulator (MMTS) developed at ETH Zurich, Switzerland [84, 85]. The original trace file has 259,978 vehicles traveling in an area of 354 kilometers by 263 kilometers for 76,603 seconds. The original trace is divided into different geographic regions for time blocks of 300 seconds. Several different city and highway models are available, each with low, medium, and high vehicle density. Each trace begins with an explanation of the trace, including map size, number of vehicles, and length of time the trace runs. The complete header is shown in Figure 33. The header information also includes the original trace the file was taken from.

The Switzerland traces [67] could not be used directly in ns-3 since the traces were developed for ns-2. The ns-3 file Ns2MobilityHelper interprets ns-2 mobility file

```
# Statistics:
# T_min:    0.000000    T_max:    300.030000
# X_min:    10    X_max:    8510
# Y_min:    10    Y_max:    4510
# V_min:    2.436752    V_max:    32.441287    V_avg:13.675002
# nn:  1275  area:  38250000.000000
# events:    8890  hosts/area:    0.333333
# Last mov would end time at (time): 457.020
#
# Processed from file: hw-ct-hurgen-jona-1day.filt.0.mov
# ADJUSTMENT: X_min=699490.0 Y_min=229490.0 T_min=25359.980000
T_max= T_scalar=1
```

Figure 33. Trace file header information.

data in the form of

$$(x.velocity, y.velocity, z.velocity). \quad (16)$$

The Switzerland trace files were in an alternate data format of

$$(x.position, y.position, velocity). \quad (17)$$

Additionally, specific ns-2 commands were in the trace files to control the wireless interface of the OBU. These commands were not recognizable to the ns-3 interpreter file. A sample of the trace file is shown in Figure 34.

Thus, some parsing of the trace files was required, as well as writing an ns-3 interpreter of the ns-2 trace format. Fortunately, an ns-3 contributor (Francesco Malandrino) had made significant progress on an ns-2 trace interpreter, Ns2WaypointMobilityHelper, so only minor work was needed to make it compatible with the Switzerland traces. Mr. Malandrino's file parsed the alternate ns-2 trace format in the form of (17) and converted to the form of (16) and passed the converted values to the original Ns2MobilityHelper file.

The first improvement we made to the program included turning wireless

```
$ns_ at 0.0 "$node_(794) switch OFF" # set_X,Y,Z
$node_(794) set X_ 0.000000
$node_(794) set Y_ 0.000000
$node_(794) set Z_ 0.0
$ns_ at 32.010000 "$node_(794) setdest 4970.000000 2970.000000
1000000000.000000" # init_node
$ns_ at 32.020000 "$node_(794) switch ON" # inside
$ns_ at 32.020000 "$node_(794) setdest 5381.624912 3248.704367 13.639450" #
$ns_ at 32.020000 "$node_(294) setdest 10.000000 2939.138577 12.486172" #
$ns_ at 57.948750 "$node_(294) switch OFF" # leaving_area
$ns_ at 32.020000 "$node_(310) setdest 3571.896934 3462.382148 10.533557" #
$ns_ at 32.020000 "$node_(124) setdest 4270.000000 1290.000000 9.418602" #
$ns_ at 32.020000 "$node_(424) setdest 4650.000000 1770.000000 2.858696" #
$ns_ at 32.020000 "$node_(234) setdest 4970.000000 2970.000000 12.499433" #
```

Figure 34. Sample from trace file hw-ct-hurgen-jona-1day.high.0.adj.mov.

interfaces on and off. This simulates the situation when OBUs enter and leave the boundaries of the simulation as well as when OBUs arrive at their destination and turn off the car, effectively leaving the simulation. The following two lines of code schedule the interface to turn on (SetUp) or turn off (SetDown) depending on the trace file.

```

 Simulator::Schedule (Seconds(pr->dvals[2]), &Ipv4::SetUp,
 (NodeList::GetNode(pr->ivals[3]+m_nRSU) ->GetObject<Ipv4>
 (), 1);

 Simulator::Schedule (Seconds(pr->dvals[2]), &Ipv4::SetDown,
 (NodeList::GetNode(pr->ivals[3]+m_nRSU) ->GetObject<Ipv4>
 (), 1);

```

The next improvement deals with cleaning up the mobility paths of the vehicles in the trace file. In the conversion from the form in (17) to the form of (16), round-off errors and position inconsistencies were introduced into the vehicle paths, resulting in a

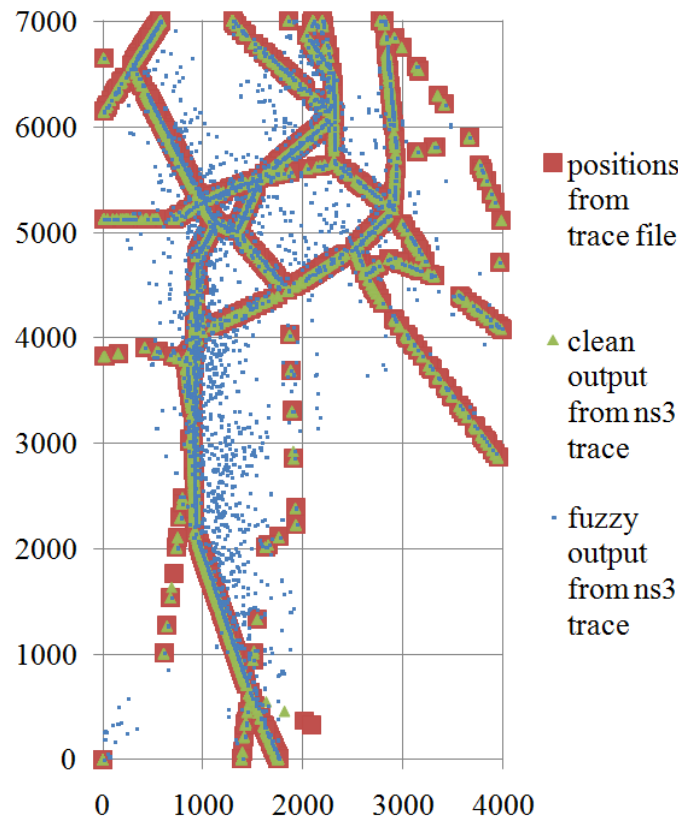


Figure 35. Comparison of vehicle locations from the original trace file, the current ns-2 trace reader, and the improved ns-2 trace reader.

"fuzzy" mobility output. These deviations from the trace file were corrected by ensuring that events were scheduled in the correct order and by correcting the location to the specified destination of the trace file, introducing minor corrections to positions. The result of the "clear" mobility output is shown in Figure 35, along with the locations specified in the trace file and the "fuzzy" output.

The improvement to the interpreter also ensured that node mobility was continuous instead of having discontinuities in the motion. This is shown in Figure 36. The trace on the left shows several places where the node stopped and then moved instantaneously to a new location. By ensuring that stops and starts were scheduled in the correct order from the original trace file, the trace on the right shows continuous mobility by the node, as intended by the original trace file.

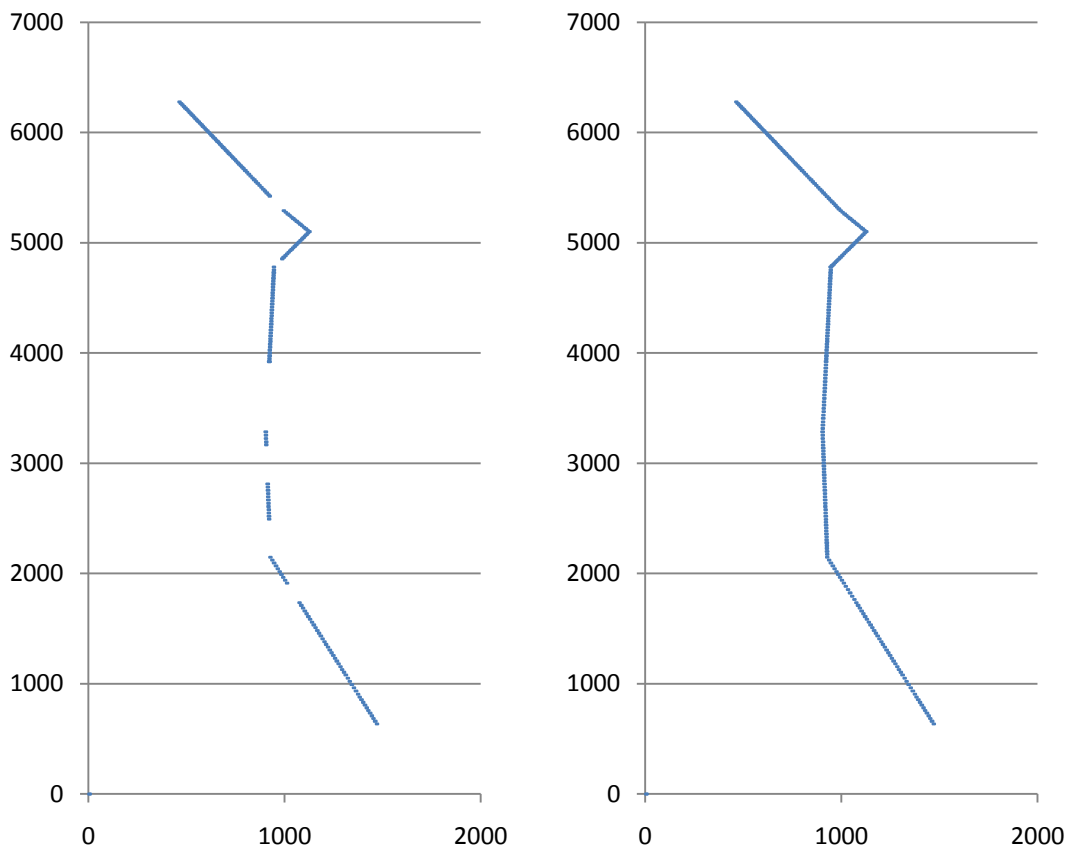


Figure 36: Mobility trace from the output of ns-3 for a single node. The trace on the left shows discontinuous movement. The trace on the right shows continuous movement.

REFERENCES

- [1] H. Hartenstein and K. P. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164--171, June 2008.
- [2] "Vehicle Safety Communications Project - FINAL REPORT - CAMP IVI Light Vehicle Enabling Research Program, DOT HS 810 591," U.S. National Highway Traffic Safety Administration, 2006.
- [3] N. H. T. S. Administration, "Traffic Safety Facts, DOT HS 811 172 ", 2009.
- [4] "Traffic Safety Facts, DOT HS 811 291," U.S. National Highway Traffic Safety Administration, 2010.
- [5] "The Economic Impact of Motor Vehicle Crashes, 2000," U.S. National Highway Traffic Safety Administration, 2002.
- [6] A. A. Carter and J. Chang, "Using Dedicated Short Range Communications for Vehicle Safety Applications – the Next Generation of Collision Avoidance," 2009.
- [7] H. Hartenstein and K. P. Laberteaux, "VANET: Vehicular Applications and Inter-Networking Technologies," John Wiley & Sons, Ltd, 2010.
- [8] M. Raya and J.-P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks* Alexandria, VA, USA, 2005.
- [9] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages*, IEEE Standard 1609.2-2006, 2006.
- [10] "FCC Report and Order 99-305," 1999.
- [11] Anon., "IntelliDrive Michigan Test Bed," <http://www.intellidriveusa.org/research/michigan-testbed.php> (Accessed March 8, 2010).
- [12] Anon., "UCLA Campus Vehicular Testbed," <http://www.vehicularlab.org/> (Accessed March 8, 2010).

- [13] T. Leinmueller, L. Buttyan, J.-P. Hubaux, F. Kargl, R. Kroh, "SEVECOM - Secure Vehicle Communication," 2006.
- [14] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager*, IEEE Std 1609.1-2006, 2006.
- [15] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services*, IEEE Std 1609.3-2007, 2007.
- [16] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation*, IEEE Std 1609.4-2006, 2006.
- [17] *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/D9.0, July 2009, 2009.
- [18] *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), 2007.
- [19] *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 Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)), 2005.
- [20] M. Torrent-Moreno, S. Corroy, F. Schmidt-Eisenlohr, and H. Hartenstein, "IEEE 802.11-Based One-Hop Broadcast Communications: Understanding Transmission Success and Failure Under Different Radio Propagation Environments," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems* Torremolinos, Spain: ACM, 2006.
- [21] *RFC5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, 2008.
- [22] B. Parno and A. Perrig, "Challenges in Securing Vehicular Networks," in *HotNets-IV*, College Park, Maryland, 2005.

- [23] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing Vehicular Communications," *IEEE Wireless Communications*, vol. 13, pp. 8--15, 2006.
- [24] J. J. Haas, Y.-C. Hu, and K. P. Laberteaux, "Design and analysis of a lightweight certificate revocation mechanism for VANET," in *Proceedings of the sixth ACM international workshop on Vehicular InterNetworking* Beijing, China: ACM, 2009, pp. 89-98.
- [25] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, "Secure Vehicular Communication Systems: Design and Architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100--109, November 2008.
- [26] P. Wex, J. Breuer, A. Held, T. Leinmuller, and L. Delgrossi, "Trust issues for vehicular ad hoc networks," in *IEEE Vehicular Technology Conference, 2008*, 2008, pp. 2800--2804.
- [27] P. D. Dawoud, D. S. Dawoud, and R. Peplow, "A proposal for secure vehicular communications," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* Seoul, Korea: ACM, 2009, pp. 1026-1032.
- [28] B. Bellur, "Certificate assignment strategies for a PKI-based security architecture in a vehicular network," in *IEEE Global Telecommunications Conference* New Orleans, LA, 2008.
- [29] M. Nowatkowski, J. Wolfgang, C. McManus, and H. Owen III, "The Effects of Limited Lifetime Pseudonyms on Certificate Revocation List Size in VANETs," in *IEEE SoutheastCon* Charlotte, North Carolina 2010.
- [30] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks* Montreal, Quebec, Canada, 2007.
- [31] P. Kamat, A. Baliga, and W. Trappe, "Secure, pseudonymous, and auditable communication in vehicular ad hoc networks," *Security and Communication Networks*, vol. 1, pp. 233--244, 2008.
- [32] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of Misbehaving and Faulty Nodes in Vehicular Networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, 2007.

- [33] A. Wasef, J. Yixin, and S. Xuemin, "ECMV: efficient certificate management scheme for vehicular networks," in *IEEE Global Telecommunications Conference* New Orleans, LA, 2008, pp. 1--5.
- [34] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, "Security in vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. pp. 88--95, 2008.
- [35] S. Eichler and B. Muller-Rathgeber, "Performance analysis of scalable certificate revocation schemes for ad hoc networks," in *The IEEE Conference on Local Computer Networks*, 2005.
- [36] A. Arnes, "Public Key Certificate Revocation Schemes." vol. Sivilingeniør: Norwegian University of Science and Technology, 2000.
- [37] M. C. Morogan and S. Muftic, "Certificate Revocation System Based on Peer-to-Peer CRL Distribution," in *International Conference on Distributed Multimedia Systems*, Miami, FL, 2003.
- [38] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J.-P. Hubaux, "Certificate Revocation in Vehicular Networks," 2006.
- [39] Z. Ma, F. Kargl, and M. Weber, "Pseudonym-on-demand: a new pseudonym refill strategy for vehicular communications," in *IEEE 68th Vehicular Technology Conference*, 2008, pp. 1--5.
- [40] A. Rao, A. Sangwan, A. A. Kherani, A. Varghese, B. Bellur, "Secure V2V communication with certificate revocations," in *2007 Mobile Networking for Vehicular Environments*, 2007, pp. 127--132.
- [41] S. Balfe, E. Gallery, C. J. Mitchell, and K. G. Paterson, "Challenges for Trusted Computing," *Security & Privacy, IEEE*, vol. 6, pp. 60-66, 2008.
- [42] FBI, "Crime in the United States 2005," 2005.
- [43] "Traffic Safety Facts," U.S. National Highway Traffic Safety Administration, 2007.
- [44] "State Transportation Statistics," Bureau of Transportation Statistics, 2006.
- [45] Anon., "A Beginner's Guide to BitTorrent," <http://www.bittorrent.com/btusers/guides/beginners-guide> (Accessed March 8, 2010).

- [46] D. J. C. MacKay, "Information Theory, Inference, and Learning Algorithms," 7.2 ed: Cambridge University Press, 2005.
- [47] P. Papadimitratos, G. Mezzour, and J.-P. Hubaux, "Certificate Revocation List Distribution in Vehicular Communication Systems," in *The Fifth ACM International Workshop on Vehicular Internetworking (VANET 2008)*, San Francisco, CA, USA, 2008.
- [48] Y. Min and Y. Yuanyuan, "Peer-to-Peer File Sharing Based on Network Coding," in *The 28th International Conference on Distributed Computing Systems*, 2008, pp. 168--175.
- [49] A. Fujimura, S. Y. Oh, and M. Gerla, "Network coding vs. erasure coding: Reliable multicast in ad hoc networks," in *IEEE Military Communications Conference*, 2008.
- [50] K. P. Laberteaux, J. J. Haas, and Y.-C. Hu, "Security Certificate Revocation List Distribution for VANET," in *The Fifth ACM International Workshop on Vehicular Internetworking (VANET 2008)*, San Francisco, CA, USA, 2008.
- [51] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Co-operative Downloading in Vehicular Ad-Hoc Wireless Networks," in *Second Annual Conference on Wireless On-demand Network Systems and Services*, 2005.
- [52] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, pp. 2551-2567, 2006.
- [53] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204--1216, 2000.
- [54] U. Lee, J.-S. Park, S.-H. Lee, W. W. Ro, G. Pau, "Efficient Peer-to-Peer File Sharing Using Network Coding in MANET," *Journal Of Communication and Networks*, vol. 10, 31 December 2008.
- [55] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, "Code Torrent: Content Distribution Using Network Coding in VANET," in *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking* Los Angeles, California: ACM, 2006.
- [56] M. Nowatkowski, J. Wolfgang, C. McManus, and H. Owen III, "Cooperative Certificate Revocation List Distribution Methods in VANETs," in *AdHocNets* Niagara Falls, Ontario, Canada: ICST, 2009.

- [57] L. Seung-Hoon, L. Uichin, L. Kang-Won, and M. Gerla, "Content Distribution in VANETs Using Network Coding: The Effect of Disk I/O and Processing O/H," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, 2008.
- [58] *RFC5053, Raptor Forward Error Correction Scheme for Object Delivery*, 2007.
- [59] R. Koetter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *Information Theory, IEEE Transactions on*, vol. 54, pp. 3579-3591, 2008.
- [60] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 450-463, 2008.
- [61] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 63-68, 2006.
- [62] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard," in *IEEE 66th Vehicular Technology Conference, 2007*, 2007, pp. 2199-2203.
- [63] M. Nowatkowski and H. Owen III, "Certificate Revocation List Distribution in VANETs Using Most Pieces Broadcast," in *IEEE SoutheastCon*, Charlotte, North Carolina, 2010.
- [64] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, "Architecture for Secure and Private Vehicular Communications," in *The 7th International Conference on ITS Telecommunications*, Sophia Antipolis, France, 2007.
- [65] F. Kargl, P. Papadimitratos, L. Buttyan, M. Mütter, E. Schoch, "Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 110--118, 2008.
- [66] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing Vehicular Communications - Assumptions, Requirements, and Principles," in *Workshop on Embedded Security in Cars (ESCAR)*, Berlin, Germany, 2006.
- [67] Anon., "Realistic Vehicular Traces," <http://www.lst.inf.ethz.ch/research/ad-hoc/car-traces/> (Accessed March 8, 2010).
- [68] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proceedings of the 7th ACM*

international symposium on Mobile ad hoc networking and computing Florence, Italy: ACM, 2006, pp. 108-119.

- [69] K. C. Lee, L. Seung-Hoon, C. Ryan, L. Uichin, and M. Gerla, "First Experience With Car Torrent in a Real Vehicular Ad Hoc Network Testbed," in *2007 Mobile Networking for Vehicular Environments*, 2007.
- [70] S. Das, A. Nandan, G. Pau, M. Y. Sanadidi, and M. Gerla, "SPAWN: A Swarming Protocol for Vehicular Ad-Hoc Wireless Networks," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks* Philadelphia, PA, USA, 2004.
- [71] J. J. Blum and A. Eskandarian, "A Reliable Link-Layer Protocol for Robust and Scalable Intervehicle Communications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 4--13, 2007.
- [72] Anon., "The ns-3 network simulator," <http://www.nsnam.org/index.html> (Accessed March 8, 2010).
- [73] F. Schmidt-Eisenlohr, M. Torrent-Moreno, J. Mittag, and H. Hartenstein, "Simulation platform for inter-vehicle communications and analysis of periodic information exchange," in *Fourth Annual Conference on Wireless on Demand Network Systems and Services, 2007*, 2007, pp. 50--58.
- [74] F. Schmidt-Eisenlohr, M. Torrent-Moreno, T. Tielert, J. Mittag, and H. Hartenstein, "Cumulative Noise and 5.9 GHz DSRC Extensions for ns-2.28," Universitat Karlsruhe (TH), 2006.
- [75] J. F. Haerri, Fethi;Bonnet, Christian, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," 2006.
- [76] V. Naumov and T. R. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1919-1927.
- [77] J. Banks, "Handbook of Simulation," John Wiley & Sons, Inc., 1998.
- [78] X. Chen, H. H. Refai, and X. Ma, "A quantitative approach to evaluate DSRC highway inter-vehicle safety communication," in *IEEE Global Telecommunications Conference, 2007*, 2007, pp. 151--155.

- [79] A. M. Law, "Statistical analysis of simulation output data: the practical state of the art," in *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come* Washington D.C.: IEEE Press, 2007.
- [80] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, Third ed.: McGraw Hill, 2000.
- [81] J. Banks, J. S. Carson, and B. L. Nelson, *Discrete-Event System Simulation*, Second ed.: Prentice-Hall, 1999.
- [82] Anon., "ns-3 Manual," <http://www.nsnam.org/docs/manual.html> (Accessed March 8, 2010).
- [83] J. Widmer and J.-Y. L. Boudec, "Network coding for efficient communication in extreme networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 284-291.
- [84] B. Raney, A. Voellmy, M. Vrtic, and K. Nagel, "Towards a microscopic traffic simulation of all of Switzerland," in *International Conference on Computational Science* Amsterdam, The Netherlands 2002.
- [85] B. Raney, A. Voellmy, M. Vrtic, K. Axhausen, and K. Nagel, "An agent-based microsimulation model of Swiss travel," *Networks and Spatial Economics*, pp. 23-41, 2003.

VITA

MICHAEL E. NOWATKOWSKI

Michael E. Nowatkowski was born in Houston, Texas. He grew up in Texas, living there through the end of high school. He received a B.S. in Electrical Engineering from Rose-Hulman Institute of Technology in Terre Haute, Indiana and was commissioned as a second lieutenant in the United States Army in 1990. He served as a signal corps officer at several posts, including Fort Campbell, Kentucky and Fort Bragg, North Carolina. He received an M.S. in Electrical and Computer Engineering from Georgia Institute of Technology in 2000 and joined the Systems Engineering faculty at the United States Military Academy at West Point, New York for three years. After serving three more years as a signal officer at Fort Bragg, North Carolina, he returned to Georgia Institute of Technology to pursue a doctorate in Electrical and Computer Engineering. After completing his doctorate in May 2010, he joins the Electrical Engineering and Computer Science faculty at the United States Military Academy at West Point, New York. His research interests include network security, wireless networks, and simulation.