Abusing Keep-Alive Forwarding to flood a VANET
- When safety messages become a safety risk

Johan Billman
Victor Hellström

Supervisor: Abdeldjalil Boudjadar
Examiner: Mikael Asplund
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Johan Billman
Victor Hellström
Abstract

Cooperative Intelligent Transport Systems (C-ITS) enable vehicles to cooperate with each other and can be used to increase traffic safety and traffic flow. There are several standard initiatives for C-ITS, such as WAVE in the US, ARIB in Japan and ETSI ITS G5 in Europe.

Decentralized Environment Notification Message (DENM) is a kind of message within ETSI TC ITS. DENMs are event-triggered, multi-hop notification messages that can be forwarded using Keep-Alive Forwarding (KAF).

In this thesis, KAF is used to flood the network with DENMs sent from a single attacking vehicle. Using the simulation tools SUMO, OmNet++ and Veins, a highway scenario is studied. In the scenario, a vehicle sends out a DENM into a network that is being attacked by one other vehicle. Due to how KAF allows messages to be forwarded the other vehicles help flood the network. The results clearly indicate that KAF can be abused to create a Denial of Service (DoS) attack. After just a few seconds, the attack has introduced large delays and very high packet loss. The delay to receive a DENM is increased by several orders of magnitude and the packet loss reach unacceptable levels. By getting DENMs several seconds after they are needed, or by not getting them at all, could lead to traffic hazards.
Acknowledgments

We would like to give our sincerest thanks to our examiner Mikael Asplund and also to our supervisor Abdeldjalil Boudjadar for their guidance and feedback. No question ever felt too big or too small to ask. We would also like to thank Ola Leifler for his insightful comments.

To our fellow students around the office Magnus Axetun, Rikard Jonsson and Alfons Råberg we would like to say Good game.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Research Question</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Methodology</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Delimitations</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Related Works</td>
<td>4</td>
</tr>
<tr>
<td><strong>2 Theory</strong></td>
<td>5</td>
</tr>
<tr>
<td>2.1 Networks and Vehicular Networks</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Protocol Stack of ETSI ITS G5</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Forwarding</td>
<td>8</td>
</tr>
<tr>
<td><strong>3 Approach</strong></td>
<td>11</td>
</tr>
<tr>
<td>3.1 Evaluation of Simulation Environment</td>
<td>11</td>
</tr>
<tr>
<td>3.2 An ETSI ITS application layer in the WAVE stack</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Simulation scenario</td>
<td>17</td>
</tr>
<tr>
<td><strong>4 Results</strong></td>
<td>20</td>
</tr>
<tr>
<td>4.1 Vehicles reached by non-attack messages</td>
<td>20</td>
</tr>
<tr>
<td>4.2 Delays</td>
<td>21</td>
</tr>
<tr>
<td>4.3 DENM Volume</td>
<td>22</td>
</tr>
<tr>
<td>4.4 Packet Loss</td>
<td>23</td>
</tr>
<tr>
<td><strong>5 Discussion</strong></td>
<td>25</td>
</tr>
<tr>
<td>5.1 Outcome of the attack</td>
<td>25</td>
</tr>
<tr>
<td>5.2 Number of vehicles in simulation</td>
<td>25</td>
</tr>
<tr>
<td>5.3 Number of simulations</td>
<td>25</td>
</tr>
<tr>
<td>5.4 Simulation length</td>
<td>25</td>
</tr>
<tr>
<td>5.5 Our implementation within a WAVE stack</td>
<td>26</td>
</tr>
<tr>
<td>5.6 Choice of the road to simulate</td>
<td>26</td>
</tr>
<tr>
<td>5.7 Errors within the standard</td>
<td>26</td>
</tr>
<tr>
<td>5.8 Why hacking is a good thing</td>
<td>26</td>
</tr>
<tr>
<td><strong>6 Conclusion</strong></td>
<td>28</td>
</tr>
</tbody>
</table>
# List of Figures

2.1 Inter-vehicle communication ........................................... 5  
2.2 The ETSI and WAVE stack in an OSI model ............................ 6  
2.3 General structure of a DENM .......................................... 7  
2.4 Forwarding methods. Courtesy of Wikimedia Commons ............ 8  
2.5 DENM Reception flowchart .............................................. 9  
2.6 KAF timer flowcharts .................................................. 10  
3.1 The architecture of Veins ............................................ 12  
3.2 OMNeT++ in action .................................................... 13  
3.3 SUMO GUI image ...................................................... 14  
3.4 The network stack in our implementation ............................ 15  
3.5 Overview of implementation .......................................... 16  
3.6 Motorway scenario .................................................... 17  
4.1 Vehicles that did not received the blue vehicle’s DENM .......... 20  
4.2 Delay of DENMs ....................................................... 21  
4.3 Sent and received DENMs ............................................. 22  
4.4 Number of DENM copies .............................................. 23  
4.5 Packet loss ............................................................. 24
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>DENM Management parameters</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>Blue vehicle’s DENM parameters</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Background DENMs</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Flooding DENMs</td>
<td>19</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Motivation

Tomorrow’s vehicles will be designed to drive themselves and to cooperate to maximize traffic flow and increase safety. Inter-vehicle communication (IVC) enables vehicles to communicate with each other and with other infrastructure, such as base stations. The communication can be short range or long range, and can help to improve the safety on the road and make the traffic flow more efficient [15]. Information that might be shared are e.g., emergencies, traffic congestion, weather and road data. Some of this information may be of interest to every vehicle in a given area and can therefore be broadcast to every vehicle [12].

Several standard initiatives for the Intelligent Transport Systems (ITS) exist. In Europe it is the European Telecommunications Standard Institute (ETSI) Technical Committee ITS (hereafter known as the standard). Within this European standard there are two primary standardized message types, Cooperative Awareness Message (CAM) and Decentralized Environment Notification Message (DENM) [8]. CAMs are used to keep roadside beacons and nearby vehicles aware of the surrounding vehicles. DENMs are used to warn other vehicles about potential hazards. DENMs are generated by ITS applications and are event-triggered, multi-hop warning messages.

ETSI ITS uses the GeoNetworking protocol for the network layer protocol, which supports multi-hop communication. This feature creates the possibility to forward messages (e.g., DENMs) further than the vehicle’s normal single-hop communication range (roughly 500m in good conditions).

CAMs and DENMs share channels within the 5.9 GHz band. The channel used has a data rate of 6 Mbps. This means that if more than 1400 messages are transferred during the same second the data rate is exceeded, assuming a message size of 550 Bytes. This limit in capacity might have effects on the delivery-rates and the message delay.

A potential flaw with DENMs is that they may flood the network if several vehicles report the same hazard and cause loss of performance. The performance loss caused by multiple reports of a single hazard has been studied by Tubbene [21]. In his studies, Tubbene found that when several vehicles report the same problem the network traffic increases disproportionately. This motivates the study of the related problem that if someone tries to intentionally flood the network with a Denial of Service (DoS) attack or alike it could potentially make it impossible to send important messages about road hazards.
1.2 Research Question

The purpose of this thesis is to analyze how vulnerable the ETSI ITS is to flooding of malicious DENMs using simulations.

Keep-Alive Forwarding is a mechanism used in ETSI ITS to forward DENMs, and Keep-Alive Forwarding makes it possible to tell others to forward messages as well. Because one can tell others to forward messages, there may be a security risk to the system to abuse the Keep-Alive forwarding mechanism to flood the network.

1. Can the Keep-Alive Forwarding mechanism be abused to overload the network with malicious DENMs?

1.3 Methodology

To be able to analyze how the ETSI ITS handles flooding, we will use simulations to answer the research question. The used simulation methodology is inspired by the work of Black et al. in Simulation methodology - A practitioner’s perspective [23]. Major phases concerning building, verifying and validating simulation models are defined and described in the article, also presentation and documentation of results are discussed. In this thesis the major phases are the following:

1. Identify and define the thesis’s scope and objectives.
2. Describe the theory needed to perform the analysis e.g. DENMs and KAF.
3. Study VANET simulations and choose a simulation framework.
4. Implement DENMs and KAF as application layer features in chosen simulation framework.
5. Verify that the implemented simulation model complies to the standard. Make necessary enchantments to other layers so that they complies with an ETSI ITS application layer.
6. Perform simulation runs and record message delays, DEMMs sent and received.
7. Analyze the data and discuss the results.

1.4 Delimitations

In this thesis the focus is on using Keep-Alive Forwarding for DENMs and its timers to cause flooding. Only features located within the network layer will be studied.
1.5 Related Works

Published results regarding Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) that touch upon the European C-ITS system have focused on the performance of CAM messages in different ways. One example is Eckhoff et al. [4] who compare ETSI ITS G5 to the American counterpart IEEE Wireless Access in Vehicular Environments (WAVE). Kloiber et al. [11] look at how CAMs perform in high-density scenarios. Choice of dissemination strategy is also an important factor to performance as shown by Bohm et al. [2], the need to keep overhead traffic to a minimum is emphasized. Other researchers have also touched upon giving different priorities to different ITS-S to ensure the delivery of time critical messages [20].

Since forwarding of DENMs is permitted in the standard, this can result in so called broadcast storms [19]. Wisitponghan et al. [27] have proposed three probabilistic and timer-based techniques to mitigate the losses in performance of communication. These techniques achieve a reduction of packet losses and keep packet delays at acceptable levels while being computationally cheap and possible to distribute.

Garip et al. [10] explored using vehicles as mobile botnets. The authors simulate cars in a "Manhattan like" environment and try to make other cars take different routes because of presumed road congestion. Possible uses of an attack like this can be to make possible customers miss a store due to the new route not passing it.

Our work gets inspiration from Garip et al. in the way they use compromised vehicles as nodes in a botnet. We explore the possibility of using uncompromised vehicles to create a Denial of Service attack on a VANET.
This chapter will provide the theoretical background needed to read this thesis.

2.1 Networks and Vehicular Networks

One could say that modern vehicles are more of a network of computers than an engine and four wheels. These computers control different parts of the vehicle. There are systems that control safety data, control data, multimedia and more [22]. Some of the networks are the FlexRay that is used for safety-critical communication [14], Controller Area Network which is a protocol that enables devices in a vehicle to communicate with each other [22], the Local Interconnect Network (LIN) that controls e.g., the windshield wipers [18] and the Media Oriented Systems Transport which is responsible for transporting multimedia in the vehicle [22].

In modern vehicles there are several of external connections as well, such as Bluetooth to connect your mobile to the media players and 3G/4G connections for online services. Inter-vehicle communications is an emerging technology which will be used by modern vehicles in the near future. All these possibilities to connect to the vehicle also give an increased exposure that might put the internal network at risk. This creates a great incentive to secure all networks in a modern vehicle to ensure that no one changes the properties of the vehicle. This overview is based on the work by Studnia et al. [18].

Figure 2.1: Inter-vehicle communication. Courtesy of Manfred Antranias
2.2 Protocol Stack of ETSI ITS G5

The European Telecommunications Standards Institute has over the recent years been publishing its own OSI-like protocol stack adapted to the demands of VANETs. Compared to the Internet a VANET has a rapidly changing topology and lacks centralized coordination infrastructure.

An overview of the ETSI ITS stack can be seen in Figure 2.2 alongside its American counterpart. ETSI uses a networking layer called GeoNetworking [5] which compared to the corresponding layer in the Internet stack has capabilities such as addressing vehicles in a specific geographical area. This specific ability is called geocasting. This is made possible using the location data contained in CAMs.

The Basic Transport Protocol (BTP) is the transport protocol in the ETSI stack. BTP is a connection-less protocol in similarity with User Datagram Protocol (UDP) in the Internet stack and therefore does not guarantee delivery.

The ITS facilities layer does not correspond to any single layer in the OSI model and should instead be viewed as a combination of the session layer, presentation layer and parts of the applications layer [6][7]. It provides services and functionalities like managing sessions between inter-host communications and encoding, decoding and encryption of application data.

DENMs are a message type specified at the application layer in the ETSI ITS stack and explained in more detail in subsection 2.2.1.

We compare ETSI ITS G5 to WAVE here due to us using parts of both stacks in our implementation.

![Figure 2.2: The ETSI and WAVE stack in an OSI model.](image)

2.2.1 Decentralized Environmental Notification Messages

A DENM contain information regarding road traffic events e.g., traffic jams or roadworks. A DENM is triggered and sent by a vehicle or a roadside unit to other vehicles to notify that a road traffic event has occurred in a given area. The vehicles that receive the DENM can then take actions to bypass the event safely [3].
2.2. Protocol Stack of ETSI ITS G5

A DENM contains an ITS PDU (Protocol Data Unit) header, management container, situation container, location container and À la carte container as seen in 2.3. The ITS PDU header includes the information of the protocol version, the type of message and the ITS-S ID (stationID) of the creator or re-transmitter of the message.

The Management container consists of parameters that are related to the DENM management and protocol [9]. An overview of the parameters investigated in this thesis within the Management container is given in Table 2.1. Other parameters contained within a DENM are either optional, or not used within KAF.

The situation container is optional. It contains the CauseCode that describes what the event is about. The CauseCode can be e.g., tire puncture or roadblock. Similar to situation container, the location container and À la carte container are optional and will not be further explained in this thesis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionID</td>
<td>A unique number used to identify the event</td>
</tr>
<tr>
<td>detectionTime</td>
<td>When the event occurred</td>
</tr>
<tr>
<td>referenceTime</td>
<td>When the event were sent to other ITS-Ss</td>
</tr>
<tr>
<td>eventPosition</td>
<td>Where the event occurred</td>
</tr>
<tr>
<td>relevanceDistance</td>
<td>The area in which the event is relevant</td>
</tr>
<tr>
<td>relevanceTrafficDirection</td>
<td>The direction which the event is relevant, i.e., north</td>
</tr>
<tr>
<td>validityDuration</td>
<td>How many seconds the event should be valid</td>
</tr>
<tr>
<td>transmissionInterval</td>
<td>How many ms between each re-transmission</td>
</tr>
<tr>
<td>stationType</td>
<td>Which type of vehicle has created and sent the event, i.e., car or bus</td>
</tr>
</tbody>
</table>

Table 2.1: Overview of some DENM Management parameters

To determine the parameter values within a DENM, the vehicle relies on its sensors to make sure that the values are correct, i.e. the vehicle can itself set them to whatever it wants them to be.
2.3 Forwarding

![Image of forwarding methods]

(a) Unicast forwarding.  (b) Broadcast forwarding.

Figure 2.4: Forwarding methods. Courtesy of Wikimedia Commons.

In networking, a node can generally only reach adjacent nodes. To solve this problem networking protocols can implement forwarding. If a node receives a packet that is addressed to some other node it retransmits the message. This is done in ETSI ITS G5 as well, so vehicles can communicate with ITS stations and other vehicles that are not within range. Forwarding can be done in several ways. Within ETSI ITS unicast and broadcast are used.

With unicast, a node receiving a packet not addressed to itself sends it to a single neighboring node. Which one it chooses to send the packet to depends on the nodes routing table. In the routing table information about distances to other nodes are stored and the node with the shortest distance is chosen. This means that there is never duplicates of the packet traversing the network.

With broadcast, a node receiving a packet not addressed to it sends it to all its connected nodes. This removes the need for a routing table. The downside is that there will be several copies of the packet traversing the network.

2.3.1 Keep-Alive Forwarding

In ETSI ITS G5, Keep-Alive Forwarding (KAF) is described as a forwarding-scheme for DENMs. KAF uses a cache for storing DENMs before forwarding them. When a DENM is received, KAF first checks if a copy of the received DENM is already stored in its cache. If a copy is stored, KAF checks if the received DENM’s referenceTime (when the DENM was sent) is equal or greater than the copy. If it is equal, KAF will restart the timer that is responsible for when to forward the copy. If the referenceTime is greater than the copy’s referenceTime KAF will calculate if the received DENM is valid. This is done by calculating the timer $T_{F\_Validity}$, that is described below. If it is valid, KAF will calculate the timer $T_{Forwarding}$, which tells when to forward the DENM, that is described below. If the two timers are valid, KAF will restart these timers and replace the copy by the received DENM. If the received DENM’s referenceTime is less than the referenceTime of the copy, KAF will discard the DENM.
2.3. Forwarding

If a new DENM that is not in the cache is being received, KAF will calculate if the DENM is valid ($T_{F, \text{Validity}}$). If valid, KAF will calculate when to forward the DENM ($T_{\text{Forwarding}}$) and start these timers and place the DENM in the cache. See the flowchart of KAF in Figure 2.5.

Figure 2.5: Flowchart of how KAF handles a received DENM.
2.3. Forwarding

There are two timers that control the forwarding of a DENM in KAF, \( T_{FValidity} \) and \( T_{Forwarding} \). \( T_{FValidity} \) is a timer that depends on detectionTime and validityDuration.

\[
T_{FValidity} = \text{detectionTime} + \text{validityDuration} \quad (2.1)
\]

If \( T_{FValidity} \) expires, it means that the validity of that DENM has run out and KAF will then cancel and remove that DENM out of the cache and will not forward this DENM again. \( T_{Forwarding} \) depends on transmissionInterval, which determines the time between retransmissions (see Table 2.1).

\[
T_{Forwarding} = 2 \times \text{transmissionInterval} + \text{ms}, t \in \{0, 1, ..., 149, 150\} \quad (2.2)
\]

When it expires it will forward the DENM [9]. A diagram of the timer’s management is displayed in Figure 2.6.

---

**Figure 2.6:** The timer \( T_{FValidity} \) expiry (left) and the timer \( T_{Forwarding} \) expiry (right).

---
3 Approach

In this chapter our implementation of the ETSI ITS G5, the simulation settings and the software used are described in our efforts to overload the network using KAF.

3.1 Evaluation of Simulation Environment

We evaluated two different open-source environments to conduct simulations in, one ETSI ITS G5 compliant called iTetris, and one WAVE compliant, called Veins.

3.1.1 iTetris

iTetris stands for "Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions". iTetris is a simulation platform that connects a traffic simulator with a network simulator and is ETSI standard compliant [16]. iTetris is distributed as virtual machine which runs an older distribution of Ubuntu, alongside old version of SUMO and ns-3, which is a network simulator similar to OMNeT++ (see 3.1.3). Running iTetris on a virtual machine would hamper simulation speeds and the fact that the version of Ubuntu within it is old makes finding drivers close to impossible if one wants to run it directly on hardware. We found the documentation lacking and the community around iTetris to be inactive, e.g. there are several forum posts with open installation issues with no replies on them for several months. These factors made us consider other options.
3.1.2 Veins

Vehicles In Network Simulation (Veins) is an Inter-Vehicular Communication (IVC) simulator that enables dynamic interactions between the network simulator OMNeT++ and the traffic simulator SUMO. The architecture of Veins is in Figure 3.1. The node mobility patterns done by OMNeT++ are statically computed and not influenced by real traffic observations, such as hazard warnings, and can be different from real life situations. Because of that OMNeT++ is not ideal for simulating IVC scenarios alone. However, SUMO uses trace files from real life measurements and can complement OMNeT++ [17]. The simulations are done in parallel via the Traffic Control Interface (TraCI) [26] that makes it possible for OMNeT++ and SUMO to communicate. OMNeT++ is providing network simulations and SUMO provides road traffic simulations. When vehicles in SUMO are moving it will be reflected in OMNeT++ as movement of nodes [17].

There is an ongoing project\textsuperscript{1} working on implementing the ETSI stack with Veins by Raphael Riebl. Since the project is ongoing there were parts of the standard that were not implemented yet, among them DENMs. After exploring the codebase\textsuperscript{2} it became apparent that implementing the parts need for us to be able to use it was a rather large task and getting a complete understanding of the project would take too much time with regard to our scope.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{The architecture of Veins [25].}
\end{figure}

\textsuperscript{1}\url{https://github.com/riebl/artery}.
\textsuperscript{2}And submitting some pull requests.
3.1. Evaluation of Simulation Environment

3.1.3 OMNeT++

OMNeT++ is an open-source component-based C++ event simulation library and framework for building communication network simulations [24]. It is available to use on common platforms such as Linux, Windows and Mac OS/X. OMNeT++ provides the framework and tools to create network simulations. It is designed to enable large-scale simulations by using hierarchical models that are reusable [24]. The relationships between the modules and the communication links between them are stored as Network Description (NED) files, and these can be modeled graphically [17]. A capture of OMNeT++ is seen in Figure 3.2

Figure 3.2: A capture of OMNeT++ graphical interface.
3.1.4 SUMO

SUMO stands for Simulation of Urban MObility and is a free and open traffic simulation suite [13]. SUMO can be used to simulate road traffic and provides a GUI to aid understanding of the situation. NETEDIT is a network editor that can be used to create own road networks that can be used in SUMO. NETEDIT also makes it possible to edit real maps with common formats such as Open Street Map (OSM). In SUMO each vehicle is defined with an identifier, departure time and its route through the network. A screen-shot is provided in Figure 3.3.

Figure 3.3: A capture of SUMO GUI with yellow vehicles in movement.
3.2 An ETSI ITS application layer in the WAVE stack

We have implemented the DENM part of ETSI ITS G5\(^3\) as the application layer of a WAVE stack, as seen in Figure 3.4. A closer look at our application layer is seen in 3.5 This was done due to no ETSI ITS G5 stack being available. Both stacks provide best-effort delivery and similar features. One major difference is the ability to use different forms of geocast, however this is optional to use when sending DENMs and we do not use it.

![Network Stack Diagram]

Figure 3.4: The network stack used in our implementation. The application layer is shown in more detail in Figure 3.5.

3.2.1 Modifying WaveShortMessages to be DENMs

The implementation was created through modifying the example provided with Veins called TraCI. To get a message containing the required headers in a DENM the file `WaveShortMessage.msg` was modified. From this file OMNeT++ then generates the C++ code required. DENMs can also contain several optional headers, however, these were not implemented.

3.2.2 Sending messages and receiving in OMNeT++

To trigger OMNeT++ to send a DENM, one has to create and add an event to OMNeT++ event queue. Events are given a trigger time when added to the event queue. When the simulation time reaches the time of an event, OMNeT++ resolves that event. For us, this means that a DENM (with its unique actionID) is to be broadcast by a vehicle. Each node (here, vehicle) reached by a message in OMNeT++ self-invokes a message that is passed to an import of the node. This message is caught by a lower layer in the network stack and sent up until it is handled. In the WAVE stack implemented in the demo provided with Veins these messages are handled by the network layer. The network layer then checks the message for different labels and calls a matching function (if one exists) in the application layer and discards the message.

---

\(^3\)More specifically ETSI EN 302 637-3
3.2.3 Keep Alive Forwarding, the OMNeT++ way

To be able to replicate the KAF flowcharts within the standard we had to rewrite the application layer provided with Veins and make minor changes to the network layer, e.g. add our functions to the message handler.

Our implementation of the application layer consists of five major parts. One is the interface to the network layer that could be kept the as provided, with just an added functionality of counting the number of passed down messages. The interface works by being given a message of type WaveShortMessage, which in our implementation is a DENM, which is passed down the network stack.

Another part is the creation of messages to be sent. These have different parameter values depending on if they are actual, spam or noise messages in our implementation. The methods handling this dynamically creates a message object, sets the values and then invokes the function handling the interface to reach the network layer.

To handle all statistics we modified the finish function. OMNeT++ calls this function when it exits a simulation. Each node formats its recorded data and identifier into a single, tab-separated, string and outputs it to a text file for further processing. Recorded data includes delays and number of messages sent and received amongst other things. OMNeT++ calls finish on objects in the same order as they are created.

The largest part is the one handling KAFs buffer. The buffer itself is implemented as an array. Due to KAF reliance on timers in its effort to keep down network traffic, our implementation dynamically has to add and remove events from OMNeT++ event queue. These events are used as timeouts for the two timers used within the forwarding scheme.

When a node receives or sends out a DENM, all nodes reached by the message go through their KAF buffer to see if it contains an entry for the specific actionID contained within the message. If it does not, its KAF buffer creates an entry for that actionID. The two timers,

Figure 3.5: Overview of the different parts of the implementation and how they are connected.

---

4We can not use timers using real time clocks due them being unsynchronized with the simulation time.
3.3. Simulation scenario

$T_{F\text{-}Validity}$ and $T_{Forwarding}$, are calculated and events are added to the OMNeT++ event queue at corresponding times. The identifiers for the different events are stored in the KAF buffer.

Whenever a node is reached by a message with an actionID that it already has in its KAF buffer, it prolongs the time before rebroadcasting that message. To do this in OMNeT++, we have to keep track of the unique identifier in each OMNeT++ event created by us. We store this unique identifier alongside its corresponding DENM within the KAF buffer. When the message handler in the network layer passes us a message we go through the node in question’s entire KAF buffer to see if we are to update and back off on sending out our copy of the received message, or to add it to the buffer and start timers, as specified in Figure 2.5. If there is an entry for the received message within the KAF buffer, the node goes through OMNeT++ event queue and cancels the event scheduled for when to retransmit the message. It then creates a new event at a later time and stores the identifier for the new event. The node might have to update the timer controlling the lifespan of a message; this is done in the same way.

The last part of our implementation handles the two flowcharts seen in Figure 2.6. When OMNeT++ signals that an event is happening to a node it is again handled in the network layer which calls upon the matching function in the application layer. There we make sure that the message is matching the event scheduled for either $T_{Forwarding}$ or $T_{F\text{-}Validity}$. If it matches the validation timer, which determines for how long to keep a message in the buffer, we go through OMNeT++ event queue and cancel the event for $T_{Forwarding}$. When the event comes from $T_{Forwarding}$ timing out it means that we are to broadcast the message, if we are within the relevanceArea of the event the DENM is regarding. The distance to the event from the node is calculated, if the node is within the specified area it broadcasts the message and schedules a new event for the next broadcast. If we are not within the specified area, a new event is not created and no message is broadcast.

3.3 Simulation scenario

We use a motorway as location for our simulations. Motorways have several nice properties like free line of sight, similar speeds among vehicles and (at least outside of cities) few other networks that can interfere with VANETs. This gives a good chance of messages to be delivered.

![Simulation scenario of a motorway.](image)

Figure 3.6: Simulation scenario of a motorway.
3.3. Simulation scenario

We performed 40 simulations, 20 without an attack and 20 with an attack, each lasts for 75 seconds long. There are 32 vehicles in movement driving in both directions on the motorway as seen in Figure 3.6.

To make our measurements reliable we have to trace how a DENM behaves normally. This was done by simulating that a vehicle encounters a problem, the blue vehicle in our case, which sends out a DENM at the same time as it starts to decelerate. The blue vehicle will do this after it reaches a predetermined position. The DENM that is sent out should be stored in the KAF-cache of the vehicles within the relevanceDistance of the event before starting their timers $T_F_{Validity}$ and $T_F_{Forwarding}$ for the received DENM. In Table 3.1 values for the DENM can be seen.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventPosition</td>
<td>516:26</td>
</tr>
<tr>
<td>validityDuration</td>
<td>600 s</td>
</tr>
<tr>
<td>transmissionInterval</td>
<td>200 ms</td>
</tr>
<tr>
<td>relevanceDistance</td>
<td>500 m</td>
</tr>
<tr>
<td>KAF</td>
<td>enabled</td>
</tr>
</tbody>
</table>

Table 3.1: Parameters for the DENM from the blue vehicle

For the scenario, we recorded the following:

- The delay for the blue vehicle to reach the other vehicles with a DENM
- Which vehicles received the DENM from the blue vehicle
- The number of total copies of the blue vehicles DENM a vehicle received
- The number of times the vehicles sent or forwarded a DENM
- The total amount of received DENMs

With this data we are able to measure possible delays introduced by an attack. This is calculated from when the blue vehicle has sent out its DENM to when a vehicle for the first time receives it. We are also able to calculate the packet loss by comparing the number of copies of the blue vehicle's DENM received by other vehicles with and without an ongoing attack. If the attack does not induce packet loss the amounts of received copies should be the same.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventPosition</td>
<td>-</td>
</tr>
<tr>
<td>validityDuration</td>
<td>10 s</td>
</tr>
<tr>
<td>transmissionInterval</td>
<td>500 ms</td>
</tr>
<tr>
<td>relevanceDistance</td>
<td>500 m</td>
</tr>
<tr>
<td>KAF</td>
<td>disabled</td>
</tr>
</tbody>
</table>

Table 3.2: Background DENMs parameters

Since we want a realistic scenario we want some activity in the network. The activity is created by vehicles broadcasting random DENMs\(^5\). See Table 3.2 for the parameter values of the background DENMs. As a comparison CAMs, the other message type specified at the application layer level, are sent out by each vehicle 10 times per second. This means that we have less noise than one can expect in a real scenario.

\(^5\)The probability to send out a DENM is $\frac{2}{100}$ each time the vehicles position is updated.
### 3.3.1 Flooding of the network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>eventPosition</td>
<td>516:26</td>
</tr>
<tr>
<td>validityDuration</td>
<td>1000 s</td>
</tr>
<tr>
<td>transmissionInterval</td>
<td>2 ms</td>
</tr>
<tr>
<td>relevanceDistance</td>
<td>5000 m</td>
</tr>
<tr>
<td>KAF</td>
<td>enabled</td>
</tr>
</tbody>
</table>

Table 3.3: Flooding DENM parameters

To flood the network the red vehicle will continuously create and broadcast DENMs with unique actionIDs. Unique actionIDs ensure that the DENMs get individual entries and timers within KAF. In our simulations the red vehicle create 307 of these unique DENMs. To trigger the forwarding of the DENMs often it is important to set the transmissionInterval to a very small value, in our case 2 milliseconds, which gives a short $T_{Forwarding}$ timer value. RelevanceDistance is set it to a very high value, see Table 3.3, since we want all vehicles in the simulation to be within the affected area and therefore also forward the DENMs used to flood the network. ValidityDuration is set to an arbitrary value larger than the simulation length.
This chapter presents the simulation results of our work.

4.1 Vehicles reached by non-attack messages

In our simulation running with an attack, the amount of vehicles that did not receive the DENMs from the vehicle getting a problem (also know as the blue vehicle), or copies of it, was 17.5 percent of the total amount of vehicles, aggregated over all simulation runs. When no attack was ongoing, all vehicles received the blue vehicle's DENM.

![Figure 4.1: The amount of vehicles reached by a DENM.](image)
4.2 Delays

The attack introduces a severe delay to reach other vehicles. Figure 4.2 above depicts the average number of reached vehicles over time. The time to reach half of the vehicles is increased by a factor of over 2000.

During the first few milliseconds a majority of the vehicles are always reached in the reference runs. The seemingly large standard deviations come from the fact that the original broadcast from the blue vehicle always reaches 21 of the vehicles but the time it took to reach them in different runs was separated by a few micro- or milliseconds. During an attack the blue vehicle tries to broadcast messages in a network that is being pressured by our Denial of Service attack. This makes it very hard for messages to be propagated to an actual recipient. This is clearly seen, as the gray line in the start is significantly lower than the reference. In some cases few vehicles are reached but after a second the average is that five vehicles are reached. As time passes the blue vehicle's message reaches more vehicles until almost all of them are reached, however most vehicles get the blue vehicle's DENMs with a delay of over ten seconds. In a traffic situation this delay could potentially be disastrous. The difference in total vehicles reached, with and without an attack, after 25 or more seconds is due to the fact that, on average, not all cars receives the message with an attack, see Figure 4.1.
4.3 DENM Volume

When no attack is ongoing the average car sends out about 16 messages and receives slightly above 300 messages. This is not the case when the 307 attack DENMs sent out by the attacking vehicle are copied and rebroadcast by KAF thousands of times. This creates a huge increase in DENMs sent and received as seen in Figure 4.3. The increase in received DENMs is close to 15 500 percent. The increase of sent DENMs is almost 190 000 percent. This puts an enormous strain on the network and induces huge packet loss (see Figure 4.5). Due to the back-off timers that reset when a vehicle receives a copy of a DENM the standard deviation is quite large for the attack runs, however, the average is still several magnitudes higher during an attack than in the reference scenario.
4.4 Packet Loss

The two graphs in this section depict the packet loss. The first graph (Figure 4.4) show the amount of packets lost that are not messages used for flooding, i.e. they are the DENMs sent out by the blue vehicle. The second graph (Figure 4.5) show the total packet loss in the attack scenario. Losing attack messages is not a bad thing by itself but it visualizes the extreme strain the attack puts on the network.

![Figure 4.4: Number of copies of the blue vehicle's DENM](image)

As seen in Figure 4.4, copies of the blue vehicle's DENM are clearly lost. The packet loss is, on average, above 60 percent during an attack. This happens due to the vehicles receiving or sending other messages at the same time as they are trying to receive. This occurs very frequently when the network is under the pressure of the attack. The difference of the number of received copies of the blue vehicles DENM in the reference runs comes from the random delay when calculating $T_{Forwarding}$. 
In Figure 4.5, the total packet loss is depicted. In the reference runs, all vehicles receive all of the messages they are within range of. When an attack is ongoing this is not the case.

A vehicle should on average reach 18 other vehicles in our topology with its DENMs. Using this we are able to calculate how many DENMs should have been received in total and therefore calculate the packet loss. When an attack is ongoing a message on average reach roughly 1.5 other vehicles. This gives a total packet loss of over 90 percent.
This chapter discusses the result of the attack and other possible interesting simulation scenarios.

5.1 Outcome of the attack

Our attack drastically increases the network traffic using the properties of Keep-Alive Forwarding. Creating large delays and high packet loss makes it impossible to rely on the network for vehicle communication. This means that the attack has to be deemed as a success. KAF enables us to amplify our attack using the other vehicles as what could be viewed as bots in a botnet, without ever having to compromise them.

5.2 Number of vehicles in simulation

The overload of the network would likely increase further if more vehicles are on the road and within reach of the attacking vehicle. The more vehicles there are, the bigger network traffic the attack will produce. This means that more vehicles will try to forward malicious DENMs and create an even smaller window for the blue vehicle's messages to be received. Having more attacking vehicles would also create smaller windows for the blue vehicle's DENMs to be sent due to more attack messages being scheduled for forwarding.

5.3 Number of simulations

We have run 20 simulations with an attack and 20 without an attack. We believe this would give enough data to get reliable results. With each attack simulation taking close to five hours to complete, pushing total simulation time well above 100 hours, made more runs was unfeasible.

5.4 Simulation length

The simulation time of 75 seconds was chosen because it is a rather long time in a motorway scenario, a car driving at 120 km/h travels 2500m in 75 seconds. The high driving speeds
also gives drivers less time to react to events than in other road scenarios. Due to this, an increased delay, or even worse, DENMs not being delivered, could be lead to a disaster that might be fatal. Having a longer simulation scenario would give the blue vehicle’s DENM longer time to possibly reach other vehicles. However, in our data there was no indication of that actually happening and with the time to run each simulation growing exponentially for each second longer we simulate, it was a trade off we chose to do.

5.5 Our implementation within a WAVE stack

Our implementation of the standard is built within a WAVE stack. This means that our implementation is not built in the way ETSI might have intended for the ITS standard and also means that we do not have access to some of the features in the facilities layer of the ETSI ITS stack, e.g. GeoCasting. As all layers in a network stack are independent of each other, we do not see this as something that makes the results less reliable.

Within ETSI ITS G5 there is a congestion avoidance mechanism called Decentralized Congestion Control (DCC). We chose not to implement DCC because its main components reside at MAC layer level, which was out of scope. Parts of the congestion control can be implemented using an ETSI facilities layer, however, this was out of scope as well. This is something that can be viewed as a weakness if our study, however, there is research evaluating DCC which points out that the mechanism have several flaws which cause rapid state switching, low predictability and large oscillations in transfer rates [1]. An attack like this with DCC implemented would be interesting in a further study to see how it is affected.

5.6 Choice of the road to simulate

The choice to simulate on a motorway, where there are very few other networks that can interfere, is due to the fact that we wanted to give the blue vehicle’s DENM a good chance to be received by other vehicles. By simulating on urban streets, with other networks and buildings interfering would worsen the situation for the blue vehicle’s DENM to be received. An urban scenario would also make it more difficult for the attacking vehicle to reach other vehicles directly. However, due to the aggressive forwarding parameters, other vehicles would help with reaching the originally unreachable vehicles, and would be an interesting scenario to look at given the time.

Due to the randomness of which lane that a vehicle is placed on, the distances to the blue vehicle may differ between simulation runs. It can also lead to the blue vehicle being slightly delayed to the position where it sends out its DENM. Due to these factors, vehicles may be very close to the blue vehicle when it sends out its DENM. The closer a vehicle is to the sender, the higher probability it has to receive messages from the sender.

5.7 Errors within the standard

There are several spelling errors in the documents defining ETSI ITS G5 along with more severe oddities such as variables used in flowcharts not being described elsewhere. In a flowchart regarding the forwarding scheme a variable called *terminationInterval* is mentioned, however, this variable is not described elsewhere. In cases like this we generally skipped the conditions involving the unmentioned variable [9].

5.8 Why hacking is a good thing

Hacking is a very sensitive subject in our society due to privacy and safety issues. It is, however, a great tool for researchers to use when testing if something is secure. When a new, big, infrastructure is released, it affects a lot of people in their daily life. Therefore it
is of uttermost importance that it is safe. When researchers perform and design attacks it is to ensure that we have a safe society. By publicly publishing results, preferably after the particular issue has been fixed, it drives awareness towards possible problems and enables solutions to be created.

Vulnerabilities presented before they are fixed creates a back with a target. This causes one to consider factors as how widespread the technology is, potential incentives and what damage can be caused. Weighing in if the vulnerability focuses on a very specific mechanism or a larger conceptual problem should also be taken in account. Technologies where a security flaw is exposed face large amounts of negative press\textsuperscript{1} and a presumed lower adaption rate.

Self-driven vehicles will be upon us in the near future, and will definitively affect almost all of us. By having a standard regarding how these vehicles will interact and communicate is important because it enables better platooning and higher traffic flows. It also ensures that vehicles of different brands can cooperate.

By testing how the standard reacts to a flooding attack enables us to draw conclusions regarding the possibilities of performing attacks that severely limit the effectiveness of the system, and possibly even create life threatening situations for the people driving vehicles using the standard.

\textsuperscript{1}Hacks of TeamViewer and Dropbox happened in the months of writing this thesis.
In this thesis, we explored the possibility to use KAF to perform a Denial of Service on VANETs. ETSI has released a standard for IVC that has KAF as a way to reach vehicles further away while trying to keep the network traffic down. To test this we created a simulation scenario based upon a motorway. We used a single attacking vehicle to inject DENMs with forwarding settings that makes the other vehicles forward the messages very often. Another vehicle gets a problem and broadcasts DENMs that we trace. To assess the efficiency of our attack we recorded the delay for the problem vehicle’s DENM to be received. The packet loss was also recorded. We have not found any studies exploring the possibility of using KAF to perform Denial of Service attacks in VANETs. Due to the fact that the other vehicles, unknowingly, help forwarding the spam messages, the attack is amplified to the point were the VANET becomes impossible to rely on. Therefore we conclude that the Keep-Alive Forwarding mechanism can be abused to overload the network with malicious DENMs.

The major cause of forwarding attack is the freedom in the choice of parameter values within the messages, and especially the forwarding parameters used in Keep-Alive forwarding. An attacker can simply select values which cause excessive forwarding, in an excessive area, for an excessive amount of time. This freedom in parameter values, call it configurability or generality, is something that is sought after in solutions due to portability and maintainability. However, we believe that when it comes to protocol design simplicity and robustness must take higher priority.

6.1 Future work

In our thesis we explored how KAF-enabled DENMs behave when implemented on as an application layer within a WAVE stack. Testing this type of attack within a full ETSI stack would significantly strengthen the conclusions. Also testing the attack in other road scenarios would also solidify the results.

Our implementation only uses a single scheme for broadcasting. There are other schemes that could have an impact on the results and therefore be better or worse with KAF. Evaluating performance of different schemes is possible to do in the simulation tools we used.

An open problem that also remains is to secure KAF from attack like the one presented in this thesis.


