Renan Domingos Merlin Greca

Trust management for vehicular networks

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Luiz Carlos Pessoa Albini.

Curitiba-PR
2017

Resumo

À medida em que computadores tornam-se menores e mais poderosos, a possibilidade de integrá-los a objetos do cotidiano é cada vez mais interessante. Ao integrar processadores e unidades de comunicação sem fio a veículos, é possível criar uma rede veicular ad-hoc (VANET), na qual carros compartilham dados entre si para cooperar e criar ruas mais seguras e eficientes. Uma solução decentralizada ad-hoc, que não depende de infraestrutura pré-existente, conexão com a internet ou disponibilidade de servidores, é preferida para que a latência de entrega de mensagens seja a mais curta possível em situações críticas. No entanto, assim como é o caso de muitas novas tecnologias, VANETs serão um alvo de ataques realizados por usuários maliciosos, que podem obter benefícios ao afetar condições de trânsito. Para evitar tais ataques, uma importante característica para redes veiculares é gerenciamento de confiança, permitindo que nós filtrem mensagens recebidas de acordo com valores de confiança previamente estabelecidos e designados a outros nós. Para gerar esses valores de confiança, nós usam informações adquiridas de interações passadas; nós que frequentemente compartilham dados falsos ou irrelevantes terão valores de confiança mais baixos do que os que aparentam ser confiáveis. Este trabalho propõe um modelo de gerenciamento de confiança no contexto de trajetos diários, utilizando o *Working Day Movement Model* como base para a mobilidade de nós. Este modelo de movimentação permite a comparação entre VANETs e redes sociais tradicionais, pois é possível observar que pares de veículos podem se encontrar mais de uma vez em diversos cenários: por exemplo, eles podem pertencer a vizinhos ou colegas de trabalho, ou apenas tomar rotas similares diariamente. Através de repetidos encontros, uma relação de confiança pode ser desenvolvida entre um par de nós. O valor de confiança resultante pode também ser usado para auxiliar outros nós que podem não ter uma relação desenvolvida entre si. O algoritmo proposto é baseado em um já existente, que foi desenvolvido para redes centralizadas e focado em modelos ad-hoc estáticos; o algoritmo anterior será adaptado para servir uma rede decentralizada e dinâmica, que é o caso de VANETs. Usando valores de confiança existentes, um grafo direcionado é modelado, no qual arestas representam a relação de confiança entre os pares de nós. Então, componentes do grafo são formados, de forma que nenhum par de nós em um componente tenha uma relação de confiança negativa. Um algoritmo de coloração de grafo é usado no grafo de componentes resultantes e, usando os resultados de coloração, é possível inferir quais nós são considerados maliciosos pelo consenso da rede. Espera-se que o algoritmo completo final seja rápido, para que ele possa ser executado frequentemente, e permita que nós mantenham um modelo da rede ao seu redor indicando quais nós vizinhos podem ou não ser confiados.

Palavras-chave: redes veiculares, gerenciamento de confiança, identificação de nós maliciosos.

Abstract

As computers become small and powerful, the possibility of integrating them into everyday objects is ever more appealing. By integrating processors and wireless communication units into vehicles, it is possible to create a vehicular ad-hoc network (VANET), in which cars share data amongst themselves in order to cooperate and make roads safer and more efficient. A decentralized ad-hoc solution, which doesn't rely on previously existing infrastructure, Internet connection or server availability, is preferred so the message delivery latency is as short as possible in the case of life-critical situations. However, as is the case with most new technologies, VANETs will be a prime target for attacks performed by malicious users, who may benefit from affecting traffic conditions. In order to avoid such attacks, one important feature for vehicular networks is trust management, which allows nodes to filter incoming messages according to previously established trust values assigned to other nodes. To generate these trust values, nodes use information acquired from past interactions; nodes which frequently share false or irrelevant data will have lower trust values than the ones which appear to be reliable. This work proposes a trust management model in the context of daily commutes, utilizing the Working Day Movement Model as a basis for node mobility. This movement model allows the comparison of VANETs to traditional social networks, because it can be observed that pairs of vehicles are likely to meet more than once in several scenarios: for example, they can belong to neighbors or work colleagues, or simply take similar routes every day. Through these repeated encounters, a trust relationship can be developed between a pair of nodes. The resulting trust value can also be used to aid other nodes which might not have a developed relationship with each other. The proposed algorithm is based on a previously existing one, which was developed for centralized networks and focused on static ad-hoc models; the previous algorithm will be adapted to serve a decentralized and dynamic network, which is the case of VANETs. Using existing trust values, a directed graph is modeled in which edges represent the trust relationship between pairs of nodes. Then, graph components are formed in which no pair of nodes within a single component has a negative trust relationship. A graph coloring algorithm is used on the resulting components graph and, using the coloring results, it is possible to infer which nodes are considered malicious by the consensus of the network. The complete algorithm is expected to be fast, so it can be executed frequently, and will allow nodes to maintain a model of the surrounding networks indicating which neighboring nodes can be trusted or not.

Keywords: vehicular networks, trust management, malicious node identification.

Contents

List of Figures

List of Acronyms

DTN	Delay-Tolerant Network
GPS	Global Positioning System
LTE	Long-term Evolution
MANET	Mobile Ad-hoc Network
WDM	Working Day Movement Model
OBU	On-Board Unit
RSU	Road-Side Unit
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad-hoc Network
WAVE	Wireless Access in Vehicular Environments

Chapter 1

Introduction

As computers grow in power and shrink in size, more aspects of everyday life can be enhanced by adding processing units to common devices. While many of these applications focus on conveniences, such as home automation [McCole, 2016] (the collection of connected and smart devices is dubbed the Internet of Things or IoT [Morgan, 2014]), the integration of computers with other objects and devices can also be important to save time and save lives [Real-Time Innovations, 2014]. One way of achieving this is by adding computers and wireless transmitters to vehicles — such as cars, buses, and trains — so they can share data which may increase traffic efficiency or reduce the chance of accidents [Saini et al., 2015].

In 2013, an estimated 1.25 million people lost their lives due to traffic accidents globally [World Health Organization, 2013]. While this number has greatly reduced over the past decades [Johnson, 2010] thanks to better safety features (seat belts, air bags, ABS, etc.) and stronger laws (drunk driving, motorcycle helmets, speed limits, etc.), it may still rise as a major cause of death in the years to come [World Health Organization, 2015], so further actions are necessary. Furthermore, as the car population increases, congestions consume ever more time of the daily commuter, peaking at over 100 hours per year for the residents of Los Angeles, CA [INRIX, 2017].

Smart vehicles and vehicular networks are ways that technology can aid both of the aforementioned problems. Through the use of sensors and wireless communications, these vehicles are able to avoid accidents by alerting distracted drivers [Lee et al., 2004], or by knowing in advance another vehicle's position and speed [Hafner et al., 2011]. By communicating, they can also collaborate to offer driving and route suggestions, therefore reducing the possibility of traffic jams [Knorr et al., 2012].

When dealing with safety or traffic-efficiency applications, it is crucial that network communications occur with low latency (approximately 100 milliseconds [CAMP Vehicle Safety Communications Consortium, 2005]). Current cellular technology, such as LTE, could be used to connect vehicles to the Internet, but the delay added by the transmission would make safety applications unfeasible or unreliable [Mangel et al., 2010]. Cellular connections also have other problems: the connection would require an active subscription with a carrier; the connection depends on available infrastructure; the wireless frequency would be

shared with phones and other mobile devices, increasing the possibility of interference and congestion; server-side issues could impact the vehicles' communications.

For these reasons, ad-hoc solutions are preferred over centralized ones. An ad-hoc network is one that has no reliance on pre-existing infrastructure (such as routers or access points) [Wu and Stojmenovic, 2004]. Instead, each node is able to communicate directly with others and a routing protocol allows for messages to be forwarded until they reach their destinations. Every time a node wants to send a message and the recipient is not a direct neighbor, it must choose which nearby node is the most likely one to get the message to its destination. Routing techniques can use either the network's topology or geographical coordinates [Saini et al., 2015] to choose which node should be the next hop.

These issues — the additional safety and efficiency as well as the low-latency communications — can be tackled through the use of a vehicular ad-hoc network (VANET), in which vehicles share data amongst themselves without relying on external devices, an Internet connection or server availability. Neighboring vehicles can share their position and velocity data at high frequencies, allowing, for example, for autonomous vehicles to plan a platooning approach to traffic [Amoozadeh et al., 2015]. In the case of a collision or other event, nearby nodes can broadcast alerts, which other nodes pick up and forward [Li and Wang, 2007]. That way, an alert can travel long distances in little time, allowing approaching vehicles to safely slow down or pick alternative routes.

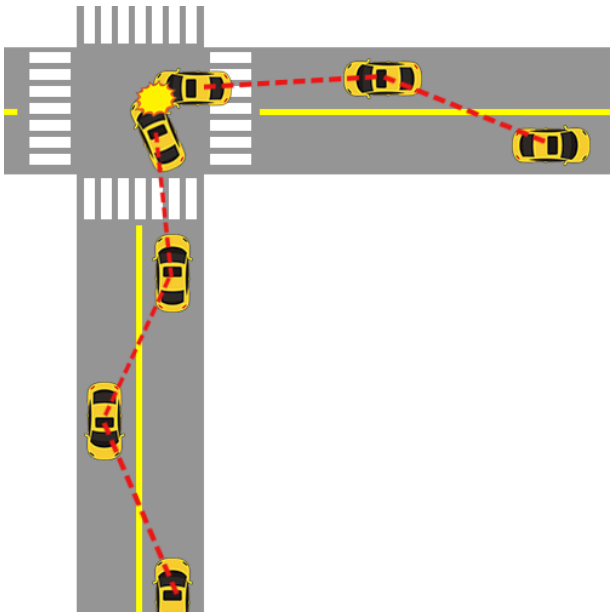


Figure 1.1: Propagation of a collision alert in a VANET

As is the case with most new technologies, VANETs are expected to be a notable target of attacks for a diversity of reasons [Isaac et al., 2010]. A local malicious user might alter the data his or her vehicle broadcasts in order to manipulate traffic conditions, while remote attackers could invade vehicles' computers and obtain partial control of the network [Garip et al., 2015].

These attacks can vary from time-consuming annoyances to life-threatening, so it is important that real-world implementations of vehicular networks are prepared to handle them.

In ad-hoc networks, one way to mitigate a number of attacks is through each node using data collected from previous experiences to filter out incoming messages that seem to be malicious, incorrect, or irrelevant. A node's degree of confidence that some data is correct and useful is called *trust*. For instance, in the example of a single malicious user broadcasting false data, nodes receiving these messages can use their own sensors to verify whether or not the data was correct, and update the *trust value* of the sender vehicle. In case the trust value of a sender is too low, a receiver node can choose to ignore the data contained in a message, as it concluded that the sender is not trustworthy. Trust allows for better cooperation of nodes in a network, since incorrect messages might be detected and discarded.

Furthermore, once nodes form their own opinions about others, they can propagate pre-existing trust values when necessary. For example, if two nodes are not direct neighbors and do not have any pre-existing trust information about each other, they can ask intermediary nodes for their opinions on the other node [Wang et al., 2009]. The management of trust values (i.e. how one node acquires and updates trust values) and the use of these values to derive further information (such as designating nodes as malicious or not) is called *trust management* [Ma et al., 2011]. The effective use of trust management allows for the detection of malicious, misbehaving or faulty nodes in the network, and for such information to be shared amongst the benign participants. Throughout this document, trust and trust management may be used interchangeably.

While the detection of incorrect nodes and/or messages is an important aspect of security and safety in vehicular networks, it does not address all of the problems. Trust solutions are not viable without a solid identity verification scheme, for instance, since nodes would not be able to form trust opinions without being sure of the others' identities (these schemes often use a Public Key Infrastructure [Wasef et al., 2010]). They also do not address issues such as driver and passenger privacy when using the facilities of a VANET. Furthermore, cryptography must be used in order to guarantee that the secrecy of messages is not violated. Therefore, trust and trust management must be viewed as an important aspect of vehicular ad-hoc networks, but not as a definitive solutions to all of the related concerns.

In order to study the implications of trust in vehicular networks, it is interesting to first take a closer look at trust in other kinds of networks. VANETs are a subset of technological networks, therefore it is useful to consider how the Internet or other ad-hoc networks handle trust. Furthermore, VANETs contain several features often found in social networks, which can be directly tied to how nodes form and develop trust relationships with each other.

This work proposes a method for identifying malicious nodes in a vehicular network, although it may also be viable for other dynamic and decentralized networks. The solution is based on a previously existing algorithm [Vernize et al., 2015], which was limited to centralized networks. In order to adapt the algorithm to a dynamic and decentralized environment, it is

necessary to locate network features that allow for trust relationships to be developed amongst the network members over time, since there is no centralizing agent to store and process the whole network's trust values. These features are found by tracing analogies to social networks, i.e. identifying how and why two nodes (in the case of VANETs, vehicles) would meet more than once and at a somewhat predictable interval.

1.1 Document organization

The remainder of this document is organized as follows. Chapter 2 explains the broad study of complex networks and the importance of trust in technological and social networks. Then, chapter 3 goes into details regarding VANETs and the importance of trust solutions in the field, presenting previous studies made on the subject. Finally, **??** shows the fundamentals of this proposed work, including the similarities found between VANETs and social networks, a realistic movement model that may be used for simulations, the previously existing study on malicious node identification for complex networks, and the important aspects that must be adapted to the dynamic vehicular environment.

Chapter 2

Complex Networks

Complex networks can describe many systems which are observed in nature and society through a collection of *vertices* (or *nodes*) and *edges* [Newman, 2010]. They can be comprised of palpable components (such as computers and cables), somewhat abstract entities (such as the World Wide Web's collection of webpages and URLs), or both (like the people and relationships that form a social network). The study of networks is tied to graph theory, since graphs are a useful way to generate models of networks, and therefore many concepts of the two fields overlap. Mathematical, computational and statistical concepts developed for graphs can be translated to be useful for a variety of different types of networks, like, for example, finding articulation vertices in a graph to locate bandwidth bottlenecks in a computer network.

Complex networks are generally divided into four categories [Newman, 2010]:

1. **Technological Networks** are grids purposefully engineered to provide services to consumers and/or citizens. The primary examples of these networks are the Internet, the telephone network, power grids, transportation and delivery networks. A commonly studied type of technological network are Mobile Ad-hoc Networks (MANETs). Although not of widespread use, MANETs can provide a way to create a network without pre-existing infrastructure, as long as each device is equipped with the proper hardware and software. Trust issues in technological networks and MANETs are detailed in section 2.2. VANETs, which are special types of MANETs, are introduced in chapter 3, along with several details regarding trust in those types of networks.

2. **Social Networks** are formed of relationships between people, or groups of people. These relationships can be familiar, friendships, acquaintance, etc. For the purposes of this work, the most relevant type of relationship is that of trust. The details surrounding trust relationships in social networks are shown in section 2.1.

3. **Information Networks** are the ones in which nodes are pieces of data or information and the edges are the connections between those pieces. Often, information networks are directly associated with technological or social networks. For instance, while the World

Wide Web is an information network (in which the nodes are webpages and the edges are the links that users click on to navigate), it relies on the Internet, as it contains the physical infrastructure that makes the web possible. Online social networks can also be classified as information networks, since their nodes are actually information about people rather than the people themselves. Trust in information networks can be observed in some instances, like peer-to-peer networks [cite], although its usages are not relevant for this work.

4. **Biological Networks** are the networks found in nature. Their nodes can be chemicals, cells, animals, groups of lifeforms, and more. An example is the brain, which contains a neural network formed by neurons, cells which enable information processing; connections in the network represent signals that are sent from one neuron to another. Another instance of biological networks are food chains, categorized as ecological webs. Species of animals are the nodes, while the predation of other species form the edges. In biological networks, it is difficult to clearly define trust, since nodes may not have any sort of awareness or intelligence (such as cells or proteins). Regardless, the study of trust in biological networks is not relevant for this proposal.

In most networks, trust can be a useful tool to aid the security and safety of its members. Therefore, the study of the concept and applications of trust is an important part of the study of networks.

Trust is a concept studied in fields such as psychology and economics, with specific definitions. In complex networks, under the perspective of computer science, it is a measure of one entity's confidence that another will behave properly and provide valid and/or meaningful data [Sherchan et al., 2013]. What follows is the basis of how a network can be modelled using a graph and how a trust model can be applied to it.

Consider an undirected graph $G = (V, E)$, which models one complex network of any kind. The vertices are the members of the network (computers, humans, etc.) and each edge represents a pair of vertices' ability to exchange data freely. This graph represents the network's *topology*, that is, the basic structure of the network. Then, there is also a directed graph $T = (V, O)$, called a *trust graph*. T contains the same nodes as G, although its edges represent the degree of trust (or opinion) each node has towards another.

There are two main ways to describe the edges in O: they can be binary, either existing when there is trust or not otherwise, or they can hold a specific trust value within a certain range. This means the shape of T is not necessarily similar to that of G. For instance, two people can have contact with each other but not maintain a trust relationship, thus altering the layout of the trust graph compared to the network topology.

In the following two sections of this chapter, trust in the contexts of social and technological networks is further explained. Both types of network can be fitted into the model above, but contain distinct features that demand closer examination. Furthermore, features of

both are relevant when analyzing network structure and trust graphs for vehicular networks, which is expanded upon in chapter 3.

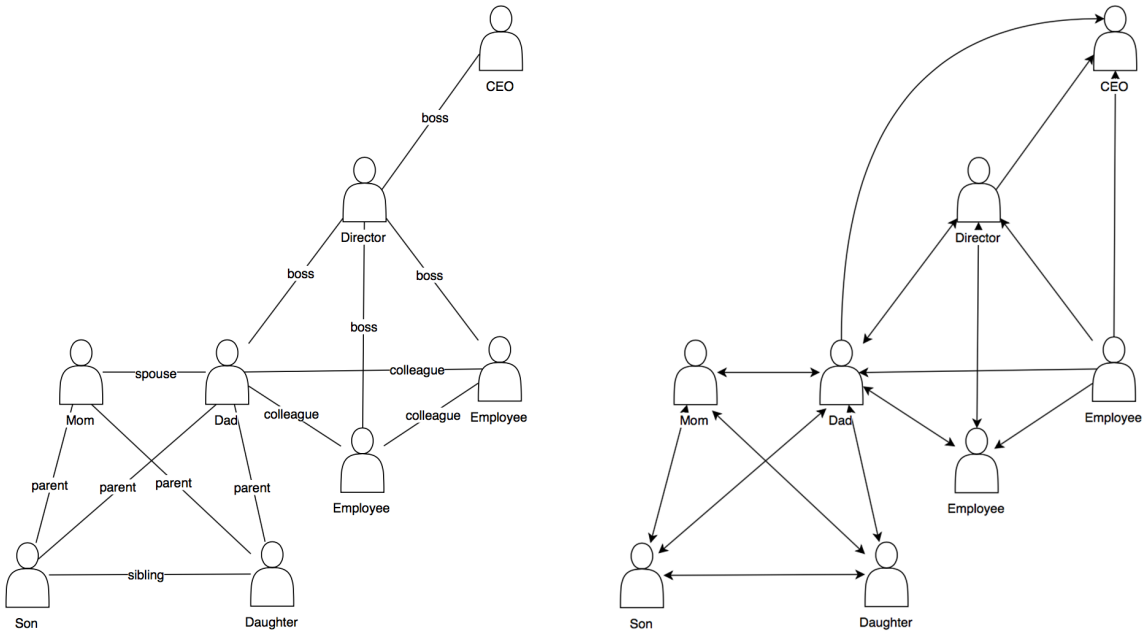
2.1 Trust in Social Networks

There are two types of social networks: real-world ones formed by relationships between people, and online ones that attempt to abstract the former into a digital environment. Examples of the first one are all around, present in any family, workplace, school or group of friends [Newman, 2010]. Online social networks started by connecting people who already knew each other and giving them an additional form of interaction (analogous to what telephones and email did before), but, today, it is not unusual for people to form relationships with others whom they have only met online.

In a traditional social network, it is simple to perceive how trust is relevant and how it works, since trust relationships between people are used on a daily basis to make decisions. When adapted to a digital environment, these social relationships can be used to automatically increase the relevance of certain information. For instance, upon reading an online review for a certain product, a user will be more likely to accept the review's conclusion if it was written by a close friend than if it were written by a stranger. Social trust is a way of estimating how much a certain recommendation will lead to a positive outcome [Golbeck and Hendler, 2006].

The absence of trust or the presence of distrust have consequences as well. Both in the real world and online, information which comes from a stranger is received with uncertainty; there is no reason to trust the sender, so the data itself must be analyzed and compared to other sources in order to judge whether or not it may be trusted. When one person actively distrusts another (that is, the person believes the other is malicious or uninformed), receiving data from the untrustworthy source will be actively avoided. In online social networks, for example, one user can "block" another in order to avoid seeing anything from the other.

Social networks also have the property of carrying trust from one relationship to another: information shared by a close friend of a person might be considered almost as trustworthy as some collected by the person him or herself. Therefore, it is possible to model social trust relationships as a graph, in which nodes represent people and edges represent a certain degree of trust [Newman, 2010]. Expanding on that property, there is the concept friends of friends [Boissevain, 1974]. If, for example, nodes A and B have mutual trust and are considered friends, then it is reasonable to assume that some of A's trust for B carries over to other nodes that enjoy mutual trust with B. In other words, a friend of a friend can be considered more trustworthy than the average stranger. This property is similar, although not identical, to transitivity, since trust is diminished for each extra step an origin node needs to reach a destination, and there is also the possibility that one node distrusts another even if they share a mutual friend. Naturally, social trust is not commutative (A trusting B does not imply that B trusts A).



(a) Topology graph (b) Trust graph

Figure 2.1: Example of a topology graph and a trust graph in a social network.

In general, social networks' topology and trust graphs are mostly static. Although friendships are formed and ended frequently (i.e. the topology is dynamic), those connections do not disrupt the general shape of the network, because members of the network will usually have other friends whose relationships remain stable. Even if a certain person's trust integrity is compromised due to a specific incident, that person's friends are not necessarily deemed untrustworthy, preserving part of the trust graph. While positive trust is often tied to the social topology, it is not always the case: one example is two work colleagues who may have a professional relationship, but wouldn't trust each other on other matters; another is the trust people place in authority figures without necessarily having met. Figure 2.1 shows an example of a small social network containing a family and an office.

In section 4.2, the argument is made that VANETs can be considered social networks in several occasions and how this can be used to develop trust in vehicular networks.

2.2 Trust in Technological Networks

In conventional technological networks, such as the Internet, trust is defined and applied quite differently from social networks since, generally, it is very centralized through services that offer security to users. Examples of this can be an IP filtering scheme to avoid distributed denial of service (DDoS) attacks or web browser extensions that block requests to domains in a blacklist; a central agent, be it a hosting provider or the extension's publisher, must maintain and update a list of untrustworthy IP addresses or domains. This means that, in the context of the Internet, trust is often derived from a secondary source: end users and their computers can't be expected to

maintain their own blacklists, so they rely on external parties which may provide these lists along with other security services. Similarly, when a user visits an e-commerce website, they must have some degree of trust on the website or the vendor; in this case as well, third-party services are used to certify the legitimacy of the transaction, based on feedback from other customers.

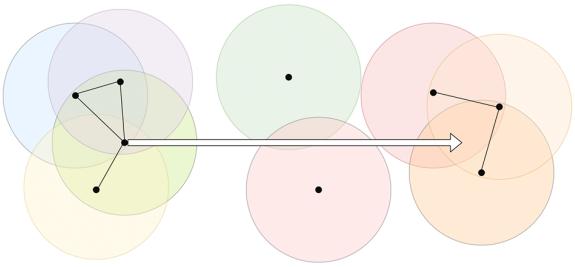
While the centralized trust solutions above serve their purpose on the security and privacy of Internet users, they would be too slow to be viable in a dynamic ad-hoc network, which cannot rely on a back-end infrastructure to distribute those lists. The most common instance of mobile ad-hoc networks, or MANETs, is using mobile devices, such as smartphones, being carried by humans. Although these networks are dynamic, their mobility is relatively low in relation to the wireless range of the devices — if two people are walking in opposite directions, their phones may communicate for several seconds before they leave each other's range. Trust solutions for MANETs can use this property to their advantage, since it allows one node to test another and check several of the messages sent between them.

Ad-hoc networks require a decentralized approach to trust management; each member of a network has its own opinions about other members, and these opinions can change over time. For these opinions to be generated and updated, two nodes must have had previous contact with each other, or derive trust from a third, intermediary, node. Hence, there is the correlation between the trust graph and the topology graph of the network. Since MANETs are dynamic, the graph that represents one network's topology is frequently changing and, with that, the opportunities to create and update trust relationships also changes. In networks in which nodes can meet more than once, it might be valuable to store information from previous encounters to use in the future, although this process can be too slow or resource-consuming to be viable in certain devices; by doing this, the trust graph maintains edges between nodes that are no longer connected in the topology.

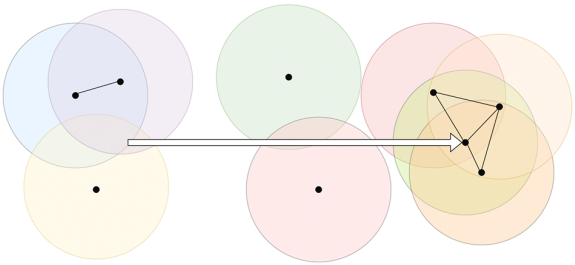
Another important aspect of trust in ad-hoc networks is that information is, generally, uncertain and incomplete [Baras and Jiang, 2005]. That is, since nodes form their own model and opinions of the surrounding network, it is unlikely that this data will be certain and accurate with reality. For this reason, it is also possible to use data gathered by neighboring nodes to complement the model. This data itself is subject to the trust evaluation of the neighbors, but it is crucial to better approximate the trust values of other nodes. Incompleteness is an inherit trace of MANETs, since it is entirely possible for nodes to be too distant to communicate, and only occasionally come into contact.

Finally, MANETs must consider the processing and battery limitations of the devices that integrate it. Nodes may disable wireless communications to save power and therefore become uncooperative, or it may be too slow to be a reliable source of information.

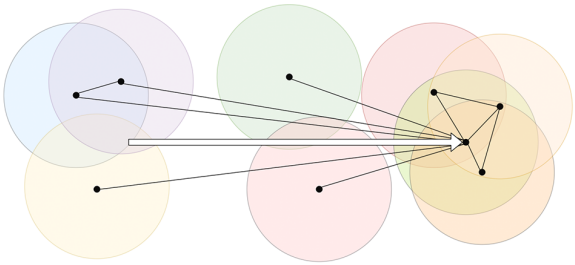
There are few examples of MANETs implemented for consumer devices. Two examples are networks created to quickly share data between devices using Wi-Fi or Bluetooth [Krochmal et al., 2014], or ones that allow for multiplayer gaming sessions amongst multiple nearby devices [Sasaki and Kuwahara, 2011]. In both cases, there is no need for a complicated



(a) Initial topology and trust graph



(b) Final topology graph



(c) Final trust graph

Figure 2.2: Example of the changes node mobility causes to the topology and trust graphs.

system-level trust model, since those activities involve active participation from the users wielding the device (that is, the user chooses whether or not to communicate with other devices); rather, the trust relationship occurs socially amongst the users themselves.

Naturally, VANETs are an instance of MANETs and therefore share some of the same features. However, the topology of vehicular networks is very different from standard ad-hoc networks, and possible trust solutions are accordingly also distinct.

Chapter 3

Vehicular Ad-hoc Networks

Today, most premium vehicles come equipped with hardware that allow for connectivity features; it is expected that, by 2022, many standard vehicles will also come with such features built-in, accounting for a substantial share of the automotive industry's revenue [Viereckl et al., 2016]. Although these features can be useful tools to aid drivers, reducing traffic and risk of accidents, they are merely a gateway to the long-term goal of truly autonomous vehicles, which might become a reality within the next decade; many automakers and technology companies have laid out their plans for the upcoming years [Stewart, 2016]. However, the proper functioning and utility of both connected and fully autonomous vehicles rely on technologies, protocols and applications that allow for the fast communication between vehicle's on-board computers.

Vehicular ad-hoc networks, which are a special instance of MANETs, are a much-studied solution to the problems in the way of smart and autonomous vehicles. In these networks, all nodes are related to traffic; they can be vehicles equipped with on-board computers, or stationary units placed near roads. By quickly sharing data with neighboring vehicles, without the need of an Internet connection, smart vehicles can alert their drivers of important road conditions [Barba et al., 2012], while autonomous vehicles can synchronize their movements to maximize traffic throughput [Amoozadeh et al., 2015].

Several current efforts to make VANETs viable in cities are centered around the IEEE 802.11p standard, also called Wireless Access in Vehicular Environments (WAVE) [Jiang and Delgrossi, 2008]. Among other aspects of the wireless technology, the WAVE standard describes two types of nodes for vehicular networks: on-board units (OBUs) and road-side units (RSUs). On-board units are computers placed within each vehicle which monitor the vehicle's data and are able to communicate with other nodes using wireless signals. Road-side units are placed in static locations near roads; they may also have wired interfaces with other RSUs and the Internet, so it is possible to use them as anchor points for Internet access for passing vehicles. When referring to the communication between two OBUs, the term vehicle-to-vehicle (V2V) communication is used [Yang et al., 2004]; when both an OBU and an RSU are involved, it is called vehicle-to-infrastructure (V2I) communication [Chou et al., 2009]. Although this nomenclature is important to understand other studies on the subject of VANETs, this study does

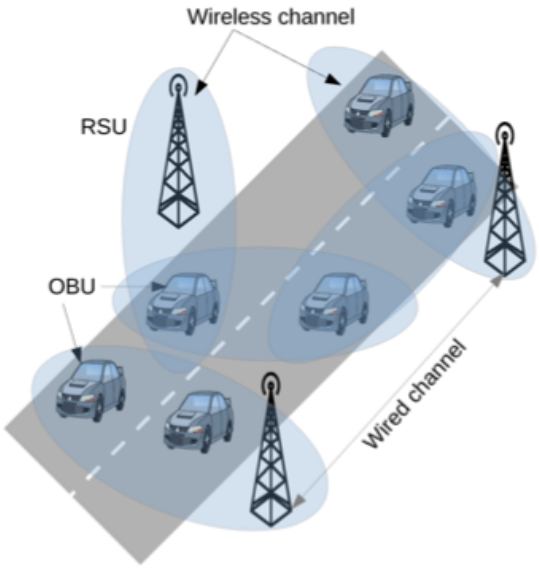


Figure 3.1: Basic elements of a VANET: OBUs and RSUs. [Saini et al., 2015]

not consider RSUs and focuses only on vehicles themselves as nodes, so references to VANETs and vehicular networks are exclusively tied to V2V communications.

In traditional networks (ad-hoc or not), routing protocols usually use the topology to choose where to forward packets; in other words, the primary metric used is the number of hops required to reach the destination. This metric is not as useful in vehicular networks, since the high mobility causes the topology to change frequently. Instead, most VANET routing protocols use geographical coordinates to forward packets [Saini et al., 2015], that is, the physical distance between two nodes is used as the primary metric. The implication is that, even if a packet requires more hops to reach its destination, it will always be traveling the generally correct direction.

As expected for a new technology being introduced, vehicular communications can become an appealing target for malicious users and attackers. These are some examples of possible issues in a VANET:

1. Vehicles with faulty GPS modules, speedometers or other sensors. If a vehicle is broadcasting incorrect data (perhaps unknowingly) because of a hardware or software fault, it can be a serious hinderance to efficiency and safety applications. It might behaving appropriately according to protocol, but the data it sends is not reliable [Isaac et al., 2010].

2. Vehicles might be deliberately broadcasting false data. In this case, there might be a specific purpose (by either the vehicle's driver or a remote attacker), like altering traffic or even cause an accident [Golle et al., 2004].

3. Attackers with control of several vehicles can propagate junk data in an attempt to flood the network, causing a distributed denial of service (DDoS) attack. Alternatively, the data propagated might have some reasoning behind it, like lying about road conditions in order to divert traffic [Garip et al., 2015].

4. Instead of sending data, some vehicles might try to eavesdrop on others' communications. The hop-based routing protocols used in VANETs facilitate this, since any node can be asked to be a hop. If the intermediary node is malicious, it may attempt to extract data contained in messages or refuse to forward them. Related to this, there is the Sybil attack [Isaac et al., 2010], in which a node lies about its position in order to seemingly alter the physical topology of the network and be chosen as a hop [Leinmüller et al., 2005].

5. Malicious vehicles may use signal jammers or other devices in order to affect other vehicles' sensors and communications [Isaac et al., 2010]. That can cause other vehicles to broadcast incorrect data, therefore obfuscating the origin of the attack.

6. A malicious user or remote attacker can monitor messages shared across the network in an attempt to stalk one specific vehicle [Isaac et al., 2010].

Each of these possible attacks requires a unique approach, though there are some broader ways to help the security and safety of VANET users. Trust, as is described in section 3.2, can be an important feature in vehicular networks, especially when attempting to filter out malicious or incorrect messages. It does not, however, avoid all possible attacks, such as a signal jammer or stalking. Rather, different mechanisms must be explored in order to avoid most problems.

3.1 Special properties of VANETs

VANETs feature several unique properties which distinguish them, and the behavior of its members, from other types of networks [Yousefi et al., 2006]. Some of these properties include:

1. Rapidly changing topology. Since the nodes are vehicles, they move frequently and at relatively high speeds. Each node's wireless communications also have a certain range, so the other nodes within that range (and, therefore, network neighbors) can change very quickly.

2. Node mobility is constrained to a pre-existing grid of roads. Within those roads, nodes usually travel in predictable directions according to local laws and historical data. The spaces in the grid, like city blocks, provide a challenge to communication both because of distance and because buildings can cause obstructions to radio transmissions.

3. VANETs are prone to fragmentation, since a gap in the network topology can make two parts of it unable to communicate with each other. Combined with the property above, this fragmentation can appear and disappear frequently, depending on the node density.

4. Due to the changing topology and possible disconnection, connection with distant nodes is not reliable. Therefore, the effective diameter of the network is relatively small for important applications.

5. Compared to devices like smartphones, vehicles have no notable power constraints.

6. In certain locations and/or moments, large vehicle density results in a large-scale network, since there are many nodes concentrated in a relatively small space.

7. The topology is susceptible to driver behavior. First, this means the topology can occasionally change in unpredictable ways. Second, contents of a message sent through the network can alter the driver's behavior and therefore change the topology.

Some of these properties provide advantages or disadvantages when developing trust models for vehicular networks, although all of them must be considered.

3.2 Trust in VANETs

Like in other types of networks, the proper functioning of a VANET depends on the reliability of the vehicles (nodes) which participate in it. If one node is malicious or faulty, it can spread incorrect data that may compromise the network's utility. Once the concept of VANETs was established, researchers have been attempting to predict ways in which malicious users might use the network to their advantage. Examples include triggering false alarms about inexistent accidents, lying about the average speed in one road to make it less desirable for others, and falsifying geolocation data to exploit location-based routing algorithms. Therefore, the concept of trust must be established in the vehicular network context, allowing for nodes to judge the validity of information transmitted by others and share those conclusions with other nodes.

There is an important distinction between a malicious node and a faulty one; both of them may be sharing false data, but for different reasons and with different consequences. For example, a malicious node may lie about its location in order to make routing protocols use it [Leinmüller et al., 2005], in order to try to store or alter messages, while a faulty GPS module may cause an accident because its position data was incorrect. However, that distinction can be hard to make, because a close inspection is necessary to determine whether the incorrect data is erratic or deliberate. Since both types of nodes are problematic to the proper functioning of a network, malicious and faulty nodes can be treated as the same in a trust model.

In general, trust management solutions for VANETs use *data-oriented trust*, *entity-based trust*, or a combination of the two. The solutions that use data-oriented trust (or *data-centric trust*) [Raya et al., 2008] focus on validating messages instead of entities. This is important when vehicles share messages about a specific event, such as a collision, which must be quickly validated by neighbors and distributed to other nodes within a relevant area. In this scenario, vehicles sharing the same road might be complete strangers to each other, and therefore would not have any trust relationship, so neighboring nodes must decide if a message is true by its contents and by other nodes' observations of the event. On the other hand, when dealing with frequent messages which contain basic information such as geolocation and speed (used for traffic-diminishing

solutions), it is too costly to judge each individual message. Therefore, *entity-based trust* becomes more appealing, since benign nodes can quickly identify a malicious node and isolate it from the network. Within entity-based trust, there are also two often-used methods of establishing trust: first, there is *role-based trust*, which is the static trust of pre-authenticated vehicles such as police units; second, there is *experience-based trust*, which is built through previous encounters shared between pairs of nodes. The model proposed in this work utilizes entity-based and experience-based trust, as it is based on the possibility of nodes meeting more than once and, therefore, being able to form a long-term trust relationship with each other.

3.3 Existing trust models for VANETs

Several models have been proposed to solve the problem of trust in vehicular networks. In this section, some of the most relevant ones are described, considering the time in which they were proposed, the advantages they bring and their contributions to later study. None of them provide a complete solution, but serve as pieces of a puzzle that is still incomplete. Many trust management solutions for VANETs have been proposed over the years, such as [Patwardhan et al., 2006], [Gerlach, 2007], [Raya et al., 2008], [Huang et al., 2010], [Ding et al., 2013], [Haddadou et al., 2013], [Liu et al., 2016], [Kerrache et al., 2016]. There are also some review and/or survey articles on the subject of VANET trust models, such as [Zhang, 2011], [Ma et al., 2011], [Zhang, 2012], [Mejri et al., 2014], [Soleymani et al., 2015] [Sengar, 2016], and [Dwivedi and Dubey, 2016].

[Dotzer et al., 2005] is one of the earliest examples of VANET trust models, establishing a system called VARS, based on the reputation of nodes and messages throughout the network. The authors use what they call *opinion piggybacking*, which means that, for each hop between the origin and the destination of an event-related message, the forwarding node appends its opinion of the message's contents. That opinion is formed using a combination of the forwarding node's own observations of the event, its opinion of the origin node and previous opinions appended to the message. This process adds credibility to a message through validation by nodes in a non-centralized fashion. It combines aspect of data- and entity- based trust, since nodes share their opinion of the data as well as their opinion of the sender. One interesting observation is setting higher trust values for certain vehicles based on their familiarity with the region (vehicles that reside in a given city may have more experience with certain types of events than newcomers). However, opinion piggybacking has its own share of problems. First, it allows forwarding nodes to access (at least some of) the contents of a message so it can form an opinion on it, diminishing privacy; a malicious forwarding node could even attempt to alter those contents. Second, using previously appended opinions from other nodes to form a new opinion causes the first nodes to forward the message to have a substantially greater impact over the final opinion than the later ones. Finally, there is an issue with scalability, since appending new information to a message on

each hop may add a significant overhead to the transmission. Additionally, the authors provide little to no experimentation or proof that their approach would be sound in a real-world network.

The model proposed in [Minhas et al., 2010] uses several criteria to judge whether or not a received message is trustworthy. First, nodes are classified by their roles and previous experience with them. Roles are used for vehicles which should be more trustworthy than the average: government official cars, traffic report vans, buses, cabs, etc. Nodes also store their experience each time an event message is received (if one neighboring node reported an event which did not turn out to be true, its trust value is reduced). Additionally, messages have higher reliability when their senders are closer in time and space to the reported event. When several messages about the same event are received, a node can either choose the n most trustworthy senders, according to the priority (fewer chosen nodes means a faster, but less precise, decision), or compute the majority opinion of the messages according to each sender's trust value. The model considers both role-based trust and experience-based trust; although the work proposed here does not use role-based trust, the authors provide a useful method of calculating and updating an experience-based trust value, which might be used or adapted. However, their model relies only on direct interaction between pairs of nodes, so no form of indirect trust (that is, trust values received from other nodes) is considered.

In [Chen et al., 2010], the authors propose to evaluate messages utilizing a cluster-based trust model. By separating nodes into clusters with their geographical neighbors, it is possible to efficiently distribute the evaluation of messages using previously formed opinions. When a node sends a message, one node in the cluster (the leader) must aggregate the other nodes' opinions on that message. Afterward, the message is only forwarded to another cluster if that aggregate opinion is above a certain threshold; furthermore, nodes that receive the message only act upon it if the overall trust on it is above another threshold, which can be different according to the nature of the message. However, it is unclear how the model behaves when the network is too sparse to form relevant clusters, neither do the authors inform how the aforementioned thresholds are decided. Furthermore, maintaining clusters in a highly dynamic network is a costly job and, if the cluster leader itself is malicious, all the information from that cluster becomes untrustworthy.

The trust model in [Park et al., 2011] takes advantage of daily commutes. In this article, the focus is on the early stages of VANETs, in which a very small percentage of vehicles are equipped with OBUs. To make trust viable in such a scenario, the authors rely on RSUs to store reputation information from passing vehicles. Each vehicle must have an "Agent RSU", which is be tasked with sharing that vehicle's trust data to other vehicles and RSUs as well as keeping the data updated when the vehicle is near it once again. To make this viable, the properties of daily commutes are used: it is assumed the vehicle would be near that RSU with reasonable frequency because it is located within the driver's home-to-work route. The main problem with this model is that it relies on the presence of frequent RSUs, which might not always be viable. It also does not make it clear what should happen when a vehicle stops using a route or does not have a daily

predictable path (it does, however, handle occasions in which a vehicle chooses an alternate route or is absent for some days such as weekends and holidays).

The authors of [Huang et al., 2014] take special note of two characteristics from social networks that can also be found in many VANET trust models: *information cascading* and *oversampling*. That is, information reported by a number of original nodes (i.e. the ones that witnessed an event) may be diluted as nodes that forward it append their own opinions on the matter. An algorithm is proposed to diminish that effect by assigning higher weights to the opinions of origin nodes and lower weights to others. However, the authors conclude that the optimal scenario is to assign no weight at all to forwarding nodes, therefore allowing each node to form an opinion based only on the original nodes' reports. Furthermore, the authors are quick to dismiss the validity of entity-based trust, instead opting for a pure data-oriented approach. Although it is true that data-oriented trust is efficient for events, which is what their model is based on, it is not ideal for sharing data quickly and frequently. When a collision or other major incident occurs, it is useful to judge each message on its own, since not all members of the network will have existing trust relationships with each other. However, when sharing location and velocity data several times per second, it is not reasonable to expect that each message will be analyzed so carefully; rather, it makes sense to form an opinion about the sender of the message and use the resulting trust value to choose which messages are relevant or not.

Chapter 4

Project

This work proposes that it is possible to adapt a trust management scheme originally designed for static networks to a dynamic environment, such as a VANET. In order to make this possible, it is necessary to identify features in VANETs that show that nodes can share a long-term relationship, as is the case for social networks. Through these long-term relationships, it then becomes feasible for nodes to store trust data and share it with other nodes. By combining a node's own opinions about familiar nodes and trust information received from its neighbors, it is possible to create a model of the surrounding network. This model includes a trust graph, showing the trust relationships between pair of nodes, which can then be used as input for a malicious node detection algorithm.

In this chapter, the details for the above process are presented. First, it is shown how vehicles can form long-term relationships and trust one another in a similar way to social networks. Then, the Working Day Movement Model is introduced, along with the reasons why it is best fitted for this work. Next, a previously existing trust model for complex networks is described, including its advantages and reasons why it could be well-fitted for a vehicular environment, followed by a list of changes that must be made to the model in order for it to handle dynamic networks. Finally, a schedule for the next steps of the project is shown.

4.1 Goals

4.2 Social Networks and VANETs

Some proposed trust models for vehicular networks, such as [Huang et al., 2014], state that the likelihood of two nodes meeting each other twice is too low to be relevant. However, it stands to reason that, throughout the course of several days, many drivers take similar routes at similar times of day (e.g. to commute to work) and, therefore, their vehicles are in similar locations each day. Additionally, many cities rely on main roads to serve as backbones to their traffic, meaning there is a high density of vehicles on those roads during rush hours. Since

that is true for a notable percentage of a city's fleet, it can also be assumed that those vehicles may frequently encounter each other during their commute. While two vehicles that share a commute route may not be direct neighbors every day, they are likely to be relatively close to each other most days, meaning few hops separate them in the ad-hoc network. Furthermore, certain pairs of vehicles are bound to be within communication range of each other nearly every day. Examples of these include vehicles whose owners are neighbors or coworkers. Such vehicles' trust relationship should become steady over time and, in the case of positive trust, they can use each other's information to learn more about other nodes in the network.

Most cities also have one or more types of mass transit systems (buses or trains). Those vehicles can also be part of a VANET and communicate with private cars. Buses share the same roads as cars, but instead of having specific destinations, they travel a predefined route during the whole day, usually tied to a tight schedule. Trains travel on rails, so their contact with cars is less frequent, but it can also happen on railroad crossings; they travel long distances in relatively short amount of time, which helps the dissemination of data in a VANET. In the same way that cars have a high probability of meeting more than once during their commutes, it is also very likely that they meet the same buses and/or trains frequently.

In [Cunha et al., 2013], [Cunha et al., 2014a], and [Cunha et al., 2014b], the authors attempt to find features usually attributed to social networks in vehicular networks. By using a data set from Zurich, they show that some metrics, such as clustering coefficient and number of encounters, have peaks during the rush hours. They note that, during rush hours, the diameter of the graph decreases to around 6 hops; additionally, the frequency of total encounters between pairs of nodes in the network increases during those hours. Although the authors do not quantify the encounters between specific pairs of nodes, these numbers support the idea that daily commutes do indeed cause vehicular networks to exhibit social network features.

4.3 Original trust model

The basis of the proposed trust model is [Vernize et al., 2015]. This article presents a malicious node identification scheme based on strongly connected components and graph coloring. The model is proposed for complex networks in general, but is not suited for VANETs because it is designed only for static networks. Furthermore, the algorithm is executed by a global observer which has information about the complete network.

The input graph $T = (V, O)$ is a static, connected, and directed graph containing all trust relationships in the network. Such relationships are binary, so there are no varying degrees of trust: either one node trusts another completely (edge value is 1), or it distrusts the other completely (edge value is 0). The relationships are also directed, meaning that if the value of $A \rightarrow B$ is 1, $B \rightarrow A$ is not necessarily 1.

The process for identifying malicious nodes within T is as follows:

First, T is separated into strongly components using Tarjan's algorithm [Tarjan, 1972]. In each of these components, all nodes are connected by edges of value 1. In other words, within a single component, all nodes trust one another completely; nodes connected by edges of value 0 are separated into different components. Each of these components becomes a node of a component graph $T' = (V', O')$.

The creation of the graph T' simplifies the remaining computation. Since each node of T' is a vertex $v' \in V'$ and each vertex v' is a component of T in which all nodes trust each other, for the purposes of identifying malicious nodes, all nodes within each of those components can be treated as one. They can either be benign nodes which legitimately trust one another, or malicious nodes colluding with each other. After the formation of T', one or more heuristics can be used to classify the nodes as benign or malicious.

The article describes the coloring heuristic, which uses a graph coloring algorithm, either DSATUR [Brélaz, 1979] or the algorithm proposed in [Mittal et al., 2011]. Graph coloring is a classic problem of graph theory, in which the objective is to assign each node in a graph a color so that no two neighboring nodes share the same color. After running either algorithms with graph T', the color with the most nodes in T' is classified as correct, and all others are classified as malicious. Once this information in T' is brought back to graph T, it is trivial to label the nodes in T as either benign or malicious based on their components' classifications.

In the experiments shown in the article, the coloring heuristic shows promising results, identifying a high ratio of the malicious nodes in the network. Other heuristics are experimented with, but were less effective in detecting malicious nodes, or provided too many false positives. Therefore, for the purposes of this paper, only the coloring heuristic is considered.

Two types of experiments were made in each network: first, all malicious nodes inverted the edge weights leading to their neighbors; second, malicious nodes randomly inverted or not the weights. In the first scenario, the results show excellent precision in most networks, detecting nearly every malicious node. Experimenting with the second scenario, the results are less precise, however still promising: with up to 20% of malicious nodes in the network, the error rate is under 7%, while with the worst case, 50% of the network being malicious, the error rate is approximately 15%.

The authors suggest running the algorithm repeatedly after removing the malicious nodes from the network. By doing this, nearly all malicious nodes are detected by it even when randomly changing edge weights.

4.4 Tarjan's strongly connected components algorithm

An important aspect of [Vernize et al., 2015] is the use of Tarjan's strongly connected components algorithm [Tarjan, 1972]. This allows a large graph to be abstracted into a smaller graph, which therefore reduces the input for further algorithms. Given a directed graph $T = (V, O)$, a strongly connected component is defined as a group of nodes in which, for any $u, v \in O$, there

exists a path from u to v and a path from v to u. Every node of the input graph T must belong to a component.

The algorithm works by performing a depth-first search, adding nodes to $stack$ as they are visited. If two nodes are present on $stack$, then there is a path from the first node to the second one (in the order they were added to the stack). Each node has two attributes assigned to it during the execution of the algorithm: $index$ is used to number the nodes in the order they are visited, while $lowlink$ is the lowest indexed node reachable from each node.

In the implementation used, $index$, $lowlink$, $count$ and $stack$ are global variables accessed from every call of the function. $index$ and $lowlink$ are arrays indexed by node IDs, $count$ is an integer and $stack$ is a last-in-first-out data structure.

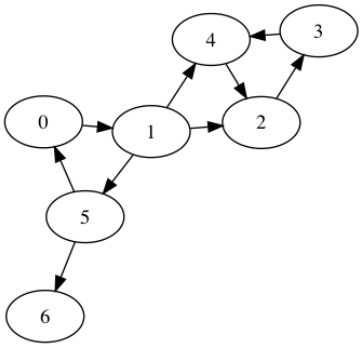
In the call that visits a node u, the algorithm must loop through each node v for which $u \rightarrow v$ exists. If v has not yet been visited, the algorithm is called for node v. The $lowlink$ of u is then calculated as the smallest value between $lowlink[u]$ and $lowlink[v]$, because any node reachable from v is also reachable from u. After the loop, if $lowlink[u]$ is equal to $index[u]$, it means that u is the lowest indexed node reachable from itself and that it is the root of a component. Therefore, nodes must be popped from the $stack$ until u is found. Each node popped, including u, is a member of a strongly connected component.

Algorithm 1 shows the general structure of Tarjan's algorithm [Tarjan, 1972] [Vernize, 2013]. The complexity of the algorithm is $O(|V| + |E|)$ for a graph $T = (V, E)$.

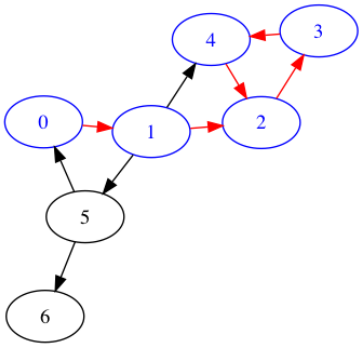
Algorithm 1 Tarjan's strongly connected components algorithm

```
1:  function TARJAN(vertex u)
2:      index[u] = count
3:      lowlink[u] = count
4:      count ← count + 1
5:      push u to stack
6:      for v in neighbors of u  do
7:          if index[v] = −1 then    // v has not been visited yet
8:              Tarjan(v)
9:          lowlink[u] ← min(lowlink[u], lowlink[v])
10:     if lowlink[u] = index[u] then
11:         repeat   // unstack nodes until u is found
12:             pop w from stack
13:             add w to component
14:         until w = u
```

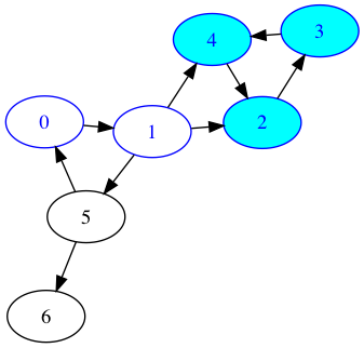
Figure 4.1 illustrates the execution of Tarjan's algorithm. The algorithm starts from node 0, with $index[0] = 0$ and $lowlink[0] = 0$. With a depth-first search, the algorithm traces the path $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2$. Since node 2 has already been visited and nodes 4 and 3 have no further outgoing edges, $lowlink[4]$ and $lowlink[3]$ receive the value 2 and the function calls return back to the first visit of node 2. At this point, nodes 2, 3 and 4 all have 2 as the smallest reachable index and, therefore, they form a strongly connected component.



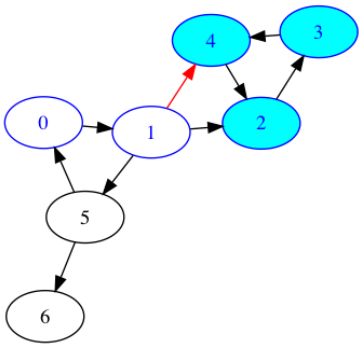
(a) Initial graph.



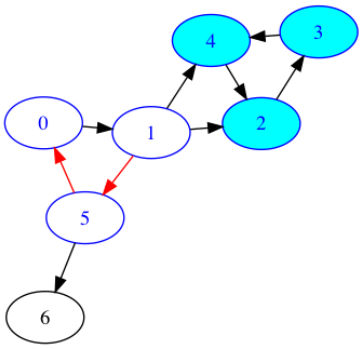
(b) The search follows the path $0 \to 1 \to 2 \to 3 \to 4 \to 2$.



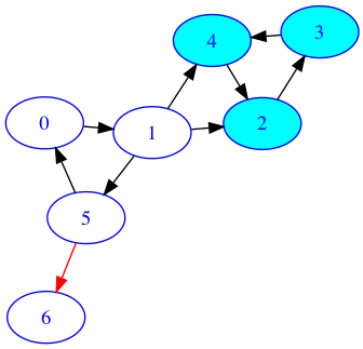
(c) Since $lowlink[2] = 2$, nodes 2, 3 and 4 are added into a component.



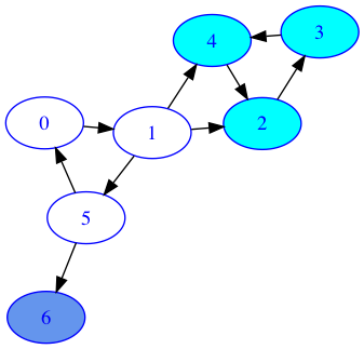
(d) The search follows the path $1 \to 4$, but node 4 has already been visited.



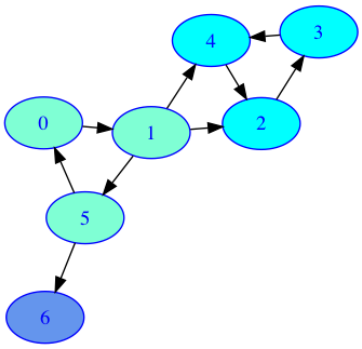
(e) The search follows the path $1 \to 5 \to 0$.



(f) The search follows the path $5 \to 6$.



(g) Node 6 does not have outgoing edges and is added to a component.



(h) The algorithm returns to 0 and, since $lowlink[0] = 0$, adds 0, 1 and 5 to a component.

Figure 4.1: Example of an execution of Tarjan's strongly connected components algorithm.

Continuing from node 1, the algorithm traces $1 \rightarrow 5 \rightarrow 0$, but stops there since node 0 has already been visited. Continuing from node 5, the algorithm traces $5 \rightarrow 7$. Node 7 has no outgoing edges, so it forms a strongly connected component by itself. Once the function calls return to node 0, a strongly connected component is formed with nodes 0, 1, and 5, since they all have 0 as their $lowlink$ value.

4.5 Graph coloring with minimum colors

The algorithm proposed in [Mittal et al., 2011] is an efficient approach to graph coloring, a classic graph theory problem. Graph coloring is one of the heuristics used in [Vernize, 2013] to detect malicious nodes after the generation of the component graph using Tarjan's algorithm. Since it was the heuristic that presented the best results, it has been chosen as the heuristic for this current study as well.

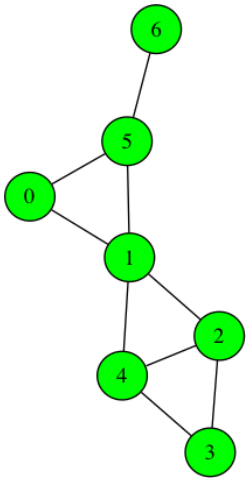
The process of graph coloring consists of giving each node a label (represented by a color) so that no two neighboring nodes share the same color. This problem has been studied in Computer Science since, at least, 1972 [Karp, 1972] and has been studied as a classic mathematics problem for even longer [Kempe, 1879]. It has been proven mathematically that any planar graph can be colored with at most four colors [Appel et al., 1976], but discovering the smallest number of colors necessary to color an arbitrary graph (called the graph's chromatic number) is an NP-hard problem [Sánchez-Arroyo, 1989].

In [Mittal et al., 2011], the authors propose to color a graph using the minimum possible amount of colors. Although they do not prove that their algorithm always uses the smallest possible amount of colors, the output is always a correct coloration and the algorithm is nevertheless efficient. The complexity of the algorithm is $O(|E|)$ for a graph $G = (V, E)$. As a comparison, the DSATUR algorithm for graph coloring [Brélaz, 1979] has complexity $O(|V|^2)$. For the purposes of this study, it is not necessary to prove that the coloring algorithm's output uses the minimum possible number of colors.

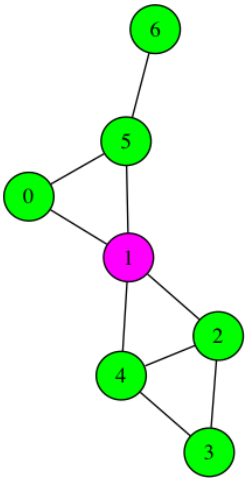
Algorithm 2 shows the general structure of the graph coloring algorithm [Mittal et al., 2011] [Vernize, 2013].

Algorithm 2 Graph coloring with minimum colors

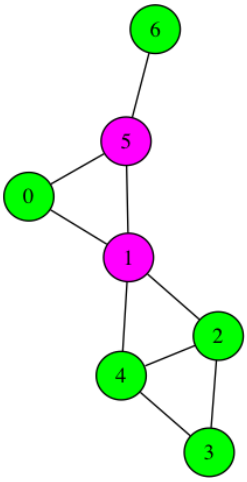
```
1: function COLORING(graph G)
2:     color all nodes of G with 0
3:     d → 0
4:     for e = (u, v) in edges of G do
5:         if u and v have the same color then
6:             if color[v] = d then
7:                 d → d + 1
8:             color[v] → d
```



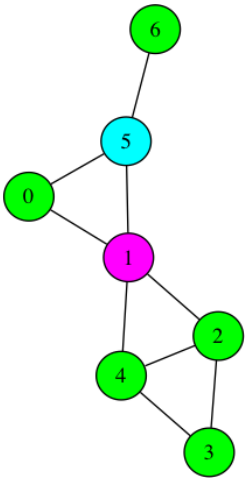
(a) Initial graph with all nodes labeled 1. $d = 1$ (green).



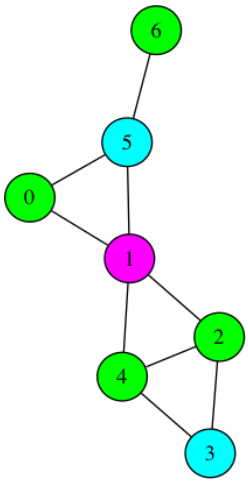
(b) Edge $(0, 1)$ is checked and node 1 gets a new color. $d = 2$ (magenta).



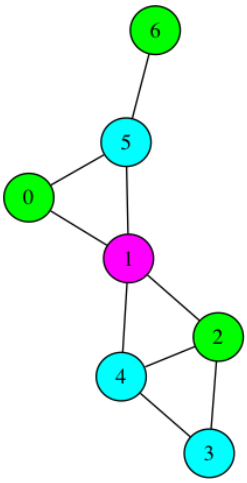
(c) Edge $(0, 5)$ is checked and node 5 gets the current value of d.



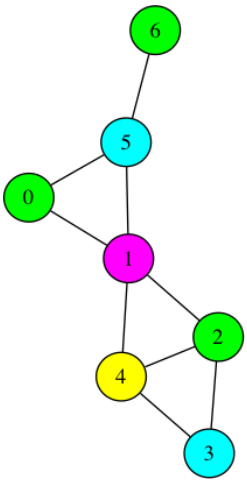
(d) Edge $(1, 5)$ is checked and node 5 gets a new color. $d = 3$ (cyan).



(e) Edge $(2, 3)$ is checked and node 3 gets the current value of d.



(f) Edge $(2, 4)$ is checked and node 4 gets the current value of d.



(g) Edge $(3, 4)$ is checked and node 4 gets a new color. $d = 4$ (yellow).

Figure 4.2: Example of an execution of the graph coloring with minimum colors algorithm.

A limitation of this algorithm is that the edges must be sorted according to node indexes. It doesn't matter which nodes get assigned which indexes, but once they are assigned those numbers, the algorithm must follow the edges in numerical order. This is demonstrated in [Vernize, 2013].

Figure Figure 4.2 illustrates the execution of the graph coloring algorithm. It is notable how few iterations the algorithm takes to fully color the graph. However, the fact that the result does not use the minimum amount of colors. By coloring node 4 as cyan and node 3 as magenta, the sample graph could have been colored with only three colors. However, as described above, this is not a substantial problem for the purposes of this study.

4.6 SNAP library

4.7 Trust management algorithm

Chapter 5

Simulations

5.1 The ONE Simulator

5.2 Working Day Movement Model

Most VANET trust models consider the Random Waypoint mobility model, i.e. each node has an origin point, chooses a random location, gets to that location, then chooses another random location and goes there, and so forth. While this model is efficient for testing trust protocols, it doesn't truly represent vehicle mobility in the real world.

To make use of the properties described in section 4.2, it is important to choose a mobility pattern that properly represents the way vehicles move on a daily basis in the real world. Therefore, the Working Day Movement Model [Ekman et al., 2008] (WDM) is useful. The model, developed for use in Delay-Tolerant Network (DTN) simulations, includes many of the features that are necessary to simulate the daily movement of a vehicular network.

As the name implies, the Working Day Model abstracts people's movement from their homes to their offices and back. Each node has a home and a workplace and they need to travel back and forth between those locations on a daily basis. Occasionally, nodes can also go to other locations for leisure. As mentioned above, many drivers have routes they travel on daily, so the Working Day model is a more accurate representation, although it represents the mobility of humans instead of vehicle and requires some adjustments, which are listed later in this section.

The model proposed by the authors makes use of several other models for specific tasks. The main mobility model defines nodes and gives them their destinations. Within it, five other models are used:

1. The **home activity submodel** describes what nodes do at night, within their homes. No movement is modeled. Nodes can be relatives or neighbors, and therefore share the same home.

2. The **office activity submodel** describes the nodes' routines within their offices. Nodes can go to other locations within the office (such as meeting rooms) and such movement is modeled. Nodes with share the same office are coworkers.

3. The **evening activity submodel** is responsible for mobility outside the nodes' standard routine. They can meet at certain locations (such as restaurants) and spend a few hours gathered with friends.

4. The **transport submodel** shows how nodes move around the city. It includes another tier of submodels, responsible for modeling three different types of transportation: walking, driving, and riding a bus. Nodes which own a car always use it, while the others can decide to walk or ride a bus depending on the distance between the origin and destination and the available bus stops. The walking and driving submodels represent similar types of movement, although at different speeds, while the bus submodel follows cyclical routes and can take or deliver passengers at bus stops.

5. The **map** represents the city in which the simulation runs. Its streets constrain the movement of nodes and all homes and offices must be within the map boundaries. The map can be divided into districts, which increases what the authors define as *locality*. In the simulation parameters, the number of nodes which reside and work within the same district can be chosen, which means those nodes rarely leave the district. Nodes which reside and work in different districts serve to connect the network with their commutes.

By thinking of these submodels for vehicles instead of people, it can become apparent how the frequency and length of encounters between nodes are similar in both instances. If two vehicles belong to family members or neighbors, they likely spend most of the night within communication distance, while coworkers' cars spend the office hours close by. Cars can also meet each other frequently if their drivers go out with friends. In the vehicular case, there is the added layer of encounters: cars can communicate frequently with buses and other cars that take the same route daily, although the drivers are likely complete strangers.

In the original article, nodes are devices (such as smartphones) being carried by humans. Therefore, the Working Day model represents not only people's movements inside their cars, but also within their offices, walking on foot, or riding a bus. To adapt it to a VANET environment, changes need to be made to their definition of node, since they now represent vehicles instead of people. Some of those changes are as follows:

1. The office activity submodel no longer needs to model movement within the office and can be identical to the home submodel. In both, a node can move a small amount once after reaching the office or home, to simulate parking. This can be done using the Working Day model's parameters.

2. The walking submodel needs to be disabled, since all nodes are either cars or buses.

3. The bus submodel needs to be changed so that each bus is one node in the network, which follows a predefined route with bus stops. In the original model, each bus could carry several nodes, but this is no longer necessary.

Other necessary changes might become apparent during the development of the study.

One important topic raised in the Working Day movement model article is the use of two metrics: *inter-contact times* and *contact duration*. The choice of this movement model for tests is more strongly related to inter-contact times, i.e. how much time it takes for two nodes to meet again. On the other hand, the contact duration is how long each meeting lasts. For the reasons explained earlier in this section, relatively short inter-contact times is important for the proposed trust model. Contact duration time is an important metric to measure how much data can be exchanged during each encounter, although taking it into consideration might add excessive complexity to the model.

5.3 Simulation parameters and methodology

5.3.1 Validation

5.4 Restrictions

5.5 Results

Chapter 6

Conclusion

Bibliography

[Amoozadeh et al., 2015] Amoozadeh, M., Deng, H., Chuah, C.-N., Zhang, H. M., and Ghosal, D. (2015). Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular communications*, 2(2):110–123.

[Appel et al., 1976] Appel, K., Haken, W., and Koch, J. (1976). Every planar map is four colorable. *Bull. Amer. Math. Soc*, 82(5):711–712.

[Baras and Jiang, 2005] Baras, J. S. and Jiang, T. (2005). Cooperation, trust and games in wireless networks. In *Advances in Control, Communication Networks, and Transportation Systems*, pages 183–202. Springer.

[Barba et al., 2012] Barba, C. T., Mateos, M. A., Soto, P. R., Mezher, A. M., and Igartua, M. A. (2012). Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 902–907. IEEE.

[Boissevain, 1974] Boissevain, J. (1974). *Friends of friends: Networks, manipulators and coalitions*. Blackwell Oxford.

[Brélaz, 1979] Brélaz, D. (1979). New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256.

[CAMP Vehicle Safety Communications Consortium, 2005] CAMP Vehicle Safety Communications Consortium (2005). Vehicle safety communications project: Task 3 final report: identify intelligent vehicle safety applications enabled by dsrc. *National Highway Traffic Safety Administration, US Department of Transportation, Washington DC*.

[Chen et al., 2010] Chen, C., Zhang, J., Cohen, R., and Ho, P.-H. (2010). A trust modeling framework for message propagation and evaluation in vanets. In *Information Technology Convergence and Services (ITCS), 2010 2nd International Conference on*, pages 1–8. IEEE.

[Chou et al., 2009] Chou, C.-M., Li, C.-Y., Chien, W.-M., and Lan, K.-c. (2009). A feasibility study on vehicle-to-infrastructure communication: Wifi vs. wimax. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 397–398. IEEE.

[Cunha et al., 2013] Cunha, F. D., Vianna, A. C., Mini, R. A., and Loureiro, A. A. (2013). How effective is to look at a vehicular network under a social perception? In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, pages 154–159.

[Cunha et al., 2014a] Cunha, F. D., Vianna, A. C., Mini, R. A., and Loureiro, A. A. (2014a). Are vehicular networks small world? In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 195–196. IEEE.

[Cunha et al., 2014b] Cunha, F. D., Vianna, A. C., Mini, R. A., and Loureiro, A. A. (2014b). Is it possible to find social properties in vehicular networks? In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.

[Ding et al., 2013] Ding, Q., Li, X., Jiang, M., and Zhou, X. (2013). A novel reputation management framework for vehicular ad hoc networks. *International Journal of Multimedia Technology*, 3(2):62–66.

[Dotzer et al., 2005] Dotzer, F., Fischer, L., and Magiera, P. (2005). Vars: A vehicle ad-hoc network reputation system. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 454–456. IEEE.

[Dwivedi and Dubey, 2016] Dwivedi, S. and Dubey, R. (2016). Review in trust and vehicle scenario in vanet. *International Journal of Future Generation Communication and Networking*, 9(5):305–314.

[Ekman et al., 2008] Ekman, F., Keränen, A., Karvo, J., and Ott, J. (2008). Working day movement model. In *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40. ACM.

[Garip et al., 2015] Garip, M. T., Gursoy, M. E., Reiher, P., and Gerla, M. (2015). Congestion attacks to autonomous cars using vehicular botnets. In *NDSS Workshop on Security of Emerging Networking Technologies (SENT), San Diego, CA*.

[Gerlach, 2007] Gerlach, M. (2007). Trust for vehicular applications. In *Autonomous Decentralized Systems, 2007. ISADS'07. Eighth International Symposium on*, pages 295–304. IEEE.

[Golbeck and Hendler, 2006] Golbeck, J. and Hendler, J. (2006). Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology (TOIT)*, 6(4):497–529.

[Golle et al., 2004] Golle, P., Greene, D., and Staddon, J. (2004). Detecting and correcting malicious data in vanets. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37. ACM.

[Haddadou et al., 2013] Haddadou, N., Rachedi, A., and Ghamri-Doudane, Y. (2013). Trust and exclusion in vehicular ad hoc networks: an economic incentive model based approach. In *Computing, Communications and IT Applications Conference (ComComAp), 2013*, pages 13–18. IEEE.

[Hafner et al., 2011] Hafner, M., Cunningham, D., Caminiti, L., and Del Vecchio, D. (2011). Automated vehicle-to-vehicle collision avoidance at intersections. In *Proceedings of world congress on intelligent transport systems*.

[Huang et al., 2010] Huang, D., Hong, X., and Gerla, M. (2010). Situation-aware trust architecture for vehicular networks. *IEEE Communications Magazine*, 48(11).

[Huang et al., 2014] Huang, Z., Ruj, S., Cavenaghi, M. A., Stojmenovic, M., and Nayak, A. (2014). A social network approach to trust management in vanets. *Peer-to-Peer Networking and Applications*, 7(3):229–242.

[INRIX, 2017] INRIX (2017). Los Angeles Tops INRIX Global Congestion Ranking. `http://inrix.com/press-releases/los-angeles-tops-inrix-global-congestion-ranking/`. [Online; accessed March 27, 2017].

[Isaac et al., 2010] Isaac, J. T., Zeadally, S., and Camara, J. S. (2010). Security attacks and solutions for vehicular ad hoc networks. *IET communications*, 4(7):894–903.

[Jiang and Delgrossi, 2008] Jiang, D. and Delgrossi, L. (2008). Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040. IEEE.

[Johnson, 2010] Johnson, T. D. (2010). U.S. traffic deaths drop to lowest level since 1949. `http://thenationshealth.aphapublications.org/content/41/4/E17.full`. [Online; accessed March 27, 2017].

[Karp, 1972] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

[Kempe, 1879] Kempe, A. B. (1879). On the geographical problem of the four colours. *American journal of mathematics*, 2(3):193–200.

[Kerrache et al., 2016] Kerrache, C. A., Lakas, A., and Lagraa, N. (2016). Detection of intelligent malicious and selfish nodes in vanet using threshold adaptive control. In *Electronic Devices, Systems and Applications (ICEDSA), 2016 5th International Conference on*, pages 1–4. IEEE.

[Knorr et al., 2012] Knorr, F., Baselt, D., Schreckenberg, M., and Mauve, M. (2012). Reducing traffic jams via vanets. *IEEE Transactions on Vehicular Technology*, 61(8):3490–3498.

[Krochmal et al., 2014] Krochmal, M., Edmonds, C., Jensen, C., and Prats, A. (2014). Discovery of nearby devices for file transfer and other communications. US Patent App. 14/037,272.

[Lee et al., 2004] Lee, J. D., Hoffman, J. D., and Hayes, E. (2004). Collision warning design to mitigate driver distraction. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 65–72. ACM.

[Leinmüller et al., 2005] Leinmüller, T., Schoch, E., Kargl, F., and Maihöfer, C. (2005). Influence of falsified position data on geographic ad-hoc routing. In *European Workshop on Security in Ad-hoc and Sensor Networks*, pages 102–112. Springer.

[Li and Wang, 2007] Li, F. and Wang, Y. (2007). Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular technology magazine*, 2(2).

[Liu et al., 2016] Liu, Z., Ma, J., Jiang, Z., Zhu, H., and Miao, Y. (2016). Lsot: A lightweight self-organized trust model in vanets. *Mobile Information Systems*, 2016.

[Ma et al., 2011] Ma, S., Wolfson, O., and Lin, J. (2011). A survey on trust management for intelligent transportation system. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 18–23. ACM.

[Mangel et al., 2010] Mangel, T., Kosch, T., and Hartenstein, H. (2010). A comparison of umts and lte for vehicular safety communication at intersections. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 293–300. IEEE.

[McCole, 2016] McCole, M. (2016). How to Make the Amazon Echo the Center of Your Smart Home. `https://www.wired.com/2016/01/iot-cookbook-amazon-echo/`. [Online; accessed April 20, 2017].

[Mejri et al., 2014] Mejri, M. N., Ben-Othman, J., and Hamdi, M. (2014). Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, 1(2):53–66.

[Minhas et al., 2010] Minhas, U. F., Zhang, J., Tran, T., and Cohen, R. (2010). Towards expanded trust management for agents in vehicular ad-hoc networks. *International Journal of Computational Intelligence: Theory and Practice (IJCITP)*, 5(1):03–15.

[Mittal et al., 2011] Mittal, A., Jain, P., Mathur, S., and Bhatt, P. (2011). Graph coloring with minimum colors: An easy approach. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 638–641. IEEE.

[Morgan, 2014] Morgan, J. (2014). A Simple Explanation Of 'The Internet Of Things'. `https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/`. [Online; accessed April 20, 2017].

[Newman, 2010] Newman, M. (2010). *Networks: an introduction*. Oxford University Press.

[Park et al., 2011] Park, S., Aslam, B., and Zou, C. C. (2011). Long-term reputation system for vehicular networking based on vehicle's daily commute routine. In *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pages 436–441. IEEE.

[Patwardhan et al., 2006] Patwardhan, A., Joshi, A., Finin, T., and Yesha, Y. (2006). A data intensive reputation management scheme for vehicular ad hoc networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–8. IEEE.

[Raya et al., 2008] Raya, M., Papadimitratos, P., Gligor, V. D., and Hubaux, J.-P. (2008). On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1238–1246. IEEE.

[Real-Time Innovations, 2014] Real-Time Innovations (2014). How the internet of things can save 50,000 lives a year.

[Saini et al., 2015] Saini, M., Alelaiwi, A., and Saddik, A. E. (2015). How close are we to realizing a pragmatic vanet solution? a meta-survey. *ACM Computing Surveys (CSUR)*, 48(2):29.

[Sánchez-Arroyo, 1989] Sánchez-Arroyo, A. (1989). Determining the total colouring number is np-hard. *Discrete Mathematics*, 78(3):315–319.

[Sasaki and Kuwahara, 2011] Sasaki, T. and Kuwahara, M. (2011). Wireless network system and wireless communication program. US Patent 8,073,923.

[Sengar, 2016] Sengar, J. S. (2016). Survey: Reputation and trust management in vanets. *International Journal of Grid and Distributed Computing*, 9(1):201–206.

[Sherchan et al., 2013] Sherchan, W., Nepal, S., and Paris, C. (2013). A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47.

[Soleymani et al., 2015] Soleymani, S. A., Abdullah, A. H., Hassan, W. H., Anisi, M. H., Goudarzi, S., Baee, M. A. R., and Mandala, S. (2015). Trust management in vehicular ad hoc network: a systematic review. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):146.

[Stewart, 2016] Stewart, J. (2016). Tesla's Self-Driving Car Plan Seems Insane, But It Just Might Work. `https://www.wired.com/2016/10/teslas-self-driving-car-plan-seems-insane-just-might-work/`. [Online; accessed April 11, 2017].

[Tarjan, 1972] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

[Vernize, 2013] Vernize, G. (2013). *Identificação de nós maliciosos em redes complexas baseada em visões locais*. MSc dissertation, Universidade Federal do Paraná.

[Vernize et al., 2015] Vernize, G., Guedes, A. L. P., and Albini, L. C. P. (2015). Malicious nodes identification for complex network based on local views. *The Computer Journal*, 58(10):2476–2491.

[Viereckl et al., 2016] Viereckl, R., Ahlemann, D., Koster, A., Hirsh, E., Kuhnert, F., Mohs, J., Fischer, M., Gerling, W., Gnanasekaran, K., and Kusb, J. (2016). Connected car report 2016: Opportunities, risk, and turmoil on the road to autonomous vehicles. `http://www.strategyand.pwc.com/reports/connected-car-2016-study`. [Online; accessed April 11, 2017].

[Wang et al., 2009] Wang, J., Liu, Y., Liu, X., and Zhang, J. (2009). A trust propagation scheme in vanets. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 1067–1071. IEEE.

[Wasef et al., 2010] Wasef, A., Lu, R., Lin, X., and Shen, X. (2010). Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]. *IEEE Wireless Communications*, 17(5).

[World Health Organization, 2013] World Health Organization (2013). Number of road traffic deaths. `http://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en/`. [Online; accessed March 27, 2017].

[World Health Organization, 2015] World Health Organization (2015). Road traffic injuries fact sheet. `http://www.who.int/mediacentre/factsheets/fs358/en/`. [Online; accessed March 27, 2017].

[Wu and Stojmenovic, 2004] Wu, J. and Stojmenovic, I. (2004). Ad hoc networks. *COMPUTER-IEEE COMPUTER SOCIETY-*, 37(2):29–31.

[Yang et al., 2004] Yang, X., Liu, L., Vaidya, N. H., and Zhao, F. (2004). A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 114–123. IEEE.

[Yousefi et al., 2006] Yousefi, S., Mousavi, M. S., and Fathy, M. (2006). Vehicular ad hoc networks (vanets): challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766. IEEE.

[Zhang, 2011] Zhang, J. (2011). A survey on trust management for vanets. In *2011 IEEE International Conference on Advanced Information Networking and Applications*, pages 105–112. IEEE.

[Zhang, 2012] Zhang, J. (2012). Trust management for vanets: challenges, desired properties and future directions. *International Journal of Distributed Systems and Technologies (IJDST)*, 3(1):48–62.