REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Sceitifique

Université Ferhat Abbas Sétif

Faculté des Sciences

Départment d'Informatique

Université Ferhat Abbas Sétif 1

Réseaux de Capteurs Sans Fils Véhiculaire Sûrs de Fonctionnement

---

Une thèse présenté par:

Khedidja MEDANI

En vue de l'obtention du diplôme de

Doctorat $3^{ième}$ cycle LMD en Informatique

Soutenue devant le jury composé de:

Nadjet	KAMEL	Prof. Université Ferhat Abbas Sétif 1	Président
Makhlouf	ALIOUAT	Prof. Université Ferhat Abbas Sétif 1	Papporteur
Mohammed	BENMOHAMED	Prof. Université Constantine 3	Examinateur
Omar	MAWLOUD	MCA Université A. Mira Béjaia	Examinateur
Malek	BOUDRIES	MCA Université A. Mira Béjaia	Invité
Mourad	AMAD	MCA Université A. M. Oulhadj Bouira	Invité

2017 / 2018

PEOPLEs' DEMOCRATIC REPUBLIC of ALGERIA
Ministry of Higher Education and Scientific Research
Ferhat Abbas University of Sétif
Faculty of Sciences
Department of Computer Science

Université Ferhat Abbas Sétif 1

Dependable Vehicular Sensor Networks

A thesis presented by:

Khedidja MEDANI

As a requirement to aim for the degree of

3^{rd} cycle Doctorate in computer science

Approved by supervisory committee:

Nadjet	KAMEL	Prof. Université Ferhat Abbas Sétif 1	President
Makhlouf	ALIOUAT	Prof. Université Ferhat Abbas Sétif 1	Advisor
Mohammed	BENMOHAMED	Prof. Université Constantine 3	Examinator
Omar	MAWLOUD	MCA Université A. Mira Béjaia	Examinator
Malek	BOUDRIES	MCA Université A. Mira Béjaia	Invited
Mourad	AMAD	MCA Université A. M. Oulhadj Bouira	Invited

2017 / 2018

Until we can manage time
We can manage nothing else.

“ Peter F. Drucker”

Declaration

I, Khedidja MEDANI, declare that this thesis, “ Dependable Vehicular Sensor Networks (Clock Synchronization) ”, and the work presented in it are my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

Signed: ....
Date: ....

Abstract

Aiming to the main purpose of improving road safety and entertainment, Vehicular Ad Hoc Networks (VANETs) have emerged as an open and interesting research topic in the last decade. The deployment of VANET systems coupled with sensor technologies has increased the gained benefits from the development of these latter. Thus, they enable real-time data gathering and sharing, so that new applications, such as traffic reporting, relief to environmental monitoring and distributed surveillance are promoted. The design of reliable, fault-tolerant, maintainable, safe and secure applications and standard for realistic and large-scale deployment environment, like in VANETs presents extraordinary challenge, especially in the lack of global memory enabling global system state recognition. In this context, the requirement of clock synchronization remains one of the most significant issues that should be addressed to the extent of that dependable systems evolve. The focal point of this Doctoral dissertation is to give an analytical study of clock synchronization issue in vehicular communication systems. The intrinsic characteristics of the unstable vehicular environments, such as the high speed of nodes and the lack of permanent network connectivity have created new challenges and requirements, so that the solutions that have already proposed to synchronize nodes in classical networks are no longer appropriate. Consequently, new and adaptive clock synchronization mechanisms should be devised and implemented. Here, we propose a new mechanism for synchronizing clocks in vehicular environments, dealing so with communication and scalability issues. The proposal, named as "Offsets Table Robust Broadcasting" (OTRB), exploits the broadcasting channel to spread the time information over the entire network. This protocol is well-adapted to random network topology changes, high node velocity while offering good precision and robustness against nodes failure and packet loss. The analytical study and protocol simulation for evaluating the system performance, carried out by a combination of VanetMobiSim and NS2 simulators, have yielded convincing results, outperforming those exhibited by the basic referred protocols.

keywords Intelligent transportation system, ITS, VANET, WSN, dependability, clock synchronization, fault-tolerant clock synchronization.

Acknowledgments

In the name of ALLAH, the most gracious and merciful. I thank ALLAH for the strength and blessing he gives me. I have never expected that I would have arrived to complete this work without the aid of ALLAH.

My most sincere gratitude goes to my enthusiastic supervisor, **Prof Makhlouf ALIOUAT**, for his tremendous academic support.

Special mention goes to **Dr. Zibouda ALIOUAT**, not only for her support and the advice she provided, but also for giving me so many wonderful opportunities on different aspects of my research.

I also appreciate the availability of **Dr. Hamida SEBA**, especially during my presence at LIRIS laboratory, University Lyon1 and for the guidance she gave me.

I wish to thank the members of my dissertation committee for generously offering their time, support, guidance and good will throughout the review of this document.

Last and not least, I would like to thank all my friends and colleagues for their encouragement.

Dedication

In memory of my father

This dissertation is dedicated

To my mother, at first and foremost

To my brothers and sisters

To all my family and friends

For all their love, patience, kindness and support.

Table of contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

MANET	Mobile Ad hoc NETwork
VANET	Vehicular Ad hoc NETwork
WSN	Wireless Sensor Network
ITS	Intelligent Transportation System
IVC	Inter Vehicular Communication
V2VC	Vehicle to Vehicle Communication
V2IC	Vehicle to Infrastructure Communication
IoT	Internet of Things
IoV	Internet of Vehicles
RSU	Road Side Unit
GPS	Global Positioning System
C2C-CC	Car to Car Communication Consortium
VASNET	Vehicular An hoc Sensor NETwork
VSN	Vehicular Social Network
VCC	Vehicular Cloud Computing
OTRB	Offsets Table Robust Broadcasting
TTD	Time Table Diffusion
ADV-T	Advertisement of Transporter node
JOIN-RESPONSE	Join Response message
TIME-TABLE	Time Table message
ACK	Acknowledgment message

Introduction

Nowadays, the use of wireless communication technologies in the transportation domain has increased with the active growth of smart cars' development. It was announced that the number of vehicles in use grew up from about tens' million in 2006 to thousands' million in 2015 and this number is still increasing for the time being. Thus, the demand for new ways to control the transportation systems is highly required. That is why Vehicular Ad hoc NETworks (VANETs) have been created.

VANETs apply data communication and processing to transport means, infrastructure and users in order to increase the effectiveness, environmental performance, safety, resilience and efficiency of the transportation eco-systems. VANETs can be treated as the application of Mobile Ad hoc NETworks (MANETs) for road traffic information sharing. They provide vehicular communications through wireless and mobile networks with a little or no infrastructures. In the case of pure ad hoc network architecture, the communications are established between the vehicles themselves (Vahicle to Vehicle communications (V2VC)). Otherwise, the vehicles can communicate with the infrastructure installed on roads (Vahicle to Infrastructure communications (V2IC)).

VANETs can provide a large amount of relevant applications and services, such as accident avoidance, collision warning, cooperative driving, wireless remote diagnosis, traffic flow regulation, route guidance, Internet access and infotainment applications like playing games and listening to music. Owed to this versatility of applications and services they promise to provide, VANETs are at the core of the intelligent transportation systems (ITSs). Even more, the trend of the Internet of Things (IoTs) enables the interaction between vehicles and IoT components which is referred to as the Internet of Vehicles (IoVs). Consequently, VANETs afford intelligent control and services to vehicular environments. In addition, the application of vehicular communications coupled with sensor technologies has been dragging the attention of both industrial and research communities. In this context, a number of VANETs' projects have been initialized by different countries and famous industrial firms, such as BMW and Toyota in order to make the ITSs a reality. Indeed, the unique properties of VANETs create new requirements and challenges. For instance, road pattern restrictions, large network size, dynamic topology and mobility models make previous MANETs solutions not applicable for VANET environments. Even so, the environments of vehicular and sensor networks are hostile, so that, vehicles and sensor inroads are subject to different failures making their real deployment a challenging task. This is due mainly to software and hardware failures, human error and unreliable networks. As a way of illustration, human errors affected by the drivers and pedestrians can damage the system in both hardware and software levels. In addition, hardware failures generally affects software failures, which in turn cause errors in the communication links leading to unreliable networks. In the context of network unreliability, the frequent network topology changes cause the communication links to be unpredictably created/broken-up. That is a node can connect or disconnect to a given set of other nodes in the laps of time. Furthermore, unexpected defects, such as malicious attacks and hardware failures, can damage system's safety or cause that the system cannot provide its specifications to the end-user. In order to guarantee dependable services, the design of VANETs' standards and protocols must be adapted to work robustly in such mobile and dense environments. In addition, the deployment must be designed to be energy-efficient, so the lifetime of the sensor deployed inroads can be maximized.

The dependability is to ensure reliability, availability, maintainability, safety and security of a system [1]. This can be realized by the means of fault-avoidance, fault-tolerance, error-removal and error forecasting, all in transparent manner [2]. Nonetheless, in the lack of global system' state recognition, to provide a global timescale for a set of connected nodes is of vital significance to support system dependability measurements. It is referred to as the clock synchronization. In other word to synchronize nodes' clocks is a cardinal requirement for distributed systems to meet their specifications to the end-user. It is a key parameter for system designing to ensure:

- Resource sharing: Several basic techniques to distribute the access to a shared resource are time-based access; where a node is allowed to access the resource in an allocated time interval. As a way of illustration, for communication medium access sharing time-based, the scheduling protocols are put in place in order to avoid communication collisions. These protocols do not make any sense if node's clocks are not synchronized with one another.

- Concurrency: Multiple nodes working jointly on a specific application need an agreement about their common progress at real-time. The difficulty of designing robust distributed systems arises in a limited knowledge of current global system's state. That is the timing sequence nature of the applications that evolve relies on an accurate clock synchronization mechanism to correctly maintain the event causality property.

- Fault tolerance: To ensure systems' reliability and maintainability, fault-tolerance is crucial. Obviously, nodes and communication links may fail independently at any time. Therefore, mechanisms to detect and recover faults must be placed in. For instance, check-pointing and redundancy mechanisms are configured to restore the system to a valid state. To this end, communications between the different nodes are inquired to be reliable and predictable in their timing behavior. Clock synchronization thus is required.

Additionally, several real-time applications and services depend strongly on the existence of synchronized clocks. Among these applications and services in VANET systems we cite: traffic information, smart parking, movement detection, localization, etc. Further, the absence of synchronized clocks may be the main cause of failure, such as time-based access to common services. For those reasons, our research tackles clock synchronization issue to ensure dependability of VANET systems.

Goals and objectives

The general goal of this PhD thesis is to deal with clock synchronization issue as a main pillar to support VANET systems dependability. However, clock synchronization issue is well studied in the classical MANET and WSN networks. Due to the remarkable lack of VANETs oriented solutions, our first objective is to survey the main research works in the specialized literature. Therefore, we give an overview of clock synchronization taxonomy in order to deal with its practicability in the context of VANET systems. However, most of the protocols in state of the art are designed to optimize the global skew of nodes using hierarchical architectures (e.g., clustering, like in [3, 4], or spanning tree, like in [5]). For instance, protocols, such as in [5] implement spanning tree architecture in order to maintain clock synchronization. Indeed, the error between neighboring nodes may increase, that is error propagates down with different rates on different paths/hops. In addition, it is very difficult to maintain a spanning tree in highly dynamic vehicular environments. On the other hand, achieving direct neighbor synchronization, such as in [6] [7] with the aid of timing messages exchange generate more communication overhead and cause transmission channel to interfere, especially in large and mobile environments. All in all, the synchronization performance is closely related to the characteristics of the network, such as type, size, node's distance, average speed and nodes' density. The analytical study reveals that the existing solutions cannot be directly implemented in the context of VANETs. That is, their intrinsic characteristics create extra challenges that should be considered. the high velocity of vehicles causes frequent changes in the network topology. Even more, the number of connected entities has incredibly increased and will increase more and more over the time.

Next, in order to bring new solution in satisfactory solving such problems, we attempt to develop a new proposal for synchronizing nodes' clock adapted to work under the conditions of unstable vehicular environments. In any case, a suitable clock synchronization protocol improves VANET requirements. Among these requirements are: low-cost in communication overhead, short time convergence (time convergence is the time taken to synchronize the network), scalability, precision and long synchronization lifetime. Robustness or fault tolerance is also an essential criterion that must be taken into consideration when designing the protocol. Fault tolerance defines the ability of the network to maintain its functionality in case of failure. The proposed protocol, called Offsets Table Robust Broadcasting (OTRB), provides accurate and fault-tolerant time synchronization, focusing on direct neighbor synchronization in which, every node runs with the value of its local clock, but maintains the time information needed to synchronize other nodes for different purposes (e.g., point to point communications, medium access control and security). The solution exploits the broadcasting channel to synchronize neighbor clocks without exchanging any time information between them, which minimizes the communication overhead and the number of delivered messages at each node.

Thesis organization

The thesis contains five chapters divided into three main parts, background, state of the art and contributions parts, and it is organized as follows:

In the overview part:

- Chapter 1 gives a background of knowledge about vehicular communication systems; their architecture, characteristics, challenges and requirements and some of the new open research topics related to the former.

- Chapter 2 discusses dependability of distributed systems, its main threats, means and measures and shed the light on the vital role that clock synchronization plays to support dependability in VANET systems. In this chapter, we give basic definitions and concepts related to clock synchronization and introduce the practical use of synchronized clocks, their requirements and challenges in vehicular environment deployments.

- The state of the art part weights the taxonomy of clock synchronization protocols with a detailed description of the existing protocols. The analysis provided for each protocol checks its practicability for VANET environments. Besides, a detailed comparison is provided considering relevant key parameters such as the density of nodes, the relative speed, the average time convergence, the communication overhead, the synchronization rate and the fault tolerance capability of the targeted protocol.

  In the second part of contributions:

- Chapter 4 presents our proposal. First, we demonstrate our basic solution, named Time Table Diffusion protocol for synchronizing clocks of nodes in vehicular environments. The proposed solution works in distributed manner and allows nodes to run freely with the value of their local clocks, but store the time information to synchronize other nodes when it is needed. By exploiting the broadcasting channel, TTD reduces the communication overhead and the number of messages required to achieve the synchronization. Secondly, as failures of clock synchronization has the potential to cause safety applications to fail, the offsets Table Robust Broadcasting (OTRB) protocol, which is a result of TTD enhancement, is proposed to overcome both communication and clock failures that can appear in the previous model. Finally, we end the chapter by giving performance evaluation.

- Chapter 5 deals with OTRB performance improvement based on clustering mechanism to reduce the number of communications caused by the high frequency of arrival/departure of nodes. To this end, we adopt the three clustering algorithms in [8–10], and give a comparative analysis of the influence of these algorithms on OTRB performance considering parameters, including the average of arrival/departure and the number of isolated nodes.

- Finally, we conclude our thesis with a brief discussion and future perspectives.

Background

Chapter 1

Vehicular communications

1.1 introduction

The increasing of vehicles on roads causes the number of accidents and traffic jams to enlarge. The demand of new way to control the transportation system is thus highly needed. To this end, Vehicular Ad hoc NETwork (VANET) has been created. VANETs enable versatility of applications that improve roads safety, driving amusement and time-road-fuel exploitation. They can be seen as the application of mobile ad hoc networks (MANET) for road traffic information sharing by enabling data communication through wireless and mobile networks with little or no infrastructure. VANETs provide communication between the vehicles themselves and/or between vehicles and the infrastructure installed in roads (see Figure 1.1). Basically, VANETs promises providing safety and non-safety applications to the classical transportation systems. The safety applications, such as collision detection and lane change warning, aims to provide the safety of passengers on roads. On the other hand, non-safety applications provide additional services to improve comfort and fun to the road users. The non-safety applications include, but not limited to: Internet access, weather information, e-commerce and route guidance.

Owed to system performances they afford to improve, VANET has been drawing attention of both industrial and research communities. However, put VANET systems into practice should meet the challenges their characteristics impose. For example, in addition to congestion problems due to the high number of connected vehicles, there is the high frequency of topology changes due to the high velocity of nodes.

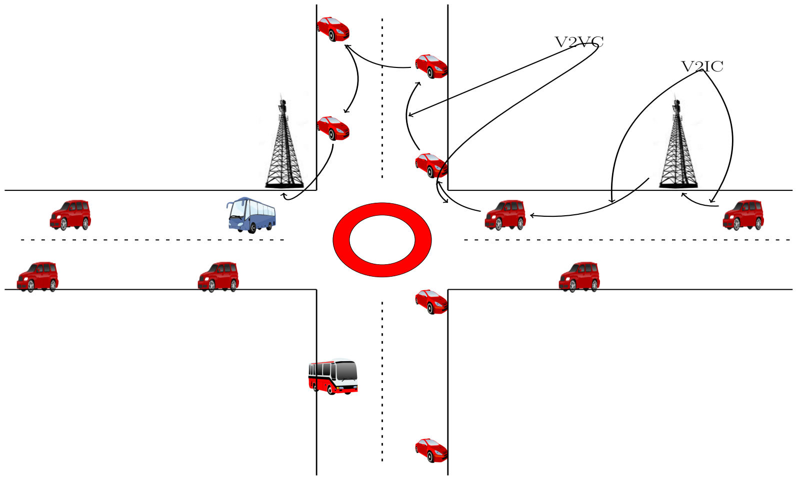


Figure 1.1: Vehicular communications.

Even more, the continuous increasing of connected entities in roads, such as vehicles, smartphones, tablets and sensors makes the real practice more challenging. The wide range of applications of vehicular communications opens up new research topics, such as the Internet of Vehicles (IoVs) and the Vehicular Cloud Computing (VCC).

The present chapter presents the fundamental concepts of vehicular communication systems. Here, we point out on the architecture, characteristics, challenges and requirements that face such systems. Also, new open topics of research related to the former are discussed.

1.2 VANETs' definition

YOUSEFI et al. defined VANET as "*computer network on wheels*" [11]. It is a network that provides data communication among vehicles as nodes. These vehicles communicate with each other, as well as with the Infrastructure installed in roads.

VANET is characterized by the movement of its nodes in an open space area with restricted direction and speed. In fact, vehicles move according to a mobility pattern, based on predefined roads, buildings, intersections, junctions and other traffic entities, in an urban, rural, or highway area [12]. The connected nodes in VANET consist of vehicles on roads able to data storage and processing. Usually, these nodes are equipped with sensing devices to collect the necessary data. Beside to GPS, the constructors of today have already equipped the new cars with multiple sensors. Among the sensors are found, radio, radar, camera, speedometer, temperature, fuel gauge, rain, wheel rotation and tachometer sensors. Also, mobile phone tends to be increasingly connected to the vehicle. These different sensing devices permit events and environment data collecting. In infrastructure mode, the collected data is spread up through Vehicle to infrastructure communication (V2IC). In turn, the other vehicles get the information needed from the available Road Side Unit (s) (RSU(s)). By the contrary, in ad hoc mode, the vehicles exchange the collected data between them through Vehicle to Vehicle communication (V2VC) in multi-hop manner. Also, to get better performances, we consider hybrid communication using both V2IC and V2VC.

1.3 VANETs' applications

VANET systems aim to increase road safety / security and make the end user more comfortable and having more fun. An extensive works providing a list of applications and services to this end have been compiled, such as cooperative driving, wireless remote diagnosis and collision warning. However, the overall of these applications can be divided into two major categories; safety and non-safety applications [13].

1.3.1 Safety applications

Safety applications focus on reducing the damages (such as the loss in humans' life) caused by the collisions and accidents commuting in roads. It is stated in [14] that if a warning is given half a second before the moment of the collision, sixty percent of the accidents could be avoided. Using inter vehicle communications (IVCs), a vehicle recognizing dangerous situation should alert its neighboring. Reporting such situations should be as fast as possible to provide immediate reaction against possible hazardous. The slow warning propagation decision causes late alert delivery. In turn, this results on unavoidable collision. There is a considerable amount of research in safety applications in order to improve the safety and security of transportation systems. According to Vehicle Safety Communication (VSC) consortium, the safety applications include eight high potential applications:

- *Traffic signal violation warning*;
- *Curve velocity warning*: Curve velocity warning applications guide the driver

about speed limit to avoid collisions;

- *Pre-crash sensing*: Warning systems can be deployed to avoid accidents, e.g., work zone warning, stopped vehicle warning, and low bridge warning for trucks. Such applications may also manage traffic flows and identify alternative routes;

- *Emergency electronic brake light*: Emergency brake announcement is the most important application for crash prevention;

- *Cooperative forward collision warning*: Using V2VC, cooperative forward collision warning to send emergency notifications to nearby emergency responders is most likely after accident;

- *Left turn assistant*: A driver getting ready to make a left turn might easily overlook a pedestrian crossing on the right side of the street. Therefore, the number of accidents can also be reduced with the help of early warning system;

- *Lane change warning*: Lane change warning applications are used to provide assistance to a drive about lane change by applying automatic emergency breaks;

- *Stop sign movement assistant.*

1.3.2 Non-safety applications

Despite the fact that increasing roads safety is the main motive behind IVCs, these latter provide supplementary applications in order to achieve better transportation systems efficiency. Non-safety applications offer value added services that increase the market penetration. Among these services, let us site as examples:

- *Infotainment and commercial applications*: these applications facilitate individuals and tourists traveling. They aid the end user to map for restaurants, petrol pump, finding nearest parking availability, navigation and route guidance.

- *Mobile internet access and peer to peer applications*: internet connectivity became a requirement for multiple usages. Passengers in their vehicles could download music, movies, and games and connect to home computers. In the absence of fixed internet connectivity, vehicles themselves can provide internet access to other vehicles. Peer to peer applications are also of vital interest. They able data sharing between the moving vehicles, such as sharing music, games, videos and chatting.

- *Road traffic optimization applications*: this type of application comes up the road user with real-time information in order to deal with minimizing congestion. Traffic information is most likely to help driver to find best route to destination when it relates to the immediate area.

1.3.3 A Literature review of VANET's projects

A number of projects that have been started are enlisted in order to roll out VANETs in real life. Let us site as examples:

i) *ACDC (2014-2016):* ACDC is a Swedish national project which aims at supporting autonomous cooperative driving with dependable wireless real-time communications. The project considers two application scenarios: platooning (road trains), such as joining in the middle of the platoon to arrange for best fuel saving; and fully autonomous driving in a confined area like a construction site, a harbor, or a mine.

ACDC project attempts to respond the following questions:

(a) How can wireless communication enable/enhance autonomous cooperative driving?

(b) What requirements on the communication for applications?

(c) How to design and configure communication protocols and methods to fulfill the requirements on wireless real-time communications?

ii) *HIGHTS (2017-2020, [1]):* HIGH PRECISION POSITIONING FOR COOPERATIVE-ITS is a European research project supported by European Union, which focus mainly on smart, green and integrated transport for the cooperative ITS. This project addresses these problems by combining traditional satellite systems with an innovative use of on-board sensing and infrastructure-based wireless communication technologies (e.g., Wi-Fi, ITS-G5, UWB tracking, Zigbee, Bluetooth, LTE...) to produce advanced, highly-accurate positioning technologies for C-ITS. The goal of the HIGHTS project is to achieve high precision positioning system with the accuracy of 25cm.

[1] `http://hights.eu`

iii) *SCOOP (2014-2018,* [2]*):* SCOOP is a Cooperative ITS pilot deployment project that intends to connect approximately 3000 vehicles with 2000 kilometers of roads. Vehicles are equipped with sensors to detect events such as a slippery road, an emergency brake, etc. and with on-board units to transmit the information to vehicles behind (V2V) and to the road operator (V2I) through road side units. The road operator can also transmit information (road works, etc.) to the vehicles through their on board units (I2V). The main objective is to improve the safety of road transport and of road operating staff during road works or maintenance.

iv) *UK Autodrive (2015-2018,* [3]*):* UK Autodrive is a UK project that is trialing automated vehicle technology as part of a government-backed competition to support the introduction of self-driving vehicles. The trials will:

(a) Integrate autonomous and connected vehicles into real-world urban environments.

(b) Show how autonomous and connected vehicles could solve everyday challenges such as congestion.

(c) Demonstrate the commercial operation of electric-powered self-driving at a city scale.

(d) Provide insight for key stakeholders and decision-makers, including legislators, insurers and investors.

1.4 VANET's characteristics, challenges and requirements

VANETs reveal the application of MANETs in the transportation domain. However, the characteristics of the environment in which the network progresses create new features of the former. For instance:

- *Velocity:* The velocity of the moving vehicles may cause frequent change in the network topology. Therefore, a high rate of arrival and departure of nodes can be affected. Dealing with such a limitation drains an important communication

[2] http://www.scoop.developpement-durable.gouv.fr/en/

[3] http://www.ukautodrive.com/

overhead. That, nodes need frequently to choose a reliable route in order to ensure data deliverance to specific destinations.

- *Road traffic:* In high traffic, the collision caused by the communication in progress cause the latency to enlarge. This issue can be solved by using short communication range. On the other hand, the nature of traffic makes some isolated vehicles in the roads. The usage of short communication range palliates the interferences issue; however the transmission range employed cannot reach further destinations.

- *Road infrastructure:* The high building and the intersections may affect traffic jams and fading. In addition to the interference, the hidden node problem is pointed out as a major limitation. The design of VANET in these features is more challenging. Even so, the control algorithm should consider the change in the mobility pattern (urban to highway e.g).

The outcome of VANETs applications and services is affected by several circumstances. To succeed in their missions, the design of VANET systems should meet the following requirements:

- *Scalability:* VANETs applications, such as safety applications, demand the correct behavior in sparse network as well as in traffic jam scenarios. However, the design of VANET systems has to cope with any increase in the traffic density.

- *Latency:* The nature of VANET system requires the immediate relaying of information, without the introduction of any delay which can be caused by different raisons, such as high traffic and network collisions.

- *Reliability:* Failures can independently appear at any time. Reliability referred as to the possibility of continuous running with and without failures.

- *Availability:* The availability defines the probability that the system is available at any time. The Mean Time Between Failure (MTBF) should be as short as possible. The shortest MTBF, the best maintainability is offered.

- *Safety:* Safety is a property of a system that reflects the system's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment. The importance of safety requirements is owed to the ability to exclude undesirable situations.

- *Security:* Security reflects the ability of the system to protect itself from accidental or deliberate external attack. The system security is an essential pre-requisite to ensure its availability, reliability and safety.

- *Quality of services (QoS):* Real-time distributed systems such as VANETs must satisfy timing, reliability and security constraints as well as application-specific quality requirements to deal with scheduling of the integrated services in real-time. To address this diversified problem, QoS requirements have to manage the network resources to provide the QoS guarantees in VANETs. The most robust VANET system provides better QoS.

1.5 VANETs layered communication architecture

Communication protocol stack is of vital importance to ensure reliable communication between any pair of nodes in the network. It is referred to as the communication layered architecture. This latter designs the important communication protocols and services, including physic, Medium Access Control (MAC), network (routing), application and security, ...etc (see Figure 1.2).

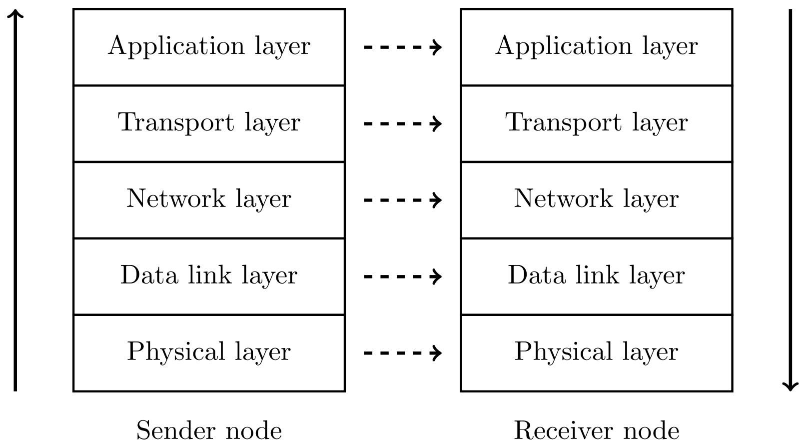


Figure 1.2: Layered communication architecture.

In order to ensure their interoperability, the design of cross layered architecture oriented VANETs has paid a tremendous attention. Many research industries address this issue, such as, the International standard Organization (ISO), the Institute of Electrical and Electronic Engineers (IEEE) and the Car-to-car Communication Consortium (C2CCC) as main actors.

i) CALM by ISO:

ISO proposal was designed to manage communication between vehicles as well as between vehicles and RSUs. It is referred to as the Continuous Air-interface for Long to Medium range (CALM). CALM proposal is based on the Dynamic Short Range Communication (DSRC) concept. DSRC was approved by US Federal Communications Commission (FCC) and works under the frequency band 5.9 GHz. The spectrum is divided into six service channels (SCH) and one control channel (CCH) with equal bandwidth of 10 MHz each. For emergency and control messages, CCH is used. SCH is used for other application packets. Figure 1.3 illustrates the proposed ISO CALM layered architecture.

<table>
<tr><td colspan="2">Application layer</td></tr>
<tr><td colspan="2">Transport layer</td></tr>
<tr><td>IPv6</td><td>Network layer</td></tr>
<tr><td>MAC layer</td><td rowspan="2">Interface layer</td></tr>
<tr><td>Physical layer</td></tr>
</table>

Figure 1.3: ISO CALM communication layered architecture.

At physical layer, CALM standard provides combination of different access technologies for communication. For short and medium distances, CALM takes advantages to infrared. In contrast, for long distance, it prefers the use of GSM, UMTS or any other available wireless access technology. In addition, it has the option to incorporate future technologies. As a result, ISO CALM may provide vehicles to other interfaces communication. CALM architecture also defines communication interface, network and application management features. The network manager enables IPv6 networking to move from a subset to another with handover to alternate media. CALM applications manager ensures application transmission requirements. It interacts with interfaces to get information about most suitable medium and instructs the network manager to establish a new connection. IETF NEMO protocol is concerned with managing the mobility of a network.

ii) WAVE by IEEE:

IEEE has proposed the 1609 family of protocols for VANET communication purposes. It is named as WAVE for Wireless Access in Vehicular Environment. The overall of WAVE communication stack is working based on DSRC radio technology. That is standardized by IEEE community as IEEE 802.11 p. Currently, DSRC is the frequency band approved by all the standards [15]. IEEE 802.11 p details the MAC layer operation in WAVE proposal. The former permits V2I communications using the WAVE Basic Service Set (WBSS). This service is based on IEEE 802.11 a specification to achieve communication between a node and the available access point. For V2V communications, the WAVE Independent Basic Service Set (WIBSS) is approved.

As shown in Figure 1.4, the proposed WAVE layered architecture consists of six standards.

(a) IEEE 1609.1, WAVE applications manager, defines the basic applications and their requirements.

(b) IEEE 1609.2, WAVE security services, defines the security considerations, such as anonymity, authenticity, and confidentiality.

(c) IEEE 1609.3, WAVE networking services, defines network and transport layer services, including addressing and routing. The standard provides a dedicated single protocol, named as Wave Short Message Protocol (WSMP).

(d) IEEE 1609.4, WAVE multichannel operations, provides frequency band management and coordination between the seven DSRC channels.

(e) IEEE 1609.5, WAVE layer management, deals with layer management.

(f) IEEE 1609.6, WAVE facilities, offers an additional middle layer between transport and application layer.

Indeed, WAVE allows only IEEE 802.11p MAC for all types of communication, which is considered as a limitation for several research activities.

iii) C2CNet by the Car to Car Consortium (C2C-CC):

The C2C-CC protocol architecture proposal aims to achieve an open European stan-

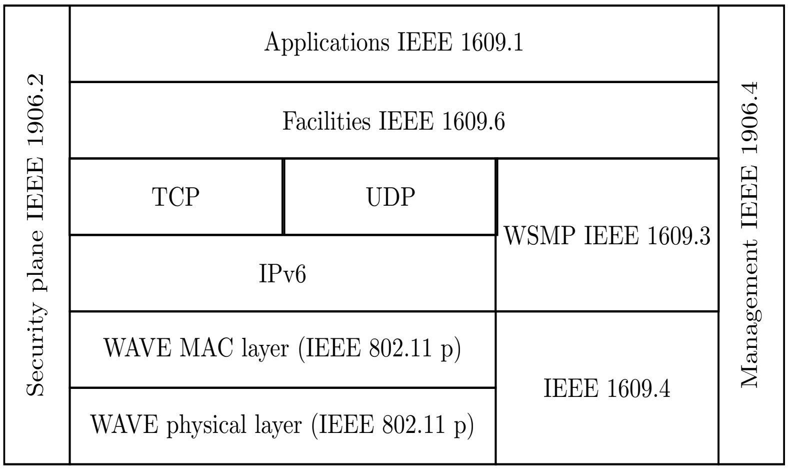


Figure 1.4: WAVE IEEE communication layered architecture.

dard, where, the interoperability among cars from different manufacturers should be allowed. Basically, the aim is to achieve real-life demonstration of safety applications in vehicular environments. The proposal architecture supports many available interfaces at physic and MAC levels. A dedicated bandwidth of 30 MHz will be available for safety applications. Also, its design for the network layer (C2CNet) supports both infrastructure-based and infrastructure-less communication modes. Differently to IP protocol, C2CNet provides multi-hop ad hoc communications based on geographical addressing and routing with location service demonstration. However, C2C-CC transport and network layers allow the use of traditional networking as well as C2CNet below IP stack. Figure 1.5 below depicts the C2C-CC layered architecture.

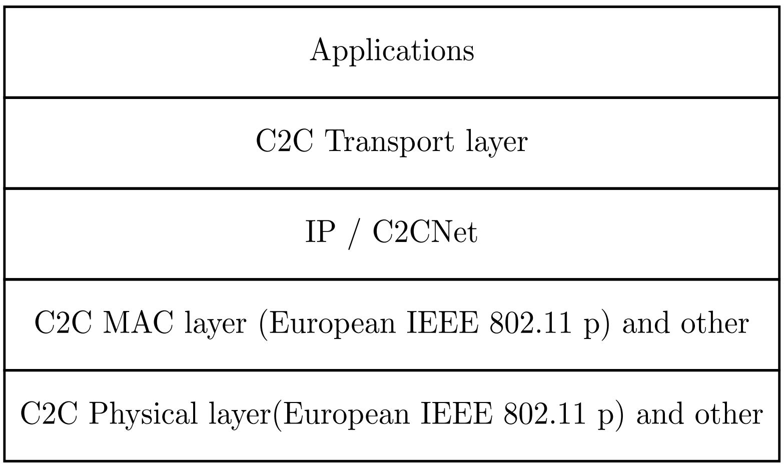


Figure 1.5: C2CNet communication architecture.

The overall of communication protocols stack must be revised in the context of VANETs. In the following, a discussion about the protocols of the different layers of the communication architecture is given.

1.5.1 Physical layer

The lowest layer in the communication protocol stack, physical layer, defines the mechanical and electrical/optical specifications of the physical transmission medium usage. It describes the transmitted data (bits) encoding and the synchronization rules. Experimental studies of the usage of both radio (very high frequency (VHF), micro and millimeter waves) and infrared waves for IVCs have been made in order to determine their efficiency. It is stated in [16] that infrared and millimeter waves are more suitable for line-of-sight communications, while VHF and microwaves provide broadcast communications. However, as VHF provides long links with low speed, the trend is to use microwaves [17].

Most of the researches rely on the physical layer of $IEEE802.11$ standard. The defined $IEEE802.11p$ for IVCs, well known as DSRC WAVE, operates in the $5.850-5.925$ frequency band in the US, and in $5.855-5.925$ frequency band Europe. In the physical layer for VANETs, the physical medium has to offer robust communication between the moving vehicles. Especially, in case of emergency where high QoS is required. It has to consider multi-path fading and Doppler frequency shifts caused by the mobility of vehicles. To mitigate (handle) these requirements, the physical layer of IEEE 802.11 p doubled the physical parameter in the time domain. So that the inter-symbol interference caused by the multipath delay spread and the Doppler spread effect was decreased [16]. Orthogonal frequency division multiplexing (OFDM) is used with data rate from 3 to 27 Mb/s. IEEE 802.11 p targets a transmission range between 300 and 1000 meters and supports vehicles' speed up to 55 m/s (200km/h). The variation in the density of nodes in different vehicular environments affects the supported transmission range. For example, in traffic jams, the usage of large transmission range may cause considerable increasing data loss due to the interferences. On the other side, when the number of moving vehicles is low, the number of isolated nodes will increase with low transmission range. To deal this problem, the node has to adjust adaptively its transmission power. That is, by increasing power when number of neighbors is small and decreasing power when the number of neighbors is large. Many VANETs' projects propose the use of the physical layer of cellular technologies instead of IEEE

802.11. Thus, the bit error rate has been improved compared to solutions based on IEEE 802.11 b [16]. For example, the European FleeNet project [18] has proposed to use the universal terrestrial radio time division duplexing (UTRA-TDD UMTS 3G) based physical layer. The offered data rate is of 2Mbps. Furthermore, 5G is expected to improve the data transfer speed, scalability, connectivity, and energy efficiency of the network [19]. However, it is stated in [20] that the use of these technologies as future communication base for VANETs is discouraged due to their high latency rate and limited bandwidth.

1.5.2 MAC layer

The main object behind MAC layer is to provide reliable radio medium sharing. MAC layer design for VANETs must overcome certain challenges in order to control channel access and ensure data convey free on error. At first, the nature of the wireless medium led the appearance of the hidden node problem. It is usually shown up when two non-neighbors attempt to transmit to a common one-hop neighbor node (see Figure 1.6). Consequently, transmitted packets can be collided because of the signals' interferences.

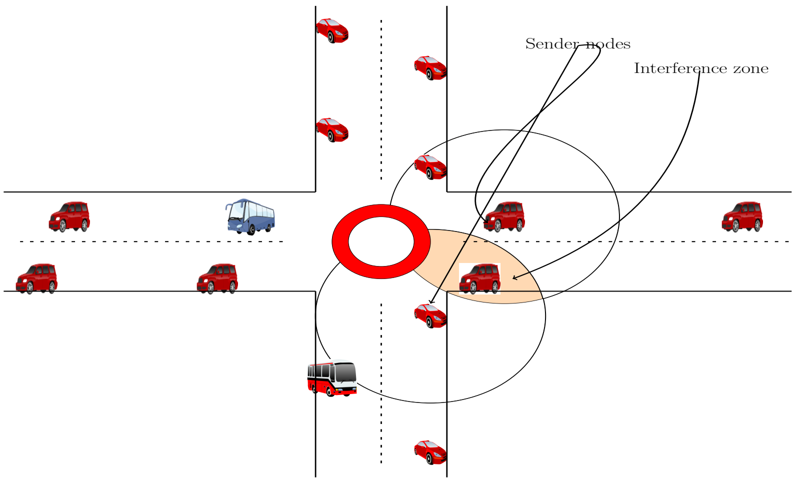


Figure 1.6: Hidden node problem.

The proposed MAC protocol ALOHA [21] uses in random access. A node sends its data and if a collision occurs, it waits for a random period and transmits again. Simulation results show that the maximum data throughput of the algorithm is 18.4 % of the channel access capacity [22].

The carrier sensing multiple access (CSMA) protocol attempts to avoid collisions caused by the random access to the channel. A node first checks the channel medium before sending its packet. If there is an ongoing transmission, it waits for a random time to transmit. Otherwise, the channel is supposed to be idle (free) and the node can transmit. Indeed, it may be possible that more than one node want to transmit at the same time. If the channel is sensed free, these nodes will engage the transmission at the same time. This leads in network collision. To deal with this limitation, the request to send / clear to send (RTS/CTS) handshake is used. Before engage the transmission, the sender broadcasts a RTS packet to warn its vicinity. If the receiver is ready, it responds by CTS packet take on the message exchange between the sender and the receiver. By this way, all the neighbors of the sender and those of the receiver will not corrupt the upcoming transmission. Thus, the sender can send its packet without risk of collision. Nevertheless, this mechanism prevents other nodes sending their data to an uncovered node by the first sender. This is known as the exposed node problem. A node receiving a RTS packet but no CTS is considered as an exposed node (see Figure 1.7).

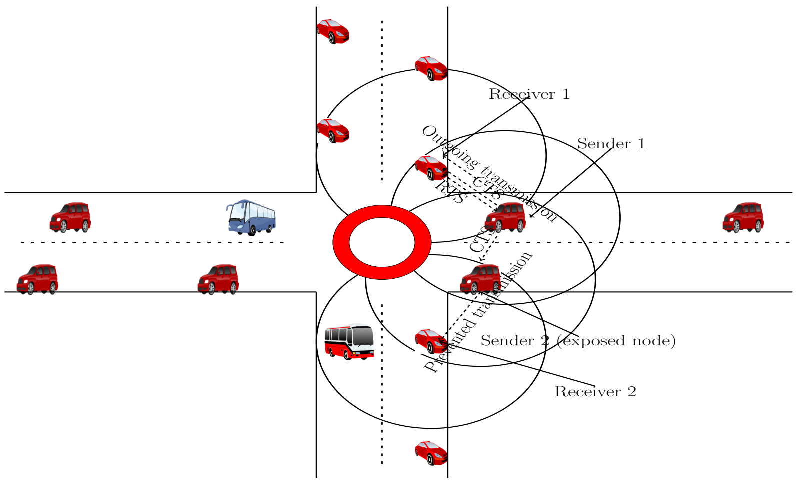


Figure 1.7: Exposed node problem.

Furthermore, if the exposed node is allowed to transmit, it may cause communication collision because of possible failure in receiving the CTS packet. IEEE 802.11 MAC layer standard uses CSMA with acknowledgment packet to indicate the termination of the transmission process. This allowed the exposed nodes to transmit their packets. It also provides robust transmission and minimizes the risk of collisions in data communication.

Time criterion can also be used to improve the shared medium efficiency and to improve hidden/exposed node problem. For instance, time division multiple access

(TDMA) protocols divide the shared medium into several time slots; so that, one time slot is assigned for each node. However, if the data rate increases, the interferences increase as well.

In addition to the hidden/exposed nodes problem, due to the high mobility, vehicle nodes may join/leave a group of nodes at any given time. Therefore, the network topology is exposed to frequent changes. Moreover, vehicles may pass from highly dense to sparse environment during their movement. MAC protocols design for VANETs must deal with those challenges. It must easily adapt to the frequent change in the topology and provide effective sharing medium during dense, as well as sparse traffic load. However, in the lack of pre-established infrastructure, vehicles usually need periodic exchange of control messages in order to manage collision-free channel access. This will generate high communication overhead, especially in dense networks. The more the communication overhead, the less medium channel will be effective.

In order to easily adapt to the physical layer based on UTRA-TDD, the European FleeNet project suggests the use of the Reliable Reservation ALOHA MAC protocol (RR-ALOHA). The RR-ALOHA [23] is an improvement of the standard ALOHA MAC protocol. The channel access is divided into several time slots, so that, the forerunner of this algorithm attempts to send its data in the beginning of the time slot. The maximum throughput is thus doubled compared to the initiative ALOHA. RR-ALOHA added an additional mechanism in order to inform nodes of the status of the slot (idle (free) or busy (used)) at two hops. A slot can be reserved only if it is not used by any nodes two hops away. However, it is very difficult to evaluate the performance in highly mobile networks, where the management of two-hop neighbors is a challenging task. In addition, as the protocol uses fixed time slots, the number of the vehicles in the communication range must not exceed the number of time slots. MAC protocols in VANETs should also consider the requirements of the application for which data exchange (a transmission) occurs. On the one hand, while comfort applications, that aim to increase passengers comfort and entertainment, need high data rate; safety applications require that the information reach the highest number of nodes in a critical time. On the other hand, safety application messages have higher priority than comfort application messages. Thus safety application messages must be differentiated from other messages.

IEEE 802.11 standard proposed the DSRC-based MAC protocol. DSRC supports channel access in both safety and non-safety applications. The standardization of DSRC is known as 802.11P (the Physical and MAC layer of the IEEEP1606 standard family implemented by WAVE). IEEE 802.11P adopts the CSMA handshake with collision avoidance (CSMA/CA). In addition, it uses the enhanced distributed channel access (EDCA) mechanism. This mechanism divides the messages (packets) considering their priority using Access Category (AC) parameter. The message with the highest AC value is the highest prioritized message. Based on AC, an amount of waiting time is adjusted within the appropriate contention window. So that, the highest priority message has less waiting time to access the sharing medium channel. Also, an Arbitration Inter-frame Space (AIFS) is adjusted considering AC. If the channel access is sensed idle for AIFS duration, the vehicle is permitted to transmit. DSRC divides the channel into several synchronized intervals (SI), $100ms$ for each SI. Further, for each SI there are the control channel interval (CCHI) and the service channel interval (SCHI). Each channel interval is of $50ms$ and they are separated by the guard interval (GA). The CCHI is used for safety application messages exchange, while the SCHI is used for the other services. During CCHI, all the vehicles broadcast and listen to messages from neighbors and RSUs. When network density increases, CCHI is not enough to control all safety messages transmitted by all the vehicles [24]. By contrast, when the network density decreases to be less than the bandwidth, CCHI is wasted.

Due to the use of the CSMA/CA mechanism, DSRC-based MAC protocols are not suitable for real-time traffic because they cannot guarantee a determined upper bound delay. Many researches in the current literature take the advantages of using the combination of DSRC-based and RR-ALOHA-based MAC protocols in order to improve the performances of the shared medium and adapt to the real-time traffic and mobility conditions [25]. Let us cite as examples [24, 26–28].

1.5.3 Network layer

The principal mission of the network layer consists on data routing from a source to a destination node. It has to provide the most optimized route that increases the network performances as possible. Vehicles' mobility presents big challenges that should be considered during the routing operation. Although, VANETs are linear networks and the movement of the vehicles may be predictable, Source, destination and relay nodes may rapidly change their position over the time. The routing protocol should be able to update or select new routes in order to ensure that the transmitted packet reaches its destination. It should also able to cope with the different types of traffic densities. While covering disconnected nodes must be handled in case of sparse traffic, scalability is a mandatory parameter to evaluate routing protocols under dense traffic conditions. Moreover, the routing protocols must cover the wide range of applications supported by VANETs. To this end, data routing process have to consider different network conditions. Among these conditions; position, velocity, road conditions, traffic information and message' priority. However, the design of a routing protocol that suits all scenarios and conditions of vehicular environments remains a debate task.

Generally, there are two main categories of routing protocols in MANETs; proactive and reactive (on-demand) routing approaches. In proactive routing protocols, each node maintains updated lists of destinations and their routes (routing table). Despite the fact that this category provides low latency in order to deal real-time traffic scenarios, there is a remarkable waste in network resources due to the high cost generated to maintain fresh routes. On-demand routing protocols try to find a route from a source to a destination at run time. The high latency time taken to find the route is the major limitation of on-demand routing.

Non-real-time traffic applications can use either unicast or multicast routing protocols. Yet, in real-time traffic applications, broadcasting protocols are highly required to flood data efficiently and reliably [16]. When unicast routing protocols perform data exchange from a source to a destination, multicast routing protocols target a set of destination nodes. The destination nodes are generally known by their locations and can be reached via multi-hop wireless communications between relay nodes. Either unicast or multicast routing protocols need to find the appropriate route, whether by implementing proactive or on-demand routing mechanisms. Whereas, metrics, such as geographic position must be used. The broadcast routing protocols spread the information to the entire network. A source broadcasts the packet to all its neighbors. In turn, each neighbor, when it receives the packet for the first time, will broadcast it again until reach the destination. Routing issue in VANETs has attracted a potential attention of the research communities. The authors of [29] survey the existing routing protocols up to 2017. For each protocol, they give their advantages and disadvantages. The paper's taxonomy divides the routing protocols in VANETs into five basic classes; Topology based, position based, cluster based, broadcast and geocast based routing protocols.

Topology based routing protocols, such as [30–34] require immediate knowledge about the network topology changes in order to maintain reliable routes. These protocols usually use the shortest path algorithm in order to provide a short route between two nodes. In case of proactive routing, the high cost in communication overhead generated in order to maintain fresh routes may degrade the network performances. On the other hand, in case of reactive routing, the delay introduced to search an optimized route is very important. Further, due to the high mobility of nodes and the nature of the network partitions, it may be possible that a route cannot be available between a source and its destination. The use of link lifetime prediction techniques in order to monitor the network connectivity may minimize the effects of network partitions [35]. Each node, thus must know its own location, velocity and the global time. It must also know the approximate positions and velocities of its one-hop neighbors. However, prediction becomes more complex in some traffic loads, such as cities' traffic [35].

Position based routing protocols rely on the geographic position information of the nodes to route data. Each node is supposed to know its own position, the position of its one-hop neighbors and that of the destination. This can be possible by the aid of any localization system, such as GPS and Galileo. The source node attaches the destination's position to the packet. The next hop then is selected based on the geographical location of neighboring nodes. The next relay must maximize the progress of the packet toward its destination. In addition to optimized route they offer, position based routing protocols cope well with dense and mobile networks. However, geographical routing can form routing loops or a packet can travel longer route due the network partitioning [13]. Examples of position based routing protocols can be found in [36–38].

Cluster based routing protocols [39, 40] divide (partition) the entire network into small manageable clusters. Several metrics can be used to cluster the network, so that, vehicles in the same cluster have common characteristics, such as geographic zone, direction and velocity. For each cluster, it is defined a cluster head (CH) which is responsible for inter and intra-cluster communications. However, the clustering in VANETs itself is considered as an issue that should be discussed. A recent survey papers that provide detailed description of clustering algorithms in VANETs can be found in [41–43].

Broadcast routing protocols [44–46] are mostly used to share information that should reach the entire network, such as traffic information and road conditions. Also, these protocols are implemented in route discovering phase of topology based protocols. Protocols implementing broadcast routing protocols must overcome the broadcast storm problem. It comes up when large number of nodes broadcast the same packet at approximately the same time.

Geocast based routing protocols [47, 48] are a special case of multi-cast mechanism. A message is flooded to a group of nodes in a particular geographic region.

1.5.4 Transport layer

The main task of the transport layer consists on the segmentation of the data stream in order to relieve congestion. The protocols of the layer provide end-to-end communication services for the application layer.

The dynamicity of VANETs and the intermittent connectivity cause in packet loss and congestion due to transmission errors. This can greatly decrease the network throughput. The transport control protocol (TCP) for wired TCP/IP stack is not suited to mobile networks exposed to frequent topology changes, such as VANETs [49]. Several modifications were performed in order to adapt TCP protocol to VANETs, such as [50, 51]. Unfortunately, the inadequacy of these protocols to applications that need multicast routing is apparent [52]. Therefore new approaches are highly required, which remains a challenging task.

1.6 Security plane

Since their principal aim is to conserve (protect) human life, VANETs applications require that the circulating information in the network to be very accurate. Indeed, in open space related to VANETs environments, it is easier to inject false information. For instance, an adversary node could announce false accident information. This could in fact cause real accident due to the emergency braking. Also, the adversary node could gather personal information about the users, their locations and mobility patterns. Security plane protects systems from thefts and damages in order to prevent any disruption or misdirection of the services they provide. To this end, the security plane has to ensure the following properties:

- Confidentiality: Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.
- Integrity: Integrity is the property of safeguarding the accuracy and completeness of assets (ISO/IEC 27001, 2005).
- Availability: Availability is the property of being accessible and usable upon demand by an authorized entity (ISO/IEC 27001, 2005).
- Authenticity: Authenticity is the property that an entity is what it claims to be (ISO/IEC 27000, 2009).
- Non-repudiation: Non-repudiation is the ability to prove the occurrence of a claimed event or action and its originating entities, in order to resolve disputes about the occurrence or non-occurrence of the event or action and involvement of entities in the event (ISO/IEC 27000, 2009).

Though security issue was extensively studied in wired and wireless networks, it is more complicated to ensure security of VANET systems due to their intrinsic characteristics. So that scalability, frequent topology changes, high mobility, and variety of applications in VANETs make the security concern more complicated. Thus, security problem must be carefully undertaken in the context of vehicular communications. Where, performances in terms of real-time delivery and low-traffic and processing overheads should be reached.

The authors of [13] enlist the types of the possible attacks occurring in VANET systems (see Figure 1.8).

Certification mechanism could be used to provide vehicles authentication. Vehicles and their drivers could be managed well defined certification authorities (CA). Each node in the vehicular system would have a unique identity (ID), public key, private key and a certificate. There also would be cross-certification between the different CAs. For

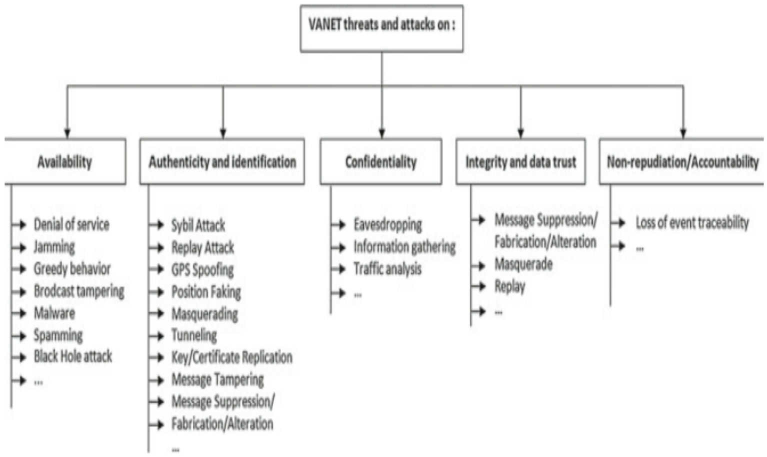


Figure 1.8: VANETs' threats and attacks.

communicating safety messages, IEEE 802.11p standard proposes to use asymmetric cryptography.

1.7 VANETs simulation and mobility modeling

Validation of proposed solutions is very important to pass from theoretical to practical aspect. Network simulation is one of the most used techniques in order to predict the performance of networks before they are physically built or rolled out. Actually, accurate VANETs simulation has to consider the peculiar characteristics of the moving vehicles, i.e., the nodes mobility model. Using random models where the speed and the position of nodes change randomly does not reflect real evaluation of the protocols designed to vehicular environments. That is, vehicles mobility is restricted to the roads and the interaction with other vehicles (acceleration, deceleration and lane change). Some studies, such as [53, 54] show that realistic traffic modeling is necessary to realize accurate vehicular simulation. Therefore, VANET networks simulation should be realized via specially-designed VANETs simulators. The designer could either use a traffic simulator for generating realistic vehicular mobility traces that will be used as the input for a mobile ad hoc network simulator, such as NS3, NS2, OMNET, etc. Among the mobility generator in VANETs they are: VANETMOBISIM, SUMO and RoadSim. These mobility generators, called also vehicular traffic simulators, rely on a traffic flow theory. Traffic flow theories explore relationship among vehicles density, velocity and flow. Where, the density represent the number of vehicles per unit distance, the velocity is the distance that a vehicle travels per unit time and the flow is the number of vehicles that pass an observer per unit time. Traffic simulation models can be classified into microscopic, mesoscopic and macroscopic mobility modeling. Microscopic mobility modeling considers flow of vehicle in detail, such as acceleration / deceleration, driver's behavior, car's length, car's speed. Macroscopic properties are global in nature. When large number of vehicles is in motion, global parameters can be represented as flow (mass and density). Mesoscopic modeling represents an intermediate modeling level of traffic flow. It may lead to efficient trade-off between modeling of individual vehicles and the modeling of large quantities of vehicles. This model may take different forms such as the modeling headway distance as a average of large number of vehicles sometimes size or density of cluster of vehicles. This model includes dynamic behavior of vehicle and macroscopic properties such as density and velocity for large number of vehicles. For instance, VANETMOBISIM utilizes the Intelligent Driver Model including Lane Change (IDM-LC) mobility model. IDM-LC implements road intersection supervising strategy: making vehicle nodes slowing down, stopping or moving in accordance with traffic lights. They are also capable to carry out overtaking to change lane in multi-lane roads.

1.8 Related research topics

1.8.1 Vehicular ad hoc sensor networks (VASNETs)

Vehicular Ad hoc Sensor Networks (VASNETs) are a fusion of WSNs and VANETs. Therefore, nodes can be vehicles on roads, sensors carried by vehicles and road side sensors (RSS) deployed besides highway roads. Sensor nodes communicate the sensed data, such as vehicles' velocity through RSS nodes. These latter may act as relay nodes to forward data towards the installed infrastructures. They may further act as cluster heads (CHs) for the moving vehicles. VASNETs use the features of sensors located in vehicles and roads for vehicular environments. Therefore, they inherit their characteristics from both sensor and vehicular networks. However, the sensor nodes receive significant power from the vehicles' battery. These nodes are not restricted to a limited energy. But, RSS nodes have no such a source of energy. Power consumption thus must be considered for a reliable deployment. On the other hand, vehicular environments are opened on a huge amount of real-time relevant data that attract the sensor nodes, such as vehicles' density and velocity. For instance, these data are mandatory for administering traffic safety in order to reduce the number of accidents and warn the driver about the danger that may face. Scalability feature has to overcome the problem of sensing, gathering, performing some treatments and communicating data in enormous and movable networks, such as VANETs.

1.8.2 Vehicular cloud computing (VCC)

Mell and Grance define the cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources, known as data centers, such as networks, servers, storage, applications, and services. These data centers can be rapidly provisioned and released with minimal management effort or service provider interaction [55]. The fast development of ITS and the integration of cloud computing for mobile networking creates the new concept of vehicular cloud computing (VCC). VCC provides traditional and emergent services in order to improve traffic management and real-time information distribution in vehicular environments. Nodes storage their data in the cloud to be used as input to various applications. The communication and the transfer of the information is carried out by the aid of V2V and V2I communications through cellular communication devices (WiFi, WAVE, WiMAx, 4G, 5G). Vehicles, such as ambulances, cars in garages, fire-trucks can serve their resources to be used by the VCC as data centers or service providers. Potential services that VCC provides include [55]:

- Network as a service (NaaS): Vehicles having internet access facility can provide internet access to other vehicles while moving on the roads.
- Storage as a service (STaaS): Vehicles having large storage capacity can provide storage pool to other vehicles in order to allow them running their applications and services.
- Cooperation as a service (CaaS): Vehicles having services subscribed provide necessary information to other vehicles which the drivers express their interest for the same services. CaaS allows drivers to evolve by the minimal infrastructure in order to obtain services.

- Entertainment and Information as a Service (ENaaS / IaaS): Vehicles on the move provide information for safe driving, such as road conditions, advance warning, road crash or any emergency event information. This service is knows as Information as a Service (IaaS). In addition, entertainment as a service (ENaaS) allows many commercials to come to the car screen of the driver, such as advertisements and movies.

1.8.3 Internet of Vehicle (IoV)

The number of connected devices, such as vehicles on road, sensors, and smartphones on the hand of people has known a continuous growth. The real-time communication of these things among advanced wireless access technologies creates the Internet of things (IoT) aspect. This integrates smartness into the existing areas, such as healthcare and industry fields. In the transportation domain, it refers to as the Internet of vehicle (IoV). IoV is a result of integration of IoT and VANETs which aim to enhance safety and efficiency in roads traffic. IoV evolve through vehicle-to-X (V2X) communications, where, X may be another vehicle, RSS, RSU or any other device. Interoperability of heterogeneous devices is one of the main challenging tasks in IoV development. Existing computing and communication devices would be compatible with vehicular networks of IoV. In addition to the smart safety management, IoV will offer smart commercial and infotainment applications to vehicular communications. It will also provide reliable Internet service in vehicles owed to the inclusion of V2X communications. However, the processing and decision making capability must deal with the enlarge in the size of vehicular networks and volume of data. Vehicular communications will generate large amounts of real-time data.

1.9 Conclusion

Vehicular communications are one of the emergent technologies that attract growing attention from academia and industry in the last decade. This is due to the vital role they play in mainly increasing safety of the intelligent transportation systems (ITS). VANETs enable V2V and V2I communication in order to develop efficient and safe future transportation systems. This chapter serves as both an overview of the current vehicular communication systems and an outlook for their future applications. An introduction to vehicular communication is given. Related definition of VANETs with their applications is given. Since their peculiar characteristics create new challenges and requirements, we discuss the overall of VANET's issues; the layered communication architectures, physical layer, MAC layer, network layer and the security plane. In addition, VANETs simulation and mobility modeling are introduced. Finally, open research topics related to vehicular communications, such as VCC and IoV are discussed.

The next chapter tackles Vehicular communication systems dependability and the clock synchronization as a main contribution to enforce system dependability.

Chapter 2

VANETs Dependability and practical use of clock synchronization

2.1 Introduction

VANETs are the core of ITS, ranging from services they provide to the road user, such as traffic information, route guidance, localization aid service, navigation, multimedia and file sharing, online gaming and internet access, to applications that aim to relieve damages caused by roads collisions and accidents.

Nodes in vehicular environments are subject to different failures that make real-time deployment of such systems a challenging task. These failures are mainly due to hardware failures, unreliable networks, software bugs and human errors. In the context of network unreliability, the mobility of vehicles causes frequent network topology changes, when a vehicle can connect or disconnect to a given set of vehicles in a laps of time, so that communication links are unpredictably created/broken. In addition to communication link errors, the effect of multi-path and fading, communication signals may also face building or any other solid structures, such as trees which disturb communication delivery. Increasing power signal to reach and manage further destinations may cause in signal interferences when the number of vehicles in the roads increase incredibly. In addition, in hybrid VANETs architecture or VASNETs, sensor nodes deployed in roads have very limited source of energy which cannot be recharged or replaced. Energy consumption is related to several factors, among there are the strength of the signal and the environment conditions. Furthermore, unexpected defects, such as malicious attacks and hardware failures, can damage the safety of a system or cause that a system cannot provide its specifications to the user. Since VANET systems are concerned by human life, such hazardous limitations are not permitted. The system should meet its specification by the means of fault-avoidance, fault-tolerance, error-removal and error forecasting, all in transparent manner. In this context, dependability is the element key to ensure reliability, availability, maintainability, safety and security for scalable, maintainable and safe VANET systems.

Indeed, in distributed systems, including VANETs, characterized by the absence of global memory, the clock synchronization is of vital significance to support dependable systems. Clock synchronization provides global timescale for a set of nodes (vehicles and sensors), so that these nodes agree about their time view with an acceptable precision, which is a key parameter for distributed systems to meet their objective (resource sharing and concurrency). Time synchronization is a cardinal requirement for predicting, recovering and correcting failures and unexpected behaviors. For instance, check-pointing mechanisms performed to realize fault-tolerance demand that communication between the nodes to be reliable and predictable in its timing behavior. In other words, clock synchronization is required. Further, the absence of synchronized clocks may be the main cause of failure, such as time-based access to common services.

The first part of this chapter discusses the dependability of distributed systems, its main threats, means and measures in accordance to VANET systems. While, the second part gives light on the vital role that clock synchronization plays to support dependable VANETs.

2.2 Dependability: Definition and related concepts

The dependability is defined as the ability of a system to deliver services that can justifiably be trusted [1]. In other words, the dependability of a system means that this system is able to deliver its services, which the agreed specification is respected. This can be reached by the avoidance of failures that are more frequent and more severe than is acceptable [1].

A delivered service may deviate from its correct function when failures occur. Jean-Claude Laprie [2] chose dependability as the term to encompass studies of fault-tolerance and system reliability without the extension of meaning inherent in reliability. Before addressing dependability of VANET systems, we give some basic and related concepts about dependability measures, threats and means.

- **Measures** The dependability integrates the following attributes that are considered as the qualitative and quantitative measures of system dependability [1]:

  (a) Availability: the probability that a system is available, for correct service, at any given time;

  (b) Reliability: the property of the continuity of correct running with and without failure;

  (c) Safety: absence of catastrophic consequences on the system user and the environment.

  (d) Integrity: the property of the absence of improper system alterations.

  (e) Maintainability: the property of ease failure reparation, self-stabilization, short detection and recovery time; however, the availability of the system is affected by the mean time to failure, failure detection time and recovery time.

  (f) Confidentiality: the property of the absence of unauthorized disclosure of information.

  Indeed, the availability for authorized actions, reliability, safety, integrity and confidentiality are common attributes of dependability and security. However, since security provide the ability of the system to protect itself against deliberate intrusion and externally originated errors; applying security measures to the appliances of a system generally improves the dependability.

- **Threats** The threats of dependability affect the system function and may cause drop in dependability. In the context of dependability threats, there are three terms that should be distinguished [2].

(a) *Fault:* A fault is a bug or a defect that may or not lead to a failure in the system. If the fault is never executed, dormant fault (in contrary to active fault), an error will not occur so that a fault will not exhibit as a failure. Faults can be internal or external to system. Indeed, the presence of an internal fault may enable an external fault to harm the system. Generally, a fault first causes an error and possibly subsequent failure.

(b) *Error:* An error is a deviation of the system from its correct state, because of an active fault, which may lead to its subsequent failure.

(c) *Failure:* A failure is the observable manifestation occurred as a consequence of an active error. A system is said to fail providing its specification when it cannot deliver its intended functionality. When the system is implemented to provide several functions, the occurred failure of one or more than one function may leave the system in a degraded mode. Although this system suffers a partial failure of its functionality, it remains provide the other non-failing services. However, the failure of one service may propagate, as a fault, into other services which uses the output data from the fail-service. Therefore a fault-error-failure chain can be formed, so that, fault leading to error leading to failure leading to error ...etc. However, failures can be classified according to their degree of gravity or according to their permanency. Firstly, According to the degree of gravity, we distinguish fail stop, omission and Byzantine failures. In fail stop, the system stops providing some services in the presence of failures. In the case of omission failures, which occurred in network level, messages can be lost and not delivered which may ruin the provided services. While, a system with byzantine failures may exhibit arbitrary behavior and provides services in an arbitrary passion at arbitrary time. Secondly, in accordance to permanency, failures are classified into transitory, intermittent and permanent failures. Transitory failures occur in isolated manner and a re-execution of the service may omit this failure. Intermittent failures occur repeatedly in arbitrary manner. At this level, it is very indispensable to detect the main source of failures in order to avoid/prevent them. At last, permanent failures persist once they occur.

- **Means** Achieving dependable system may be handled using techniques so that the overall operation of the system will conform to the specification. The means to attain its measures and increase system dependability can be categorized into [2]:

  (a) *Fault-avoidance:* Means implemented to prevent the occurrence of faults.

  (b) *Fault-tolerance:* Means implemented to avoid system failures in spite the presence of faults, Generally, this is made by the replication of the components that are vital for the correct behavior of the system (redundancy techniques).

  (c) *Fault-removal:* Means implemented to reduce the number of faults, especially the severe faults.

  (d) *Fault-forecasting:* Means implemented to estimate the present, future creation and the likely consequences of faults. It is useful to record failures over the time to understand how their frequency is measured in order to avoid them.

  While, the means of fault-avoidance and fault-tolerance attempt to allow a system the ability to deliver its specification with dependability procurement; the means of fault-removal and fault-forecasting validate the dependability of a system [2].

2.3 Failures in vehicular communications

Since vehicular communications are maintained in harsh-nature areas, their systems are subject to several unexpected defects. Therefore, failures may occurring when VANET system real application must be analyzed in order to avoid/prevent them. In fact, different kinds of failures can demur VANET systems and cause to degrade their performances. These failures are mainly due to human errors, hardware and software failures and unreliable networks.

Human errors caused by driver and pedestrian account for 90% of road crashes and accidents [56]. At this stage, the replacement of the human judgment by computer-driven decision is likely suggested. Thus dependability is primary for providing correct services enabled by VANET systems. However, as vehicles in roads are managed by a driver, the behavior of this latter can damage the system, in both hardware and software levels so that nodes may suffer malfunctions. Incidentally, high potential nodes, such as vehicles having sensitive roles as data centers, relay nodes and cluster-heads are not excepted from suffering failures in some of their hardware and software components. For instance, if a relay node or a cluster-head fails forwarding emergency information, hazardous situation in the network will occur. In addition, nodes used as data centers may exhibit some failures conducting to offline services and applications. Furthermore, sensor nodes deployed in roads may also provide incorrect reading when their battery level reaches a certain threshold [57]. Therefore, imminent sensors' battery failures may be the main reason of many outlier reading in the network. As a way of illustration, infotainment applications will not perform properly if the underlying sensors are providing incorrect velocity and density readings.

Generally, hardware failures lead to software failures [57] and may cause permanent faults. Also, faults may propagate from node's level (vehicle, sensor and RSU) to network's level. For instance, vehicles' mobility and road traffic conditions cause to create/break-down communication links between the connected nodes leading to unreliable network. Actually, several factors can affect the network performances; namely, physical obstructions, nodes mobility, power depletion and network density. The physical obstructions made by the continuous atmospheric changes and solid structures (building, trees, tunnel ...etc) may prevent wireless signals to reach the desired coverage area. This is mainly duo to the effects of multipath and signal fading which reduce the real communication scope. Increasing power signal may be a solution to this problem; yet, this may increase signal's interferences occurred within communications in progress, especially in highly dense networks. In addition, sensor nodes will exhibit power depletion.

However, as it is mentioned before, there are principally four means to, at least, achieve system operation conform to its specification in spite the presence of fault. I other words, to achieve dependable systems. To this end, it is very indispensable to localize the main source of the occurred failure (detecting failure) in order to achieve the appropriate failure recover process (active fault-tolerance) or to avoid them in future operations (proactive fault-tolerance).

Fault detection may be carried out by:

- **Node self diagnosis:** By node self diagnosis, a node itself is able to detect faults in its components. For instance, if a node receives the same event advertisement from two different nodes with approximately the same timestamps, it can detect whether the frequency its local clock is slow or rapid.
- **Neighbors coordination diagnosis:** In neighbors coordination diagnosis, a group of nodes can detect a failure/attack in some services due to incorrect values. This allow to identify suspicious nodes whose reading have large difference against their neighbors [58]. As a way of illustration, nodes having large clock drift can be easily identified as bad or defected nodes against their neighbors.
- **Distributed detection diagnosis:** In distributed detection, nodes are supposed to have the same global information about the network before making a decision, as faulty nodes may forward inconsistent information.

In the following, we present clock synchronization as a main contribution to increase systems' dependability.

2.4 Timing notions and the practical use of clock synchronization for dependable systems

2.4.1 Clock terminology and basic notions

A clock is an electronic oscillator that keeps time, generally, regulated by crystal quartz. At each time period T, the crystal oscillator generates a signal with a specific frequency, f. This signal will be deciphered to the clock time T. Therefore, the clock is defined as a function from the real time t to the clock time $T : C(t) = T$ [59]. The frequency of the oscillator is not immutable and can change over the time. We note that the time of the ideal clock is always equal to the real time t.

Table 2.1 summarizes the most important clock's parameters:

Table 2.1: Clock's notations.

Parameter	Notation	Definition
Clock time	$C_a(t)$ or T_a^t	It is the time reported by the clock of a node a at the real time t.
Frequency	f_a^t	The frequency of a clock a at the real time t, is the rate at which the clock progresses. The frequency is given as the first derivative of the clock value with respect of the real time t
Frequency offset/ skew	S_a^t	The skew is defined as the difference in the frequency of the clock and that of the real time. Also, we define the skew of a clock C_a relative to a clock C_b at the real time t, by the difference between the frequencies of the two clocks $(f_a^t - f_b^t)$.
Clock offset	O_a^t	It is the difference between the clock time and the real time $(C(t) - t)$. The offset of a clock C_a relative to a clock C_b at time t, O_{ab}, is given by $C_a(t) - C_b(t)$ (see Figure 2.1).
Drift	D_a^t	The clock drift of node a at time t is defined by the fluctuation of frequency of the clock. It is given as the second derivative of the clock. The drift of clock C_a related to clock C_b at time t , D_{ab} , is given by $D_a^t - D_b^t$.

Figure 2.1 depicts the basic clock notation.

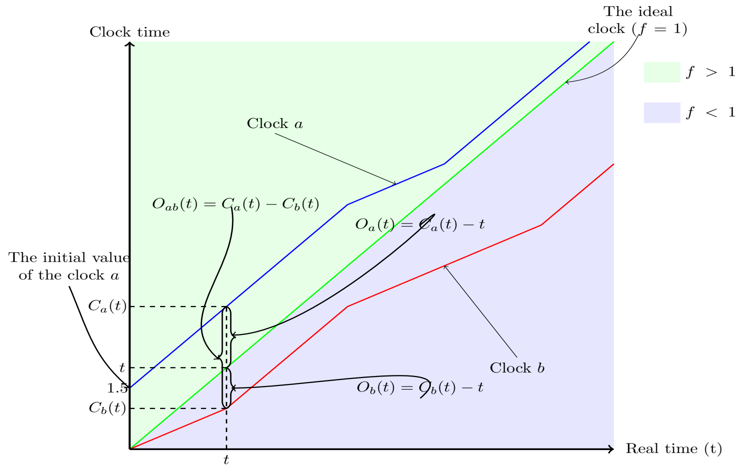


Figure 2.1: Notation.

Setting the clock time at a particular epoch is known as time synchronization. Adjusting the clock's frequency is referred to as frequency synchronization. Synchronizing both time and frequency is referred to as clock synchronization [60]. The following clock synchronization metrics hold:

- Accuracy: The clock synchronization accuracy refers to how much the maintained clock value is close to the standard time.
- Stability: The stability refers to how much a constant skew can be maintained.
- Average error: the average synchronization error is the average of time difference between every pair of clocks.
- Precision: The clock synchronization precision is the maximum time difference measured between any two clocks.
- Synchronization phase: The synchronization phase refers to the time period taken by a node to synchronize its clock.
- Synchronization interval: The synchronization interval refers to the validity period of synchronization. In other words, it is the time period between two consecutive synchronization phases.

As the clock is modeled as a function from real time to clock time, we distinguish three models of clock synchronization as below [61]:

- ***Offset estimation model:*** According to the offset estimation model, the clock value of node a is modeled as follows:

  $$T_a^t = t + O_a^t$$

  Where T_a^t is the clock value of node a at the real time t and O_a^t is the time difference between the clock value of node a and the real time.
  To synchronize node b, node a estimates the value of the local clock of node b as:

  $$T_b^a = T_a^t + O_b^a$$

  The offset estimation model can be only used to synchronize clocks evolving roughly with the same frequencies. However, since the frequency differs from a node to another, this model does not ensure long-term synchronization.

- ***Joint offset-skew estimation model:*** To improve the synchronization accuracy and precision, this model considers the frequency offset of the clock. The clock time of node a is calculated as follows:

$$T_a^t = S_a^t * t + O_a^t$$

The estimation of the clock value of node b related to the clock value of node a is given by:

$$T_b^a = S_b^a * T_a^t + O_b^a$$

- ***Offset-skew-drift estimation model:*** Several applications need synchronized clocks to be used for several hours. To this end, the offset-skew-drift estimation model can be used to ensure long-term synchronization. The estimation of the clock value of node a is given as below:

$$T_a^t = D_a^t * t^2 + S_a^t * t + O_a^t$$

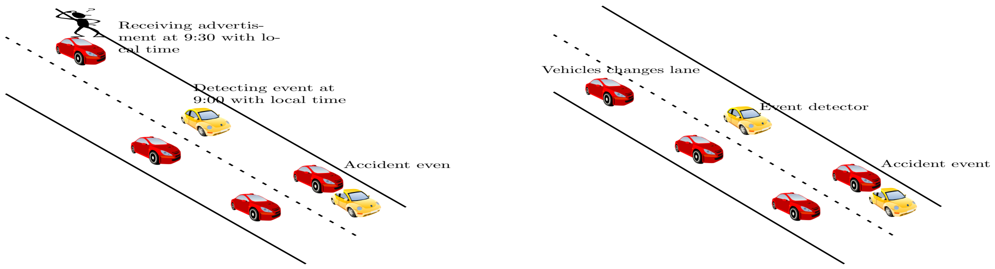
Node a achieves the synchronization with node b by the following estimation:

$$T_b^a = D_b^a * T_a^2 + S_b^a * T_a + O_b^a$$

2.4.2 Needs, challenges and requirements of clock synchronization in vehicular communication systems

As the clock synchronization is very important for any type of networks, it is even much more for VANETs. Not just for basic communications, but also for providing the ability to detect movement, location, proximity, security... etc. Even more, when VANETs become a sub-domain of the Internet of Vehicles, which in turn are part of Internet of Things, the clock synchronization takes more and more importance. That is, a good clock synchronization mechanism can significantly improve the quality of vehicle services. Let us cite:

i) Time synchronization is required for real time applications, such as safety applications. A detector of an accident event may broadcast a timestamped message to alert other vehicles in the road. If the timestamp received by another node is too late or too in advance regarding its local clock, that possibly causes that receiver ignoring the alert message without making any decision (see Figure 2.2).



(a) Non-synchronized clocks. (b) Synchronized clocks.

Figure 2.2: VANETs' safety application.

ii) The lack of a shared memory in distributed applications makes time synchronization of vital concern. For example, in vehicular cloud computing, suppose an application running at different vehicles, A, B and C, in shared mode. Each node needs to update and reuses the changes made by the other nodes. Supposing now that an error occurs in vehicle A at the instant t_A. To deploy the error, node A alerts the other nodes by an error message and requests to re-execute the application at this instant. Ultimately, the operation will fail if the clocks of nodes A, B and C are not synchronized. Depending on the application, this problem could be more severe in Big database manipulation, for example.

iii) When time scheduling protocols, such as Time Division Multiple Access (TDMA), are deployed to share a medium access, the clock synchronization between nodes sharing the same channel access is required (see Figure 2.3).

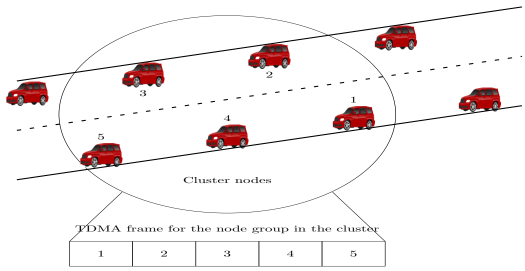


Figure 2.3: Time scheduling.

iv) For security issues, timestamping can be used to indicate whether the received packet is the original copy or a replay attack. A replay attack may be triggered by an adversary node, which is able to record, read, modify data and resend a packet in the network acting as the original sender (see Figure 2.4). To prevent this type of attacks, timestamping can be used and consequently synchronization should be achieved. A node A periodically broadcasts a packet timestamped with its local clock. Node B, which aims to communicate with node A, includes in its packets an estimation of the timestamps. Node A will only accept messages for which the timestamp is within a reasonable time tolerance. Also, many authentication protocols require a synchronization of clocks.

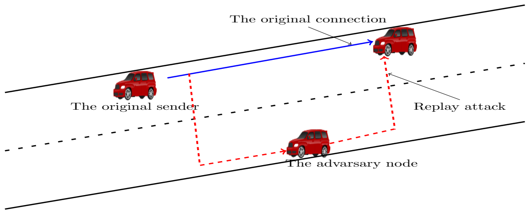


Figure 2.4: Replay attack.

v) For informational applications, and due to the limited bandwidth of the wireless communication channel, nodes can aggregate data from multiple other nodes in order to reduce the amount of data and disseminate them in large region. Data aggregation and transmission have the nature of timing sequence, thus clock synchronization is one of the most important issues for VANETs. Vehicular environments can be classified into three classes; urban, sub-urban and highways environments. The urban environments (Roads in the city) are characterized by the high density of vehicles. The synchronization process will generate more communication overhead. Therefore, collision and hidden node problems are pointed out as the major limitation in city roads. In addition, the high building and the intersections may affect traffic jams and fading. Even so, the retransmission to ensure reliable delivery cannot be delayed because of the constraints of clock synchronization. Thus, clock synchronization schemes would have to deal with packets loss in the protocol itself. In contrast, the highways environments are characterized by the high mobility of the moving vehicles on multi-lane roads. The network disconnection problem is more severe. Thus, vehicles would face high frequent disconnections when the transmission range employed cannot reach further destinations. However, to meet their desired objectives, clock synchronization protocols should improve the following requirements:

- Convergence time: the convergence time is the time taken to synchronize the network. Safety applications require the immediate relaying of information, without the introduction of any delay.

- Synchronization precision: clock synchronization process needs to ensure that the error at any real time t is less than the maximum bound on synchronization error.

- Scalability: The synchronization protocol has to cope with any increase in the traffic density like traffic jams. Also ensuring correct operation of safety applications in such scenarios.

- Synchronization rate: It is the percentage of vehicle nodes in a network that can correctly synchronize their clocks. The synchronization protocol has to ensure that all nodes can synchronize with one another. The synchronization of new arrival nodes should be managed.

- Communication overhead: It is the number of messages generated to achieve time synchronization process. Low message overhead is required for dense networks to avoid collisions and mitigate with the problem of packet loss.

- Synchronization in sparse area: In sparse area, the protocol should provide time synchronization as well as in dense networks. Otherwise, important time information can be lost when the network in the destination region is not fully connected.

- Robustness: Failures can independently appear at any time. The protocol should ensure the synchronization in transparent manner. Fault-tolerance should be deployed to cope with node failures in both clock and communication models.

- Time synchronization sustainability: The protocol should provide an overall time synchronization during the distributed system lifetime.

2.5 Conclusion

In the first part of this chapter, we presented the definition of dependability and its related basic notions, including threats, means and measurement of dependable systems.

While the second part is considered for pointing out on the role that clock synchronization plays to increase system dependability. Also, we presented the basic notion of timing and clock synchronization. In the next chapter, we review the taxonomy of clock synchronization and give an analytical study of the issue in the context of vehicular communications.

Chapter 3

State of the art

3.1 Introduction

The particular use of wireless communication technologies in the transportation domain has been increased with the active growth development of smart cars. This led to the emergence of Vehicular Ad hoc Networks (VANETs) as a hot topic of research. With or without the aid of a pre-established infrastructure, VANETs enable interaction between vehicles on roads aiming to the main purpose of improving road safety and online infotainment. Even more, the trend of the Internet of Things (IoTs) enables the interaction between vehicles and IoT components which is referred to as the Internet of Vehicles (IoVs). Consequently, VANETs afford intelligent control and services to vehicular environments.

In distributed systems, each node may obtain the time value from its local associated clock. This latter is implemented as a counter incremented with each tick of quartz oscillator. Also, the oscillator frequency is never fixed and depends on several variable factors. Therefore, the time value varies from one node to the other and may change over time (clock drift). However, the lack of global time view contradicts the nature of coordination and concurrency of distributed applications. Hence, the requirement to dispose a mechanism that provides appropriate clock synchronization throughout the distributed system lifetime.

Lamport [62] introduced the relationship “happened before” to determine events ordering. When two nodes communicate through message exchange, the sending event is said to ’happen before’ the receiving event. Therefore, logical time synchronization can be established. However, the logical ordering among the events does not meet the requirement of physical clock synchronization. This is a cardinal requirement for all dependable (reliable) distributed systems.

As it is known, time is a key parameter for system designing to ensure:

- Resource sharing: the main goal of distributed systems is to facilitate the access to shared resources, such as communication medium access.
- Concurrency: Multiple nodes working jointly on a specific application need an agreement about their common progress at real time. The timing sequence nature of these applications relies on an accurate clock synchronization mechanism.
- Fault-tolerance: fault-tolerance is an essential distributed system attribute that ensures reliability and maintainability. In distributed systems, nodes and communication links may fail independently at any time, so, fault recovering can be done by check-pointing or redundancy mechanisms. The lack of shared memory makes clock synchronization of a vital concern in any distributed applications needing to meet time requirements.

Additionally, several real time applications and services depend strongly on the existence of synchronized clocks. Among these applications and services in VANET systems we cite: traffic information, smart parking, movement detection, localization, etc.

Clock synchronization is well studied in wireless ad hoc networks (MANET). So far, many survey papers can be found in the current literature [63–68]. Nevertheless, the high velocity of vehicles that causes frequent changes in the network topology is a very challenging issue. Even more, the number of connected entities has incredibly increased and will increase more and more over time. These characteristics present extra challenges compared to the classical MANET. Furthermore, the performance of the synchronization process is closely related to the network characteristics, such as type, size, node distance, average speed, node density, etc.

This chapter weights clock synchronization in many aspects, in which, the practicability of the existing solutions in VANET environments mainly contributes to our work. First, we give a related work to clock synchronization issue. Then, we introduce the taxonomy of clock synchronization protocols with a detailed description of the existing protocols. The analysis provided for each protocol checks its practicability for VANET environments. Finally, a detailed comparison is provided considering relevant key parameters such as the density of nodes, the relative speed, the average time convergence, the communication overhead, the synchronization rate and the fault tolerance capability of the targeted protocol.

The rest of the chapter is organized as follows: In section 2, we describe clock synchronization protocols and their features. We also discuss the requirements and technical challenges of clock synchronization issue in VANETs. Section 3 surveys the main research works available in the specialized literature and gives an overview of the clock synchronization taxonomy. Furthermore, we present an analytical study in order to deal with their practicability in VANET environments. Finally, we conclude the chapter in section 4.

3.2 Clock synchronization message exchange mechanisms

Clock synchronization protocols determine how the nodes work to get synchronized. The main features that define a synchronization protocol can be described by the synchronization model and the message exchange mechanism employed during the synchronization process. The clock synchronization model affects the protocol accuracy. In turn, a good accuracy improves the synchronization interval. Thus, less communication overhead is generated. Also, convey time information between nodes is an essential process to achieve clock synchronization. The message exchange mechanism used affects strongly the communication overhead. The following section describes the synchronization message exchange mechanisms. We also discuss related work of clock synchronization protocols classification.

To achieve clock synchronization, nodes need to exchange timing information between them. The message exchange mechanism influences, positively or negatively, the performance of the synchronization protocol. A suitable mechanism should improve the synchronization accuracy with less communication overhead. Also, in mobile networks, the protocol should consider the mobility of nodes. The message exchange mechanism to be used has to be fast enough to avoid that nodes get out of one another transmission range.

Four message exchange mechanisms are captured; Unidirectional Broadcast (UB), Sender to Receiver (SR), Receiver to Receiver (RR), and Receiver Only (RO) based message exchange mechanisms.

- ***Unidirectional broadcast message exchange mechanism:*** In unidirectional broadcast model, the sender broadcasts a timestamped synchronization packet. All the receivers can update their clocks according to the time information sent by the sender node. For instance, let us cite the type example of the UB message exchange based synchronization mechanism. The time transmission protocol (TTP) [69] is used by a node which aims to synchronize another target node. To achieve time synchronization, the transmitter node A sends a sequence of synchronization packets to node B (see Figure 3.1). Relying on the timestamps on the synchronization packets, node B estimates the time of node A, T_{est} as follows:

$$T_{est} = \overline{R_n} - (\overline{R_n} - \overline{T_n}) + \overline{d}$$

Where $\overline{T_n} = (\sum_{i=1}^{n} T_i)/n$, $\overline{R_n} = (\sum_{i=1}^{n} R_i)/n$ and $\overline{d}$ is the estimated propagation delay.

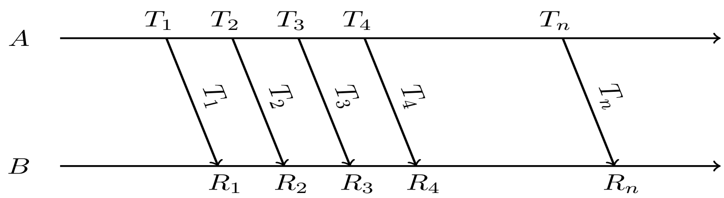


Figure 3.1: Unidirectional broadcast (UB) mechanism.

Obviously, TTP is more accurate when using the broadcasting channel to synchronize a set of target nodes. The advantage of simplicity in UB based synchronization comes at the price of accuracy. That is, because the message exchange mechanism does not consider the propagation delay. To deal with this limitation, several synchronization packets are used. However, more frequent overhead will be generated.

- ***Sender to receiver message exchange mechanism:*** Sender Receiver (SR) based synchronization protocols perform the two-way message exchange mechanism. A node A that aims to synchronize a node B, sends out a synchronization packet timestamped with its local clock. Node B marks the received packet using its local clock and sends it back to the sender. Based on the time information in the response message, this latter estimates the clock value of node B. Two clock estimation models can be distinguished; the round trip time synchronization and the set valued estimation method.

Figure 3.2 illustrates the four timestamps used by node A to estimate the clock value in node B.

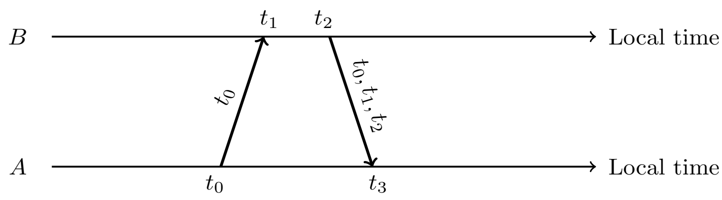


Figure 3.2: Round trip time synchronization method.

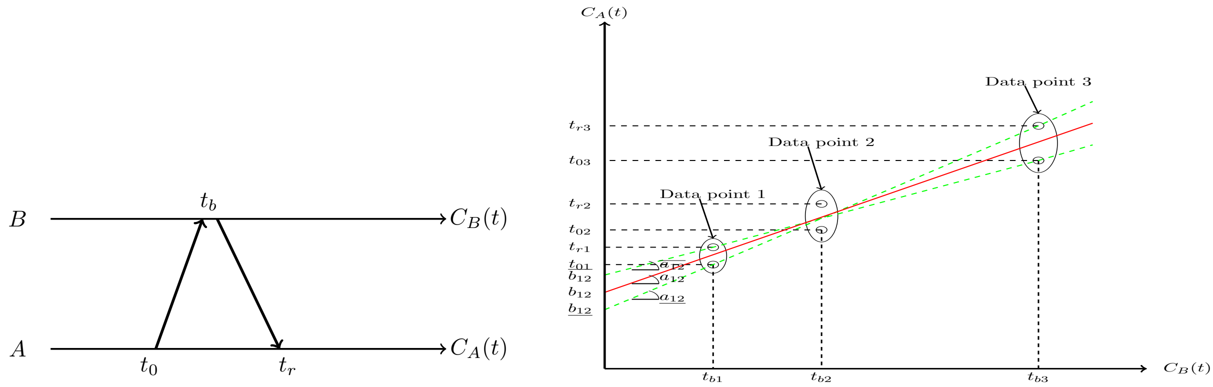
This model is also known as the offset delay estimation method. That is, node A estimates the offset and the propagation delay of node B as follows:

$$O_b^a = \frac{(t_1 - t_0) - (t_3 - t_2)}{2}$$

$$d = \frac{(t_1 - t_0) + (t_3 - t_2)}{2}$$

Where O_b^a is the the relative clock offset and d is the propagation delay.

Example of protocols using the round trip time synchronization are described in [70, 71]. Other protocols, such as [72], use the set valued estimation method to achieve time synchronization. The triple of timing information exchanged between A and B are used to plot the time graph; where the local time on the receiver node is on the X-axis and the local time on the sender node is on the Y-axis. Each of data triple can be plotted as an error bar, as shown in Figure 3.3. Also, the relative skew and offset are calculated from the slope and Y-intercept of any line that passes through all of the error bars [72].



(a) Time information collecting.

(b) Offset and skew estimation.

Figure 3.3: Set valued estimation method.

In contrast to UB-based synchronization, SR-based methods ensure better accuracy by using pair-wise synchronization. Indeed, accurate time estimation depends on a reliable delay estimation of the message exchanged between a sender and a receiver. The non-determinism in the latency estimation of the message delivery delay is the principal cause of SR-based synchronization errors. As shown in Figure 3.4, the uncertainty of the estimation is mainly due to send time, access time, propagation delay and receive time.

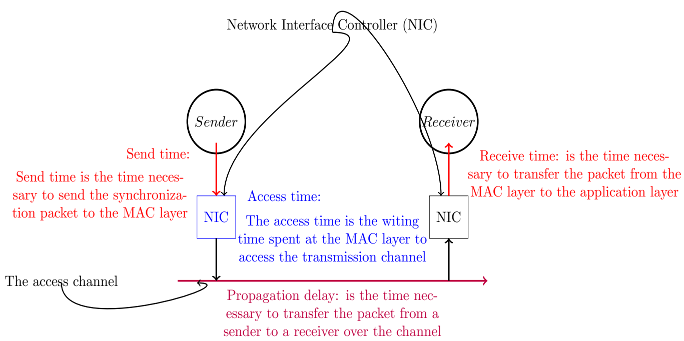


Figure 3.4: Delay uncertainties in sender to receiver communications.

To deal with the uncertainty of message transmission delay latency, researchers propose timestamping the synchronization packet at the MAC layer stack. SR mechanisms have got better performance. This is due to the elimination of the non-determinism in the sending, accessing and receiving times.

- ***Receiver to receiver message exchange mechanism:*** The main idea of receiver to receiver (RR) based synchronization protocols is to synchronize a set of receivers to one another. It depends on the existence of at least two nodes in the vicinity of the sender nodes.The sender initiates the synchronization process by broadcasting a synchronization packet. After receiving this packet, the receiver nodes will record the time of the arrival packet. Also they will exchange their recorded times to calculate the relative clock offset. Figure 3.5 hereafter, illustrates the message exchange under RR approach.

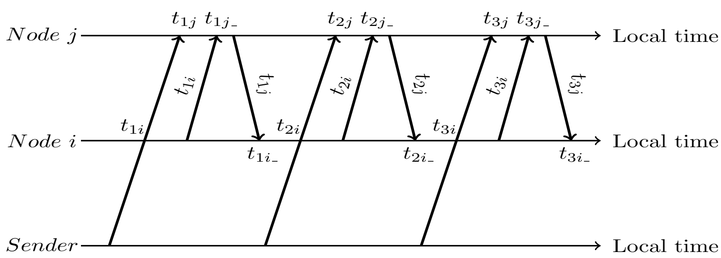


Figure 3.5: RR messages exchange.

The reference broadcast synchronization (RBS) protocol [73] is the first initiator of the RR mechanism. In the proposed protocol, several synchronization packets are sent out by the reference node. Each receiver i calculates its relative offset with respect to another receiver j. It is the average of clocks difference for each packet received by node i:

$$\forall i, j \in n : O_j^i = \frac{1}{m} * \sum_{k=1}^{m} t_{jk} - t_{ik}$$

Where m is the number of referenced broadcast messages, t_{kj} is the recorded time sent by the node j and t_{ki} is the time on node i when it receives the message from node j.

The RR mechanism exploits the broadcasting channel to eliminate the delay latency and introduces the concept of time critical path. Thus this mechanism provides a high degree in term of synchronization accuracy.

Figure 3.6 compares the critical path length in RR-based synchronization with SR-based synchronization. It is shown that the time-critical path length, and thus time synchronization are improved in RR-based synchronization protocols compared to SR-based synchronization protocols.

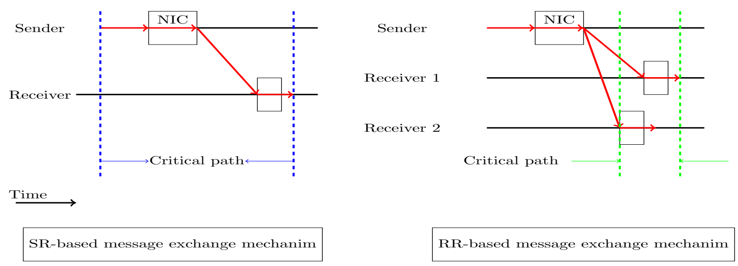
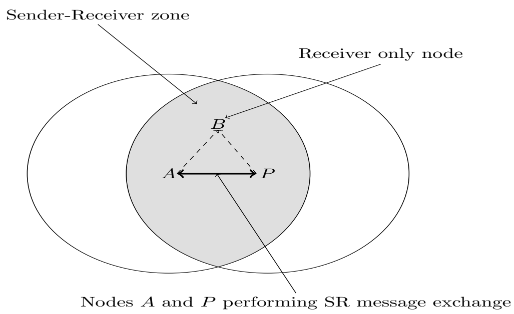


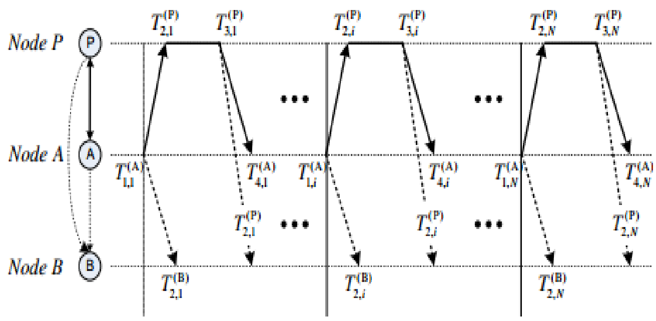
Figure 3.6: Critical path comparison between SR and RR approaches.

- ***Receiver only message exchange mechanism:*** In receiver only (RO) based synchronization protocols, a node can be synchronized by only listening to a SR message exchange between two other nodes. Noh et. al [74] introduce the RO mechanism for the first time and propose their PBS scheme. The aim is to achieve a network-wide synchronization with a significant reduced number of generated overhead.

The idea is to benefit from the overhearing a SR message exchange in progress. A third node in the scope of two other nodes performing SR synchronization can receive a sequence of synchronization packets.



(a) Receiver only synchronization model.



(b) Receiver only message exchange.

Figure 3.7: Receiver only synchronization.

As shown in Figure 3.7, nodes A and P perform SR-based synchronization. Node B is in the communication range of the two nodes. Therefore, node B can receive the synchronization packets and observe a set of time reading at its local clock. Then, the linear regression technique can be applied by node B to compensate the effects of the relative clock skew. The RO-based synchronization approach efficiently combines both SR and RO techniques to improve the communication overhead. Compared to RR approach, RO protocols achieve the same accuracy with minimal number of generated messages [74].

Table 3.1 compares the four mechanisms regarding to the number of generated messages and the average synchronization error. Where n is the number of nodes and x is the number of pair-wise synchronization packets.

Table 3.1: Number of message exchange on different clock synchronization mechanisms.

Synchronization approach	Number of messages	Average error	Basic protocol example
Simple uni-directional broadcast	$O(n)$	$1.48\mu s$ shown by FTSP protocol for one hop synchronization and $3\mu s$ for multi-hop synchronization [70].	TTP, FTSP
Sender to receiver (SR)	$O(n)$	$16.9\mu s$ shown by TPSN protocol [71]	TPSN, Tiny-sync
Receiver to receiver (RR)	$O(n^2)$	$29.1\mu s$ shown by RBS protocol [73].	RBS
Receiver only (RO)	$2 * x$	$29.1\mu s$ shown by PBS protocol [74]	PBS

3.3 Synchronization via GPS/GNSS component

Global Positioning Systems (GPS), as well as Global Navigation Satellite Systems (GNSS) provide navigation and precise positioning information. It can also provide accurate time to the end user. However, despite the requirement for GPS support, time synchronization via GPS component cannot be a good solution. First, GPS signals are extremely weak in the open space area related to VANET environments. That is, the most accurate read of time takes place in clear sky away from any obstructions, such as building, heavy trees, mountains and electronic obstacles that may prevent GPS signals. In addition, GPS signals in tunnels are too weak, because these latter cannot pass through solid structures. Indeed, an accurate estimation of time requires receiving signals from at least four satellites. The more the signals are received from more satellites, the more the accuracy is improved. However, GPS signals may raise the problem of transmission signal power, which interfere with the communications in progress within nodes. Furthermore, some geometry errors may occur when the satellites are on the same way. Signal corruptions thus obstruct using of robust GPS. All these challenges prevent GPS component to meet the requirement of time synchronization in VANETs for different purposes.

3.4 Related Work

There are several surveys on clock synchronization in distributed systems and ad hoc networks that have been published in the specialized literature [63–68]. An interesting recent survey paper is given by Amulya et. al in [64], where, the authors categorize the features of clock synchronization protocols into three main classes: structural, technical and global objective features. The structural features include the characteristics of the environment in which the system evolves. The main features in this category are:

i) *Internal vs. external synchronization:* In external clock synchronization, all the nodes synchronize their clocks to an external, from the network, clock source, such as Universal Time Coordinator (UTC) for example. Instead, in internal clock synchronization, the nodes make agreement to synchronize their clocks with respect to one or more clocks source in the network.

ii) *Stationary vs. mobile nodes:* The design of clock synchronization protocols for mobile networks is quite challenging as compared to that for stationary networks. This is due to the frequent change in the network topology in mobile networks which cause new challenges.

iii) *Single hop vs. multi-hop networks:* In case of single hop networks, a node communicates only with its one hop neighbors. But in case of multi-hop networks, nodes may not be able to directly communicate with one another because of the multi-hop topology (relay nodes). In that case, clock synchronization protocol need to support the multi-hop topology.

The technical features identify the techniques that have been used in the different phases of a clock synchronization protocol. At this level, the authors attempts to respond to the following basic questions:

i) *Master slave vs. peer to Peer:* the source from where the clock value is propagated may classify the clock synchronization protocols into master-slaves and peer to peer synchronization. In case of master-slave synchronization, a set of reference nodes are used to propagate the time information. The clocks of reference nodes may or not be synchronized to an external reference node. In such a case, the protocol provides both internal and external clock synchronization. Otherwise, peer to peer (P2P) synchronization comes out in the case when the nodes are synchronized with one another, without using any reference node.

ii) *Synchronous vs. asynchronous clock synchronization:* Depending on when the clock value is propagated, the clock synchronization protocols are classified into synchronous and asynchronous synchronization. In the contrary of synchronous clock synchronization protocols, where all the nodes perform the synchronization in fixed round periods, in asynchronous clock synchronization protocols, each node performs the synchronization independently of the other nodes.

iii) *A-priori vs. post-facto synchronization:* In a-priori clock synchronization scheme, the protocol runs the synchronization algorithm all the time (even so periodically). Instead, in post-facto synchronization, the synchronization is carried out whenever it is required to be done. Post-facto synchronization is usually suggested for small networks.

iv) *Probabilistic vs. deterministic synchronization:* probabilistic clocks synchronization schemes aim to guarantee that the synchronization error is always smaller than the prescribed bound with certain probability; whereas, in case of deterministic clock synchronization schemes, the protocols guarantee an absolute upper bound on the synchronization error. Therefore, compared to deterministic clock synchronization schemes, probabilistic clock synchronization schemes trade off synchronization accuracy with less communication overhead.

The global objective features emphasize the objectives that need to be achieved to meet the requirements of the system. The main features given in this section are:

i) *Fault tolerance:* It is required that the clock synchronization protocol is fault-tolerant to both message loss and node failure.

ii) *Scalability:* If the cost associated to the synchronization protocol, in terms of the number of generated messages and the amount of time taken to achieve the synchronization, increases as the number of nodes increases, so the clock synchronization protocol is not scalable.

iii) *Global vs. local synchronization:* In case of global clock synchronization, all the nodes are synchronized, whereas in case of local clock synchronization, the synchronization is restricted to a certain geographical area, specifically among neighboring nodes.

The simulation and empirical evaluation of clock synchronization protocols do not consider several aspects. Djenouri and Bagaa in [65] give a review on the implementation issues of clock synchronization methods. The authors discuss the state-of-the-art implementations up to 2015 and give a comparative analysis of the existing protocols. Among the aspects considered in the study are: the packet loss, mote limitations, clock drifting, fault tolerance, security and packet handling jitters.

Based on the type of the timing information exchanged between nodes, Rentel [66] classifies the clock synchronization protocols in three approaches.

- ***Burst position measurement-based protocols:*** this category of protocols rely on the periodic transmission of bursts or pulses. This requires a large bandwidth, and possibly a dedicated channel [73]. Each node measures the power associated with these pulses, and the delay of the detected pulses with respect of its local burst. The difference is used to correct its local clock. Most of the work belonging to this class [3, 7, 75] aim to mark the start of the packet data slots.

- ***Continuous correlation of timing signal based protocols:*** In continuous correlation of timing signal schemes, each node continuously transmits a sequence of signals and computes the phase offset according to the receiving sequences.

- ***Clock sampling based protocols:*** the information exchanged in this category of protocols is simply the clock value. Each node can read the time of its local clock and transmits it to neighboring nodes. When receiving this information, each node performs certain operations, depending on the clock model, to synchronize its local clock.

The synchronization process is closely related to the network type, size and other properties related to the environment. The aim of the present study is to give a survey of clock synchronization in VANET environments. There is a remarkable lack of solutions proposed to deal with critical situations and problems caused by non-synchronized clocks in such environments. Thus, this chapter weights the practicability of the existing time synchronization protocols (for classic MANETs and WSNs for example) to VANETs.

3.5 Taxonomy of clock synchronization in vehicular ad hoc networks

Most of the works in the literature are generally proposed to provide time synchronization in the context of MANETs and WSNs. The following sections analyze the existing taxonomy and its practicability to VANETs context.

The network time protocol (NTP [76]) has been widely used to synchronize clocks of nodes over the Internet network. The design of NTP relies on a hierarchical tree of time-servers. The primary server at the root derives its time directly from the Time Universal Coordinate (UTC) reference, typically using GPS. The next levels contain secondary servers and / or clients. The secondary servers act as backups to the primary server. Each NTP client performs the SR message exchange with its direct server. Therefore the offset delay estimation method is used to calculate the clock offsets. In a wireless network of 100 nodes, the time synchronization precision turns out to be of 2.8 ms. The precision varies with the number of nodes; more nodes will result in higher channel access delays which in turn degrades the precision [77]. By timestamping synchronization packets inside the interrupt service routine of the device driver level, Mahmood et al. can avoid channel access delays [78]. Time synchronization precision has been thus improved to be over 5.27 μ s. Furthermore, time stamping in the MAC layer may result in improving the synchronization precision and enhance the performance [79]. Therefore, the precision depends on where the time stamping is done, whether in the application layer or in the MAC layer. However, performing time synchronization using NTP protocol over large-scale and mobile networks is very challenging. This is due mainly to the huge communication overhead generated to synchronize each client with its server. In addition, the frequent change in the network topology leads to frequent link break up and makes the tree construction very hard.

The timing synchronization function (TSF) provides time synchronization in wireless networks. The TSF function is implemented under the IEEE 802.11 standard and works in both infrastructure and ad hoc modes. In case of infrastructure mode, the available access point (AP) provides mater-slave synchronization for all the stations within. Otherwise, the algorithm performs fully distributed synchronization. Each node maintains the transmission of a time stamped beacon every fixed period of time, eBeaconPeriod, within a contention window. Before transmitting its beacon, the node waits for a random-delay period. During this period, if a beacon arrives to the node, this latter will cancel the pending beacon transmission. The beacon will be transmitted only if no beacon arrives during the random period. Also, the channel should be sensed idle. Upon receipt the beacon, the receiver will set the timestamp only if it is later than the local timestamp. The probability of successful sending of one beacon is found under the assumption that perfect synchronization has been achieved. That is, the beacon contention window of every node starts at the same time for all the nodes in the network. Also, TSF-based synchronization does not include any propagation delay during the offset estimation. The maximum clock offset between nodes can reach over 400 μ s for large ad hoc networks [80]. On the other hand, the standard IEEE 802.15.4 e, that targets industrial applications, provides time synchronization with accuracy in the range of 1 ms and latency $\geq$ 100 ms [81]. However, the analysis proves that TSF-based synchronization is not scalable over large networks [82]. Extensive researches can be found in the literature in order to improve scalability and accuracy over IEEE 802.11 based synchronization [80, 83–85].

In [73] Elson et al. have proposed the reference broadcast synchronization (RBS) protocol for WSNs. The algorithm uses RR-based message exchange mechanism to eliminate the uncertainty of communication in the sender side and get better clock synchronization performance. The chosen reference node broadcasts a synchronization packet to initialize the process within its vicinity. Each receiver records the arrival packet with its local clock and then broadcast the recorded time to the group of nodes. Using this time information, each receiver calculates the offset relative to the reference node neighbors. Several synchronization packets can be sent by the reference node. So the average of the offsets would be calculated to improve the accuracy of the protocol. Also, to get better accuracy, the protocol suggests using the least square regression method to calculate the clock drift at each node [64]. However, RBS is not scalable over large networks. The overhead generated for synchronizing n nodes is in the order of $O(n^2)$. Further, the reference node, that initiates the synchronization process for its neighborhood, is not synchronized. Additionally, the performance of the protocol depends on the selected reference node. Also, in case of mobile networks, the algorithm must be re-executed to synchronize the new neighbor.

The TPSN protocol proposed by Ganeriwal et al. [71] performs time synchronization in WSN networks. Similarly to NTP, TPSN works in hierarchical manner, but does not require the use of an external time source, like GPS. The proposed protocol achieves clock synchronization in two phases, level discovery and time synchronization phases. In level discovery phase, the root assigned at level 0 broadcasts a level discovery packet to its neighbors. When receiving this packet, each node will update its level and, in turn, broadcast a level discovery packet to its neighbors. The construction of the hierarchical structure is attained by the level affection for each node in the network. Each node obtains its level by incrementing the level it received as the first level. In time synchronization phase, each node gets synchronized with it parent by executing the offset delay estimation method. Obviously, the time needed to synchronize the network converges with the size of the network itself. Also, in case of failures or arrival/departure of nodes, the maintainability of the tree structure takes more time. Lee et al. [86] propose a new scheme to synchronize new arrival nodes. The proposed algorithm acts without need to reestablish the hierarchical topology. The new arrival node broadcasts a timestamped request packet. Then, using the offset delay estimation method, it synchronizes with the node having the largest synchronized group. However, this solution can be practicable with low frequency of arrival nodes. This is due to the high number of message exchanges in order to deal with the high frequency of the arrivals, such as in VANETs.

Maróti et al. have designed the Flooding Time Synchronization Protocol: FTSP [70] to provide clock synchronization in large WSNs. Similarly to TPSN, the proposed protocol works in hierarchical manner. Also, it passes by two phases; the level discovery and the time synchronization phases. Instead of using the offset delay estimation method, FTSP relies on the simple UB based mechanism. Each node, starting with the root node, broadcasts a sequence of timestamped synchronization packets. The receiver records the arrival time of each synchronization packet from sender belonging to the sequence of its parent. Then, by using the linear regression technique, the receiver calculates the relative offset and the clock drift in order to correct its local clock. In addition, to face topology dynamicity and link changes, the root node is periodically re-elected. However, the peculiar nature of MANETs necessitates that two neighbors even if they do not appear in the same tree branch, may have to exchange their messages. Also, the distance between the node and the root affects strongly the synchronization error. The more the distance, the less is the accuracy. Although the algorithm is designed to provide time synchronization in large networks, the synchronization error grows quickly with the increasing of the network size [64].

The Time Diffusion Synchronization Protocol (TDP) [87] provides master/slaves time synchronization. When a precise time-server can be reached for a long period, it can act as a server to provide time for the nodes residing in the network field. This latter broadcasts a reference time to all the master nodes in the network. In turn, the master nodes use the received reference time to synchronize their neighbors. Otherwise, master nodes are randomly elected based on election/reelection procedure (ERP). The ERP includes a False ticker Isolation Algorithm (FIA) to remove the clocks that deviate from their neighbors. The master node starts the time diffusion procedure (TP) in order to diffuse the timing messages to its neighbors. Then, all the receiver nodes adjust their clocks by using the time adjustment (TAA) and a clock discipline (CDA) algorithms. Among its neighbor nodes, a few are elected as diffused leader nodes to diffuse the timing message. The diffused leader nodes are elected using the same ERP technique applied for electing the master nodes. At each hop, few diffused leader nodes are elected to diffuse the timing information to their neighbors. By this way, this timing message is diffused over the whole network. The protocol schedule consists of two parts, active and inactive parts. Each active part consists of a number of cycles of s time units. During which, TDP performs the synchronization process. At the beginning of each cycle, the master nodes and the diffused leader nodes are reelected. For the rest of the time, nodes will execute TP for a duration of d time units repeatedly. The number of cycle in the active operation, even the number of TP depends on the synchronization bound required by the application. The TDP clock synchronization protocol is scalable over large networks. Also, the protocol handles node mobility, and is fault-tolerant against node failures and message loss. The load balancing algorithm applied at the beginning of every cycle will prolong the network lifetime. Due to the usage of multiple master nodes, the hierarchy is of less height. Therefore, the tree structure may not influence the synchronization process. However, in the absence of external time source to synchronize the master nodes, the protocol takes more time [64].

Li and Rus [88] adopted four approaches for global clock synchronization in WSNs context. The proposed algorithms take into consideration the environment properties and the mobility challenges. These four algorithms are listed below:

- The all-node-based method,
- The cluster-based method,
- The fully localized diffusion-based method, and
- The fault-tolerant diffusion-based.

The all-node-based algorithm aims to synchronize the entire nodes when the network size is relatively small. The synchronization process is divided into two phases. Firstly, the initiator node records its starting time (t_s) and sends a synchronization packet along a ring of nodes. Upon receipt of the synchronization packet, each node in the ring records its local clock. Also, the number of hops that the packet has traveled is maintained. When the packet comes out to the initiator node, it will record its local time as the ending time (t_e). Secondly, along the same ring of nodes, the initiator node sends a correction packet. The correction packet contains the difference $t_e - t_s$ and the number of hops in the ring. Each node computes its clock adjustment based on this information. The cluster-based synchronization algorithm intends to synchronize nodes in a network that can be organized into clusters. The all-node-based algorithm is used to synchronize the cluster heads (inter-cluster synchronization). For the intra-cluster synchronization, the authors adapt the RBS protocol or any other synchronization scheme. All the nodes participate to the synchronization process roughly at the same time. This makes the implementation of the two previous methods too hard in large scale networks. Additionally, the hierarchy increases the flexibility at the price of the precision. Furthermore, failures that might encounter the initiator node make the solution non fault-tolerant. To palliate these drawbacks, the fully localized diffusion-based method has been proposed. In this method, nodes may participate to the synchronization process in synchronous or asynchronous manner. To achieve clock synchronization, neighboring nodes exchange their time values. Based on the received time information, each node updates its clock value. In the case of synchronous diffusion, every node has to wait for all the other nodes to finish their operation in the current round before starting the next round. Contrarily, in the case of asynchronous diffusion, nodes can perform their operation at any time in any order. The algorithm provides internal clock synchronization for the entire network in fairly simple manner. The protocol is fully distributed and fault-tolerant to node failures. The fault-tolerant diffusion based algorithm is an extension that considers clock synchronization in the presence of byzantine node failures. However, the complexity of the algorithm is quite high because each node sends out the computed average to all its neighbors. Additionally, in a single round of synchronization, every node may update its value multiple times. Hence, the protocol requires many synchronization rounds to reach a reasonable convergence [64]. The authors analyzed the algorithm behavior in case of fixed networks only. The analysis of dynamic networks with potentially non-equal constant node frequencies is stated as an open problem in their work.

In [74], Noh et al. devised a new message exchange based synchronization mechanism. This approach can be partially or fully applied for the implementation of clock synchronization protocols. Noh et al. have designed the pairwise broadcast synchronization (PBS) scheme using jointly the SR and RO message exchange mechanisms. So, nodes in SR zone of two nodes can be synchronized by only overhearing the exchanged traffic without any additional message exchange. This works on a strong assumption that nodes are in the transmission range of the two parts, sender and receiver nodes (Refer to the section 3.2 for more detail). To overcome such a limitation, Bagaa et al. [89] propose a synchronization protocol that uses piggybacked reference timestamps (SPiRT). SPiRT enables RO-based synchronization for nodes that are only in the range of one of the SR pairs. The proposed protocol works in cluster-based network architectures. The cluster-head is synchronized to a reference node through a SR-based mechanism. Since the cluster members can overhear the transmissions of the cluster-head, they easily can synchronize their clocks. However, the synchronization needs multiple rounds to be made. This takes much time for some nodes and may affect the delay for some applications. Indeed, in highly dynamic networks, such as VANETs, it is very hard to maintain the structure and the consistency of clusters. This is basically due to the high frequency of arrivals and departures of nodes to the clusters. The validity of the proposed solution depends closely on the resolution of this issue.

The Time Table Transfer protocol (TTT) [90] provides time synchronization in mobile WSNs. To synchronize the whole network, TTT exploits the mobility of nodes to move the timing data. The clock offsets relative to each node is kept in memory as an array record. The algorithm executes the offset delay estimation method to calculate the relative offset and the propagation delay between the interacting entities. Upon communication with a new node, the time table will be transferred to it. This enables the new node to synchronize all the nodes in the time table without having any message exchange between them. TTT provides incremental synchronization. Each node is synchronized with a set of nodes before fully synchronization can be reached. This is an advantage for different applications, because a node should not wait until the network is completely synchronized to perform its task. The simulation results show that the algorithm converges very slowly in the case of low velocity (less than 20 m/s). This is due to the use of node mobility to spread the timing data. Also, the number of messages generated will increase and may take additional time to synchronize the whole network.

The Time Table Diffusion (TTD) protocol proposed by Medani. K et al., provides time synchronization among the nodes in vehicular ad hoc network in distributed fashion [91]. The fact that each node exchanges its time information, like in RBS and JSL protocols, generates more communication overhead. The solution improves the ability of nodes to synchronize without any message exchanged between them. The main idea is to set a group of nodes to spread the time information over the networks. In general, TTD algorithm works in four principle phases. A set of nodes, called transporter nodes, broadcast a synchronization packet to initiate the synchronization process. The sender's receiver mechanism is used by the transporter node to estimate the clocks offset relative to its neighbors. Then, the transporter node broadcasts these offsets to its neighbors, allowing them to synchronize with each other without message exchange. Simulation results show that the proposed protocol has better performance in term of time convergence and message overhead with respect to the main constraining properties of VANETs, which are the high density and velocity of nodes. However, the protocol lack of the robustness over nodes failure.

The Gradient Time Synchronization Protocol (GTSP) [92] attempts to provide accurate synchronized clocks between neighboring nodes in WSNs. The protocol works locally in a fully distributed way to achieve global clock synchronization. Each node periodically broadcasts its global clock time and relative drift rate to its neighbors. After receiving the synchronization packets, each node computes the global clock offset and the drift rate by the use of the averaging technique, and updates its global clock. Comparing to the hierarchical synchronization schemes, the algorithm improves the synchronization accuracy between the neighboring nodes. In order to achieve better synchronization accuracy, the GTSP protocol runs repeatedly in very small periods which leads to generate more communication overhead in the network. Since the global clock is updated in every broadcast period, global time computed by each node is a discrete time-series value, which can introduce errors in data interpretation [93].

Ryad and Giovanni [91] propose the timeRemap scheme for stable and accurate time in vehicular networks. The authors point-out the following assumptions: On one hand, the operating system (OS) clocks can be stabilized over the long run but the OS timestamps are not accurate. On the other hand, the physical (PHY) timestamps are accurate but show a drift. For this, Ryad and Giovanni attempt to overcome these limitations by using the two systems jointly. The OS clock permanently adjusts its time to the UTC standard by the use of accurate GPS/PPS signals. The protocol provides a mechanism to translate timestamps in the PHY clock base into UTC base on demand [94]. When an internal event arrives, the occurrence time will be remapped to the UTC base. This time can be shared and exploited by other network interface controllers (NICs). Also, it can be used by an application or the OS to convert the PHY timestamps of the received packet into the UTC base. A mechanism that differentiates good observations for bad ones might be used. However, the GPS receiver needs to include a PPS output and an internal oscillator stable enough to keep the rate of the PPS signal constant in case of GPS signal lose [95].

The hybrid clock synchronization (HCS) protocol proposed by D. Sam et al. [96] aims to provide clock synchronization in hybrid VANET architectures. The nodes can be vehicles, sensors or RSUs. The proposed protocol performs time synchronization in a completely distributed manner, without any knowledge of the local or the global network topology. HCS utilizes the same idea of [4], which aims to synchronize the largest group of nodes. Each node in the network maintains a list of its neighbors and a list of nodes synchronized within. Any node in the network can randomly initiate the synchronization process. The initiator node broadcasts a synchronization packet to collect the time differences of its neighbors. The time difference is the deviation of node clock with respect to the standard time reference. Each neighbor responds to the synchronization packet according to its time slot in the sequence reply found out in the synchronization packet. In addition to the time difference, the reply message includes also the sync scale of the node. The sync scale is the number of nodes of the synchronized group. The node having the highest sync scale will be chosen by the initiator node to become the reference node. Then, the reference node sends out an adjustment message allowing the participants to the synchronization process to adjust their clocks. The proposed protocol outperforms RBS and other protocols in the state of the art in terms of time convergence and scalability over the density and the velocity of nodes. Nevertheless, simulation results show that the algorithm latency converges with the node velocity. Also, the aim to synchronize the largest group of nodes with one another, makes neighboring nodes which are not in the same group, not well synchronized. This increases the average synchronization error. Indeed, the protocol operates under a strong assumption that it uses an accurate time slot scheduling in order to avoid communication collisions and packet loss during the synchronization process.

The authors of [97] have focused their work to present clock synchronization in highly dynamic networks. The so-called diffusive clock synchronization scheme communicates only by means of pulses and computation evolves in rounds. At each round k, a node i broadcasts its pulse and waits until its local clock has increased by a constant R adjusted by a correcting offset. Assuming that a node i always receives its own pulses. The generation of round $k+1$ pulses depends on the received k pulses. The term R should be large enough to ensure that all the nodes receive all round k pulses of their incoming neighbors before they broadcast their round $k+1$ pulses. The evolution of the network dynamicity is modeled by a sequence of directed graphs. The fact that the round k pulse generated by node i is received by node j is modeled by a link from i to j in the k^{th} communication graph. Also, the fact that node i receives its own pulses is modeled with a self-loop at every node. The correcting term is a weighted average of the time differences of received round k pulses:

$$t_i(k+1) = t_i(k) + T_i(k+1) + corr_i(k+1)$$

Where $t_i(k)$ is the real time when node i broadcasts its own pulse,

$$(1-\varrho)R \leq T_i(k+1) \leq 1+\varrho)R$$

and, $\varrho \in [0,1]$ is the drift of the clock. Indeed, to perform time synchronization, each node has to transmit its pulse to other nodes. In addition to the large communication overhead, the transmission of pulses has the disadvantage that it necessitates large bandwidth.

The weighted consensus clock synchronization (WCCS) proposed by Aissaoua et al. [98] is a distributed clock synchronization algorithm for WSNs. The algorithm employs the elapsed time on arrival technique [99] to deal with the low convergence speed. In addition, WCCS utilizes the exponential smoothing technique in order to ensure an accurate clock skew synchronization and to enhance the robustness against network topology changes. The algorithm performs clock synchronization in a fully distributed manner, where each node communicates with its neighbors without any prior knowledge of the network topology. The objective is to synchronize all the compensated clocks in the network with respect to a virtual consensus clock. To this end, each node tries to estimate both skew error and offset compensations in rounds. The estimation takes place by employing a distributed iterative scheme based on a linear consensus algorithm. To evaluate the algorithm, the authors consider offset and skew error, scalability and convergence speed, while a lack of mobility parameter consideration is remarkable.

Table 3.2 gives a basic classification of the selected related work.

Table 3.2: Protocols' classification.

						Classification				
Protocol	Year	Reference	Timing information exchanged type	Message exchange mechanism	Clock model	Master-slave vs. P2P	Clock correction vs. Relative clock	Internal vs. External	Probabilistic vs. Deterministic	Global vs. Local
RBS	2002	[73]	Clock sampling	RR	Offset skew	P2P	Correction	Both	Deterministic	Local
TPSN	2003	[71]	Clock sampling	SR	Offset model	Master-slave	Correction	Internal	Deterministic	Global
FTSP	2004	[70]	Clock sampling	UB	Offset skew	Master-slave	Correction	Internal	Deterministic	Global
TDP	2005	[87]	Clock sampling	UB	Offset model	Master-slave	Correction	Internal	Deterministic	Global
Global clocks synchronization	2006	[88]	Clock sampling	UB	Offset model	Master-slave	Correction	Internal	Deterministic	Global
PBS	2008	[74]	Clock sampling	RO	Offset model	P2P	Correction	Internal	Deterministic	Local
TTT	2009	[90]	Clock sampling	SR	Offset model	P2P	Relative clock	Internal	Deterministic	Incremental
GTSP	2009	[92]	Clock sampling	UB	Offset skew	P2P	Correction	Internal	Deterministic	Global
timeRemap	2010	[91]	Clock sampling	——-	——-	P2P	Relative clock	External	Deterministic	Local
CTS	2010	[4]	Clock sampling	UB	Offset model	P2P	Correction	Both	Deterministic	Global
HCS	2014	[96]	clock sampling	UB	Offset model	P2P	Correction	Both	Deterministic	Global
Diffusive clock synchronization	2015	[97]	——-	——-	——-	P2P	Correction	Internal	Deterministic	Global
WCCS	2017	[98]	Clock sampling	UB	Joint offset-skew model	P2P	Correction	Internal	Deterministic	Local

3.6 Discussion

Clock synchronization is a crucial issue that ensures reliability and QoS of distributed systems. The main purpose is to deal with the lack of global time among all the entities in a distributed system. Wide range of protocols relies on the agreement on a common time notion. This is why the requirement of clock synchronization is a key aspect of successful application and services providence. Furthermore, with the application of distributed systems in different domains, clock synchronization has gained an increasing importance. However, the performance of clock synchronization method is closely related to the characteristics of the system. The accuracy and the energy efficiency are the primary concerns in wired and WSN networks. Nevertheless, other purposes should be taken into consideration when designing clock synchronization protocols in VANET system, including, but not limited to, scalability, relative speed handling, robustness and the synchronization of new arrival nodes.

Despite the extensive research that has been carried out on clock synchronization in distributed systems, there is a remarkable lack of study that focuses on the effects and requirements imposed by inter and intra vehicular communications. In fact, wired-based synchronization schemes cannot be applied in VANET systems. This is due mainly to the particularity of the wireless medium in vehicular environments, such as the effect of multi-path and fading. The TCF function under IEEE 802.11 standard is implemented to perform time synchronization in MANETs. The function suffers from scalability and accuracy problems, especially in multi-hop networks. In addition, this latter does not work well in high relative velocities conditions because of the high Doppler frequency shift [85].

Several packet based synchronization schemes for wireless networks can be found in the literature. A state of the art is discussed in this chapter. Our main purpose is to study the practicability of the state of the art discussed in the VANET domain. To this end, Table 3.3 compares these protocols considering the key parameters of VANET networks.

Table 3.3: Comparison

Protocol	Relative speed	Density	Communication overhead	Time convergence	New arrival nodes synchronization	Fault-tolerant
RBS	Low	Low (20 nodes)	High	High	No	No
TPSN	—	Low	High	High	No	No
FTSP	Low	Low (60 nodes)	Medium	High	Yes	No
TDP	Medium	Medium	Low	Medium	Yes	No
Global clocks synchronization	Low	Medium	High	High	Yes	No
PBS	Low	Medium	Medium	Medium	Yes	No
TTT	Average	Low	High in dense networks	High in sparse area	Yes	No
GTSP	—	Low	High	High	No	No
timeRemap	High	High	No communication overhead	—	Yes	No
CTS	Medium	High	Low	Low	Yes	No
Diffusive clock synchronization	—	High	—	—	—	No
HCS	High	High	Medium	Low	Yes	No
WCCS	—	Medium	High	—	—	No
TTD	High	High	Medium	Low	Yes	No

Briefly, a suitable clock synchronization protocol should improve low-cost message overhead, short time convergence, scalability, accuracy and long synchronization lifetime. The accuracy of the synchronization protocol is important, but accurate clock synchronization between neighboring nodes is also of great importance. Protocols using hierarchical architectures do not ensure accurate synchronization of all the neighboring nodes. As these protocols attempt to achieve global synchronization, the error is propagated down at different rates on different paths. In addition, in spanning tree structures, it is very difficult to maintain highly dynamic networks, like VANETs. Also, every hop will increase the synchronization error. On the other hand, synchronizing direct neighbors using timing message exchange generates more overhead and leads to transmission channel interfering. Furthermore, the clock synchronization mechanism should be able to maintain its functionality in case of faulty or adversary nodes. However, fault-tolerant and secure clock synchronization is still an open research topic.

3.7 Conclusion

Clock synchronization is one of the open topics that attract research community. This chapter provides an analytical study of clock synchronization issues in VANETs, which has emerged as an important research field in the last decade. Indeed, the characteristics of VANETs impose new challenges and requirements that should be considered. Since there is a notable lack of oriented solutions, a selective taxonomy of protocols in the current state of the art must be reviewed in order to evaluate their practicability and adoption. To this end, a comparative analysis with respect of the key parameters of VANETs is given in the present chapter. In the next chapter, we present our proposed solution for clock synchronization which is adopted to vehicular communications environments.

Contributions

Chapter 4

Offsets Table Robust Broadcasting for Clock Synchronization in VANETs

4.1 Introduction

Recent advances in technologies enable a variety of objects, such as laptop computers, smartphones, wearable sensors, vehicles and other smart devices, to interact with each other easily and effectively via wireless ad hoc networks. Wireless ad hoc network cut across many applications that improve and facilitate human lifestyle. Among these applications are the healthcare, industrial, social, transportation and communication industries. The integration of mobile ad hoc networks (MANETs) in transportation means brings a new paradigm called Intelligent Transportation Systems (ITS). The ITS advocates safety, efficiency, conviviality and performance when driving.

Vehicular Ad hoc NETworks (VANETs), a sub-class of MANETs, consider vehicles as mobile nodes. Nodes in VANETs are characterized by high-velocity and predictable movement in open space areas (roads). VANET applications can be divided into two major categories: safety and user applications. The safety applications increase road safety using traffic information systems to prevent accidents and road collisions. The user applications provide additional, interesting and useful on-road services that aim to increase passenger comfort via Internet connectivity, mobile applications, multimedia and peer-to-peer applications.

Recently, VANET design and modeling have drawn significant attention in large scale networks such as future Internet of Vehicles (IoV). The ideal solutions applied in wireless ad hoc networks field cannot be directly integrated in vehicular communications context. This is mainly due to the high-mobility of vehicles, which frequently changes the network topology. Also, roads infrastructure (intersection, traffic jams, and the presence of buildings beside the roads) imposes new constraints, like radio obstacles, the effects of multipath and fading, disconnection in sparse areas, and the bandwidth issue.

VANETs constitute an emergent and attractive research field; however, some obstacles are slowing down their development leading to their lack of maturity. Among these obstacles, we can mention some of issues, which remain without satisfactory solutions, like: security, connectivity, robustness, reliable communication, and time synchronization. Our study considers time synchronization and to point out the problem let us use an example:

When an urgent event is detected by a node N_i, it will be directly delivered to its neighbors N_j and N_k . Since the node clocks are non-synchronized, then node N_j will get an urgent event notification at 10:11 PM, while node N_k will receive it at 00:01 AM. The serious problem encountered by such a case is that the recipient nodes find difficulties in making consistent decisions about the arriving messages. They cannot distinguish whether the received real-time data is recent or out-of-date. This generates decision-making errors in the environment in case of deleting a recent real-time packet because it appeared to be outdated. For that reason, clock synchronization is one of the most significant issues that must be addressed, especially in distributed environments. To avoid such problems, node clocks must be accurately synchronized to correctly maintain the event causality property.

Time scheduling protocols are put in place to avoid communication collisions. If the distributed node clocks are non-synchronized, time scheduling protocols do not make sense since node clocks are not synchronized with one another.

Synchronization is a vital concern that must be taken into consideration when evaluating a performance system. Communication, coordination, security services all strongly depend on synchronized clocks of the different nodes in the network. A search of specialized literature revealed that the most recent research focused on addressing the following issues: how to rapidly disseminate emergency messages [100, 101], how to optimize data transfer from a source to a specific destination [102–107], and how to control channel medium access [108–114]. Also, wireless mobile services and applications have paid more attention last few years [115–119]. Unfortunately, there is a remarkable lack solutions for the clock synchronization problem in VANETs. For such a reason, in this chapter, we bring new solution in satisfactory solving such problems in unstable environments like VANETs.

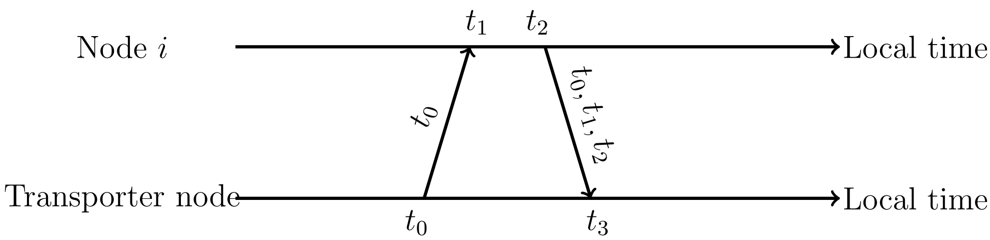
A suitable clock synchronization protocol improves VANET requirements. Among these requirements are: low-cost low-message overhead, short time convergence (time convergence is the time taken to synchronize the network), scalability, precision, and long synchronization lifetime. Robustness or fault tolerance are also essential criterion that must be taken into consideration when designing the protocol. Fault tolerance defines the ability of the network to maintain its functionality in case of failure.

The accuracy of the synchronization protocol is important, but accurate clock synchronization between neighboring nodes is also of great importance. Most of protocols in state-of-the-art systems are designed to optimize the global skew of nodes using hierarchical architectures (e.g., clustering, like in [3, 4], or spanning tree, like in [5]), which ensures that some neighboring nodes are not well synchronized because the error propagates down at different rates on different paths. Indeed, in spanning tree structures, it is very difficult to maintain highly dynamic networks, like VANETs, as every hop increases the synchronization error. On the other hand, in other protocols like RBS [6] and GTSP [7], synchronizing direct neighbors using timing message exchange generates more overhead and leads to transmission channel interfering.

In the current chapter, we present our contribution for clock synchronization in VANET systems. Here, we propose a new solution that attempts to eliminate the drawbacks of the related work. Among these drawbacks are: large communication overhead and time convergence, synchronization of neighbor nodes, and the lack in fault-tolerance. Firstly, we demonstrate our basic solution, called Time Table Diffusion (TTD) protocol. TTD provides time synchronization in distributed environment independently on the network topology, in which every node runs with the value of its local clock, but maintains the time information needed to synchronize other nodes for different purposes (e.g., point to point communications, medium access control and security). The proposed solution exploits the broadcasting channel to synchronize neighbor clocks without exchanging any messages between them, which reduces the communication overhead and and the number of delivered messages at each node. Secondly, TTD performance analysis leads us to enhance the solution to provide accurate and fault-tolerant time synchronization. The enhanced solution, referred to as Offsets Table Robust Broadcasting (OTRB), overcomes faults in both communication and clock models.

4.2 Time synchronization using time table diffusion protocol

Time table diffusion protocol (TTD) provides time synchronization in vehicular networks independently of the network topology [91]. The main idea is to set a group of nodes to spread the timing information. Based only on V2VC, TTD algorithm performs time synchronization in the following steps. Initially, any node can randomly initiate the synchronization process and broadcasts a synchronization packet to its two-hops neighbors. Other nodes, mark the packet receiving instance using their local clocks and send a reply message. The initiator, say transporter node, relies on the offset delay estimation method to calculate the offset relative to node i (see Figure 4.1), as well as, the offsets relatives to all its neighbors.



$$Offset = ((t_3 - t_2) - (t_1 - t_0))/2$$

The transporter node estimates the clock value of node i following the offset delay estimation method as follows:

$$C(\mathrm{i}) = C(Transporter) + Offset$$

Figure 4.1: Offset delay estimation mechanism.

The initiator node broadcasts the time information, in table form, to its neighbors, allowing them to synchronize each other without any message exchange. Each node runs repeatedly while the algorithm runs using a fixed amount of time to deal with the mobility of nodes and to prevent clock drift. Figure 4.2 hereafter encapsulates the synchronization process using the TTD protocol in a single round.

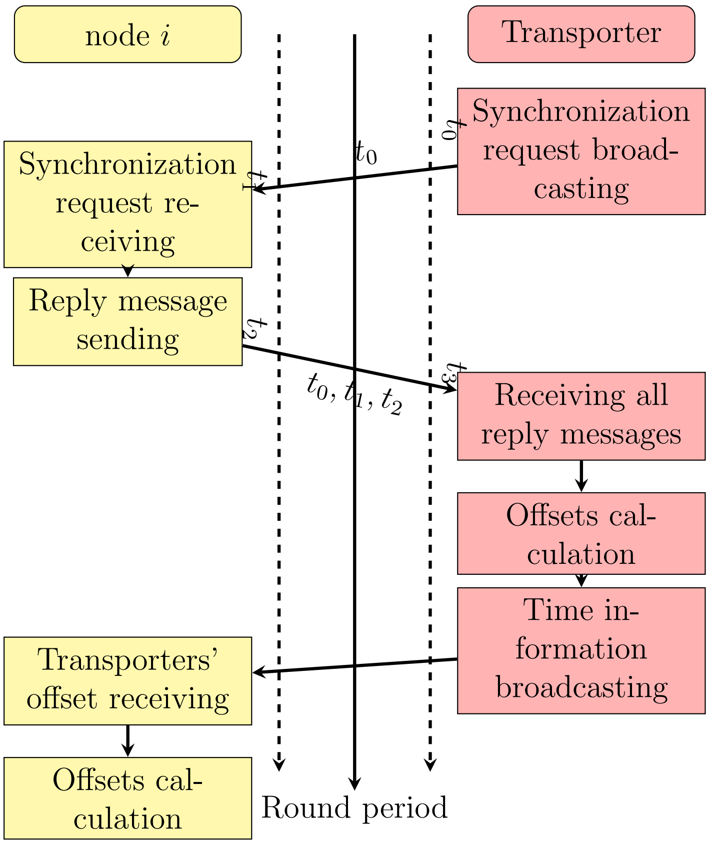


Figure 4.2: TTD protocol flowchart.

As shown in the example of Figure 4.3, nodes 0 and 5 are two transporter nodes. These transporter nodes broadcast a synchronization request to its neighbors 1, 2, 3, 4 and 1, 3, 4, 6 respectively. After they receive all the responses from its neighbors, each transporter nodes estimate and broadcast the offsets relative to its neighbors. Figure 4.3 shows the synchronized group of each node. Nodes 1, 3 and 4 synchronize with the two groups because they are in the scope of the two transporter nodes.

The most important requirements in VANET protocol design are the convergence time and the average overhead with respect to the parameter of vehicles density. The number of messages generated by the synchronization process is important because it may influence the network behavior positively (i.e., messages flow smoothly and easily over the network) or negatively (i.e., messages contribute to network congestion). The convergence time is the time it takes to synchronize the network. For analyzing the performances of the TTD proposal, we calculate the number of messages and the

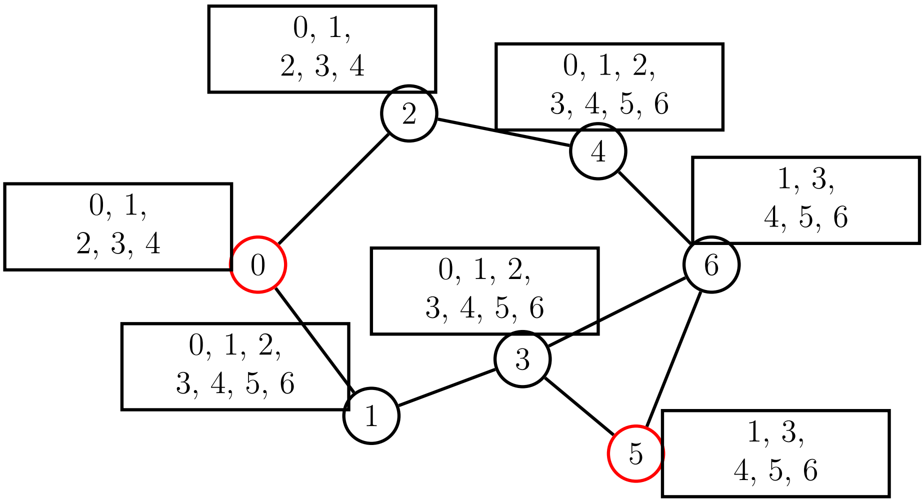


Figure 4.3: Illustrative example explaining TTD protocol.

convergence time as below:

- The number of messages ($nbMsg$) generated to execute the TTD algorithm is estimated as follows:

$$nbMsg = \sum_{i=0}^{n_t-1} (N_i + 2) \tag{4.1}$$

  Where, n_t is the number of transporter nodes in the current round, and N_i is the number of neighbors of the transporter node.

  Suppose there are n_t number of transporter nodes in the current round. The number of messages generated to initiate the synchronization process is equal to n_t, i.e., the number of transporter nodes. Neighbors responses phase generates several messages equal to the number of neighbors of all the transporter nodes, $\sum_{i=0}^{n_t-1} N_i$. In addition, each transporter node broadcasts its time table, which generates n_t time table messages. The total number of messages generated to accomplish the synchronization process is given by:

$$nbMsg = n_t + \sum_{i=0}^{n_t-1} N_i + n_t$$

  After simplification, we obtain:

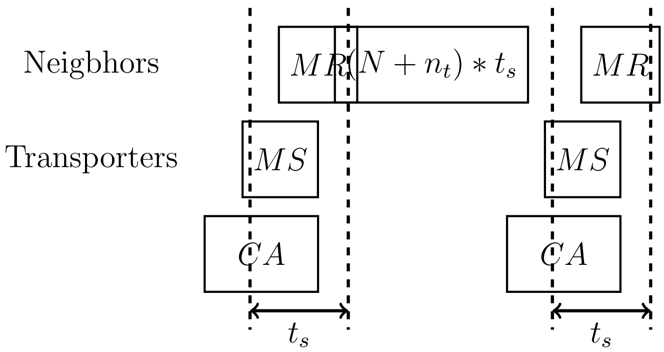
$$nbMsg = \sum_{i=0}^{n_t-1} (N_i + 2)$$

- To synchronize a network having N number of nodes, TTD algorithm must take place within the total time:

$$t_{total} = (N + n_t + 4) * t_s \tag{4.2}$$

where t_s is the maximum time required to deliver one message.

The timing sequence relative to each operation of the TTD protocol is given in figure 4.5.



Where:
- CA refers to thhe collision avoidance period. At max $CA = t_s$;
- MS = message sending;
- MR = message receiving;
- $MS/MR = t_s$.

Figure 4.4: The timing sequence of TTD protocol.

Figures 4.5 and 4.6 illustrate the change in the number of messages and the convergence time with respect to nodes number.

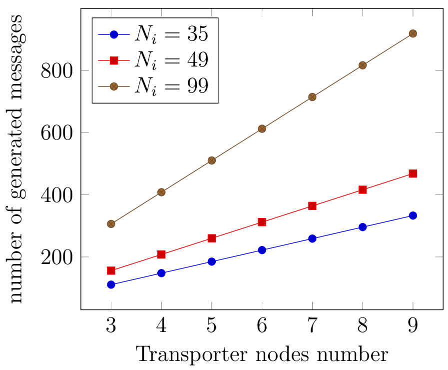


Figure 4.5: Messages complexity TTD solution.

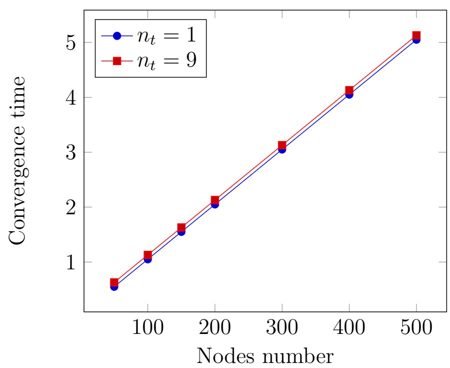


Figure 4.6: Convergence time in TTD solution.

Note that the number of generated messages is closely related to the convergence time. When the former increases, it contributes to the network's congestion. A congested network with collision produces message loss, which in turn slows down the convergence time. Another evaluation metric that should be considered is the synchronization rate. The synchronization rate refers to the rate of nodes that can become well synchronized. The frequency of departure of the vehicles lead to an increase in dropped communication links during the synchronization task. On the other hand, the new arrival of vehicles creates new communication links between the nodes. A reliable communication is quietly needed to synchronize moving vehicles, which in turn increases the synchronization rate. The proposed protocol would be able to face those challenges (new arrival node, link life time ...etc.). The TTD protocol, as described in [91], does not consider the reliability of message exchange, nor does it consider the synchronization of new arrival nodes, which affects its reported synchronization rate metric.

In the following sections, we describe our contribution of fault-tolerant TTD, called Offset Table Robust Broadcasting (OTRB). The OTRB solution considers the synchronization of new arrival nodes and studies the effect of dropped links. Additionally, the aim of the fault-tolerance in our setting is to prevent the fault-time synchronization information in the network, and to ensure the appropriate behavior of the solution in the presence of faulty nodes and communication channels.

4.3 Contribution

The enhanced OTRB protocol provides efficient time synchronization in VANET, thus optimizing message complexity. The system consists of vehicles communicating with each other over multiple wireless hops by using an embedded WiFi card. Inter-vehicle communications may or may not include a Road Side Unit (RSU) access point. Vehicles communicate with each other through Vehicle to Vehicle communication (V2VC), as well as they communicate with the available RSU through Vehicle to Infrastructure Communication (V2IC). The network relies on the ability of transmission time synchronization messaging to perform the time synchronization task. The messages are sent via a wireless channel based on the $IEEE$ 802.11 standard. We assume that the wireless channel is symmetric, e.g., if node A hears node B, then node B can also hear node A. For time synchronization, we adopt the following model:

- Initially, each node has a unique identity in the network, and maintains a list of neighbors, that are nodes within its transmission range.

- Every node has a notion of time that is based on the oscillation of crystal quartz. The clock time, $C(t)$, is the time reported by the clock at the real time t. For an ideal clock, the clock time $C(t)$ is equal to the real time t. The clock offset is defined as the difference between the clock time and the real time $(C(t) - t)$.

- The offset of the clock C_j relative to the clock C_i in real-time t is given by:

$$\Delta_{ij} = C_i(t) - C_j(t) \tag{4.3}$$

  Or:

$$\Delta_{ij} = \Delta_{ik} + \Delta_{kj} \tag{4.4}$$

- A node is considered well-behaved if the time shown by its local clock is bounded by the maximum drift (ρ), which is defined by the constructor.

- Each node moves freely per its local clock but maintains a time table containing the offsets related to neighboring nodes.

- A node is considered well-synchronized if the time difference between this node and each one of its neighbors does not exceed an acceptable average ϵ, which depends on the application.

- For mobility management reasons, and since the clock drifts naturally, achieving a consistent synchronization is important. Each synchronization period consists of Δt time units. The re-synchronization Δt value must be large enough to deal with all nodal mobility and clock drift issues.

Table 4.1: Clocks' notations table

Notation	Value
$C_i(t)$	The time value shown by the local clock of node i at the real instance t.
Δ_{ij}	The clock offset of node j relative to node i.
ρ	The maximum clocks' drift defined by the manufacturer.
ϵ	The average error tolerated by the application.
Δt	The round period, that defines the validity period of the calculated clocks' offsets.

The following sections describe the algorithm in detail.

4.3.1 OTRB algorithm

The enhanced OTRB algorithm performs time synchronization message-passing by using four principal phases. The initialization phase consists of the *transporter node selection*. The latter broadcasts a synchronization packet to request its two-hop neighbors to participate in the synchronization process and to collect the timing information. In the second phase, the *information retrieval phase*, the neighbors synchronize their local clocks and then reply to the transporter nodes within their scope. The transporter node uses the timing information received to *calculate an offset table and update the round period value* (Δt). Finally, the broadcasting of the calculated offsets table by the transporter allows other nodes to calculate their own offsets and synchronize with its neighbors, which is the *synchronization phase*.
Each normal node runs continuously via these four steps every Δt time unit (see figure 4.7).

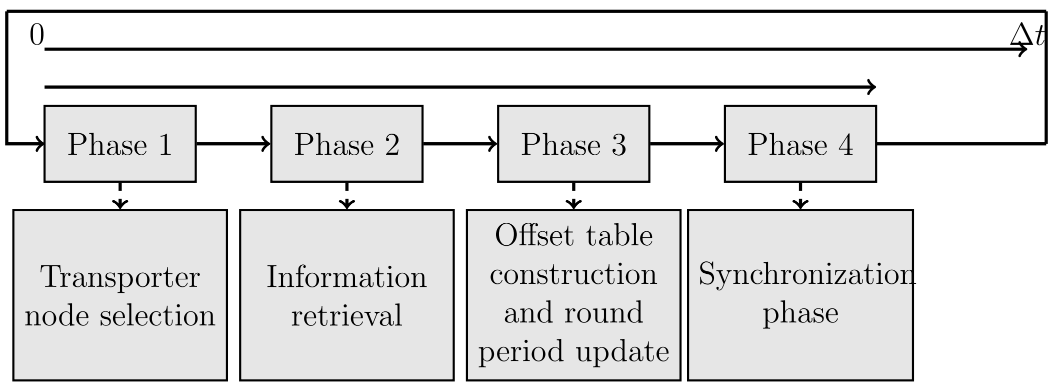


Figure 4.7: OTRB protocol phases.

(i) **Transporter node selection phase**

The transporter node selection phase strongly affects the algorithm's performance. Hence, the better the choice of transporter node, the better algorithm performs. Each transporter node and in-range neighboring nodes form a small network unit, or cluster. Selecting the wrong transporter node may cause an instability of the cluster. In the cluster, the transporter node is the cluster head (CH) and its neighbors are the cluster members (CMs). The vehicles that participate in the synchronization process under multiple transporters are called gateway members (GMs) (see figure 4.8).

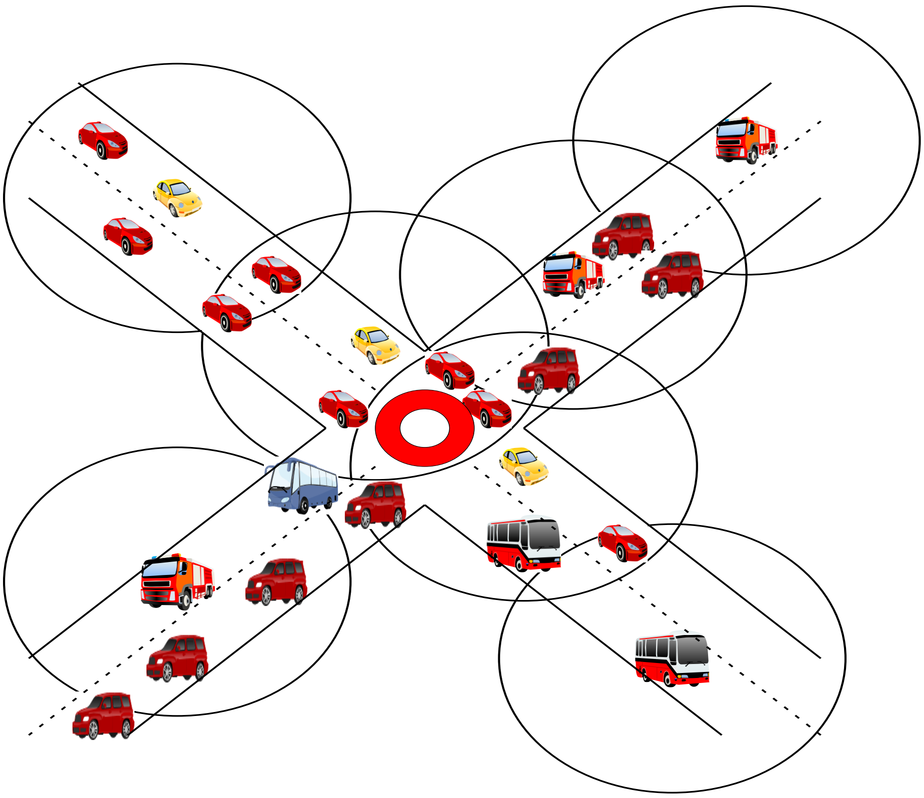


Figure 4.8: Network partitioning.

To improve the algorithm performance, the cluster formation should be as stable as possible; this means that the transporter node should meet some basic requirements, including some that set limits to the position and the velocity of the individual nodes. For this reason, the proposed method adopts a novel clustering algorithm [120–125]. The OTRB protocol can also adopt the transporter node selection with and without RSUs. If an RSU is available, the latter acts as a coordinator to set transporter nodes with respect to a set of conditions.

(ii) **Information retrieval phase**

In this phase, messages between the transporter node and its neighbor nodes are exchanged. The time they take to send and receive a reply message is used to estimates the offsets and update the round period value Δt.

Algorithm 1: Information retrieval algorithm

```
if isTransporter(i) then
    TID ← i; t_0 ← now_time; NT_ID ← nearest(i);
    broadcast ADV − T(TID, t_0, NT_ID)
    for each JOIN − RESPONSE received from node j do
        t_3j ← now_time
        t_1j ← t_1
        t_2j ← t_2
    end
else
    wait(ADV − T)
    for each ADV − T received do
        t_1 ← now_time
        insert(MyTransporterList, TID)
        wait(random_period)
        t_2 ← now_time
        send JOIN − RESPONSE(TID, i, t_0, t_1, t_2)
    end
end
```

Each transporter node broadcasts a synchronization packet, called an advertisement message (ADV-T), which is timestamped by its own local clock. The synchronization packet contains the transporter node identity (TID) and the timestamps t_0. The transporter node also sends the identity of its nearest node, NTID. The nearest node will be used to improve the reliability of the message exchange, it can also calculate and broadcast the offsets table in case of transporter node failure.

For any other node, say for example node i, when node i receives the ADV-T message from one transporter node, it marks its local clock at t_1 and then sends a reply message, called a JOIN- RESPONSE message. The reply message contains the identity of the node i, the identity of the transporter node TID and the timestamps t_0, t_1 and t_2, where t_2 indicates the JOIN-RESPONSE message sending instance. Each node must reply to all so that the ADV-T messages is received by the entire cluster.

(iii) **Offsets table construction and round period update**

After receiving the JOIN-RESPONSE message from each one of its neighbor nodes, the transporter records the time of its local clock at the instance t_{3i}. By using the timestamps founded in the reply messages, the transporter node relies on the offset delay estimation method (see Figure 4.1) to calculate the clock offsets relative to its neighbors.

$$\Delta_{(i\_TID)} = \frac{(t_1 - t_0) - (t_{3i} - t_2)}{2} \tag{4.5}$$

The obtained result is stored in a time table as:

$$time\_table_{TID}(i) = \Delta_{(i\_TID)} \tag{4.6}$$

Also, considering only non-bad clocks and the average error ϵ, the transporter node updates the round period Δt value as follows:

$$\Delta t = \frac{(n-r)\epsilon - 3m\epsilon}{2(n-r)\rho} \tag{4.7}$$

For any Δt, two well-behaved clocks may drift away at most by $\delta = 2\rho$, where ρ here is the maximum drift of all normal nodes defined by the manufacturer. In distributed systems, it is possible to have clocks synchronized with a maximum average ϵ, if no more than one-third of the clocks are bad or byzantine. Bad clocks exhibit large drift, they can be easily identified and their time values will be removed. Byzantine clocks are inconsistent nodes that transmit different timestamps to different nodes at the same time. The presence of one byzantine clock in a distributed system with n nodes can make any two arbitrary nodes to differ by $\gamma = \frac{3\epsilon}{n}$ sec. If the maximum drift between any two clocks is restricted to ϵ, the transporter node can calculate the round period Δt as follows:

Suppose there are n nodes within the transmission range of the transporter node in which m byzantine clocks, and r defective clocks, where $m + r \leq \frac{n}{3}$. So, the time difference increases from $\frac{3m\epsilon}{n-r}$ to ϵ is:

$$2\rho\Delta t = \epsilon - \frac{3m\epsilon}{n - r}$$

By simplification, the transporter node calculates the round period value as:

$$\Delta t = \frac{(n - r)\epsilon - 3m\epsilon}{2(n - r)\rho}$$

Algorithm 2: Offsets table construction and round period update algorithm

$Initialization : n \leftarrow 0;\ r \leftarrow 0;\ \rho \leftarrow 10^{-4};$

for *each join_msg received* **do**

  $\Delta_{TID,j} \leftarrow ((t_{3i} - t_{2i}) - (t_{1i} - t_{0i}))/2$

  if $|\Delta_{TID,i}| \geq \epsilon$ **then**

    $r \leftarrow r + 1$

    $insert(BadClockList, i)$

  else

    $n \leftarrow n + 1;\ time\_table_{TID}(i) \leftarrow \Delta_{TID,i}$

  end

end

$\Delta t \leftarrow \frac{(n*\epsilon - 3*r*\epsilon)}{(2*n*\rho)}$

$Broadcast\ TIME - TABLE(TID, time\_table_{TID}, BadClockList, NT\_ID, \Delta t)$

Table 4.2 hereafter lists the type, number and content of all the message exchanged during the synchronization process.

Table 4.2: Exchanged messages notation table.

Message notation	Designation	Content	Number
ADV-T	The advertisement message sent by the transporter node to initiate the synchronization process	• TID, the identity of the transporter node • NT_ID, the identity of the nearest node of the transporter • t_0, local time in the transporter node.	n_t, the number of transporter nodes in the current period
JOIN-RESPONSE	The reply message send by neighbors of the transporter node	Timestamps: t_0, t_1, t_2	$\sum_{i=0}^{n_t} N_i$
TIME-TABLE	The offsets table sent out by the transporter node	The offsets table and the round period Δt	n_t
ACK	The acknowledgment message sent out by NT_ID node to confirm the time table deliverance	TID, the identity of the transporter node, and the timestamps t_4	n_t

(iv) **Synchronization phase**

At this phase, the transporter node broadcasts the updated offsets table to all its neighbor nodes allowing them to calculate their own one.

Algorithm 3: Synchronization phase algorithm

```
for each TIME − TABLE received do
    if TID ∈ MyTransporterList & i ∉ BadClockList then
        time_table_i(TID) ← −time_table_TID(i)
        for eachnodej ∈ time_table do
            time_table_i(j) ← time_table_TID(j) + time_table_i(TID)
        end
        if i = NT_ID then
            t_4 ← now_t ime
            broadcastACK(TID, NT_ID, t_4)
        end
    end
    if TID ∉ MyTransporterList then
        send request_sync(i, gateway)
    end
end
if no TIME − TABLE received then
    wait(ACK_period)
    if ACK received then
        sendrequest_sync
    else
        insert(nonTrustList, TID)
    end
end
```

To get synchronized, when the node i receives the time table, it will proceed as follows:

- Initially, to synchronize with the transporter, node i estimates the offset relative to this latter as:

$$time\_table_i(TID) = -time\_table_{TID}(i) \tag{4.8}$$

We know that the offset of node i relative to the transporter node TID, Δ_{TID_i}, is given by:

$$\Delta_{TID\_i} = C_i(t) - C_{TID}(t)$$

Multiplying both sides of this equation by (-1), yields:

$$-\Delta_{TID\_i} = -C_i(t) + C_{TID}(t)$$

$$\Rightarrow -\Delta_{TID\_i} = C_{TID}(t) - C_i(t) = \Delta_{i\_TID}$$

$$\Rightarrow \Delta_{TID\_i} = -\Delta_{i\_TID}$$

$$\Rightarrow time\_table_i(TID) = -time\_table_{TID}(i) \qquad \square$$

- After estimating the clock offset relative to the transporter node, node i relies on equation (4) to calculate the clock offsets relative to each node j in the received table as follows:

$$time\_table_i(j) = time\_table_{TID}(j) + time\_table_i(TID) \tag{4.9}$$

Also, the nearest node of the transporter broadcasts an acknowledgment message (ACK) to confirm the time table delivery. The absence of messages or responses from the nearest node (NT) during a threshold time is used as an indicator of failed offsets. Figure 4.9 illustrates the message exchange during the synchronization process in one round.

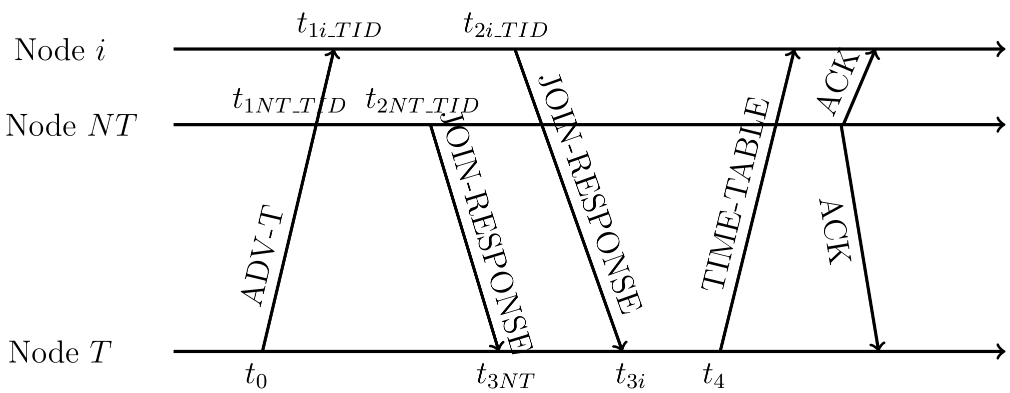


Figure 4.9: Message exchange during one round in the synchronization process.

There are two cases that should be considered:

1. If node i doesn't receive any time table from the transporter node, it waits for a threshold time to receive the ACK message:

(a) If no ACK message is delivered, because of a faulty transporter node, node i updates its list of non-trusted transporters. The non-trust list contains the identities of all transporter nodes joined but node i does not receive any time table sent by them. If the occurrence rate of the transporter node exceeds a certain threshold value, node i affirms that this transporter node is no more trust again (Figure 4.10 illustrates the comportment of node i when receiving the offsets table).

(b) If node i receives the acknowledgment message, which means node i has been moved out of transmission range of the transporter node. In that case, node i sends a request message to the nearest gateway member.

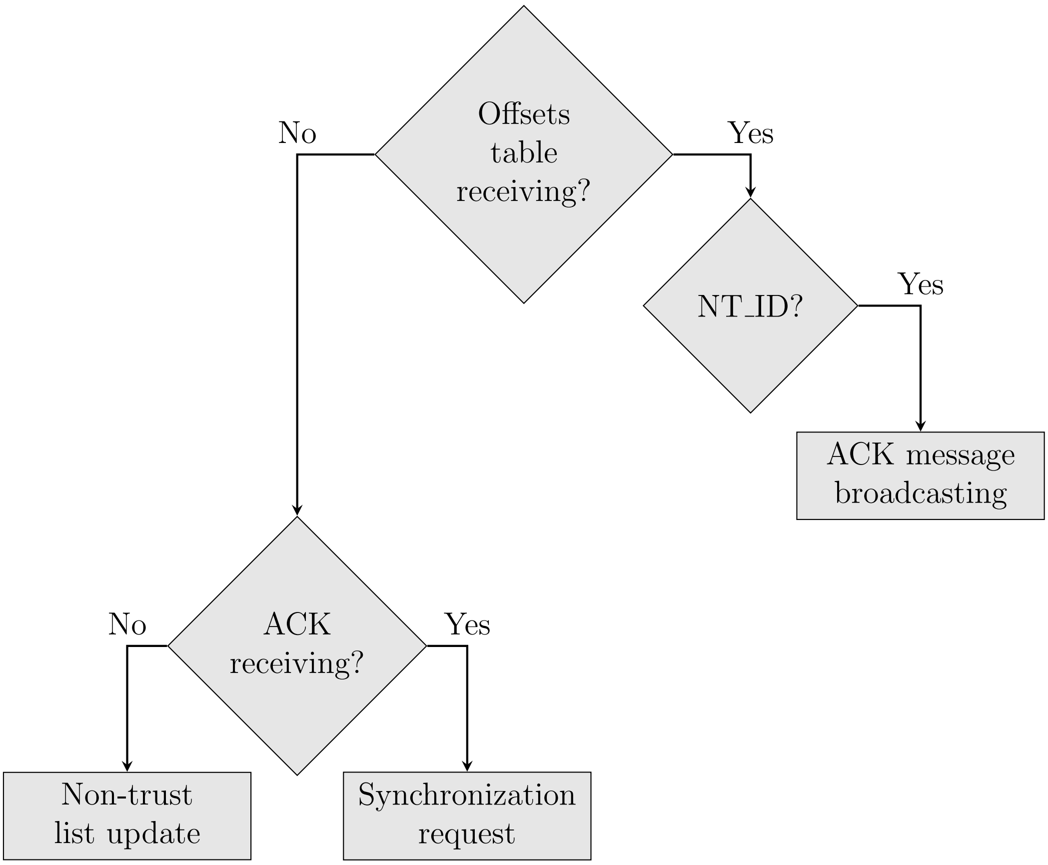


Figure 4.10: Receiving offsets table by node i.

2. Whenever new vehicle enters the group, it will be synchronized to the existing group by getting the offsets table from the gateway member or from its nearest neighbor. Contrary to the RBS protocol, it is not possible to synchronize new arrivals joining

the group; therefore, all vehicles should restart the synchronization process, which may incur some drawbacks.

4.3.2 Performances analysis

To evaluate protocol performance, the proposed protocol OTRB is implemented using the NS2 simulator. The physical channel characteristics are according to the specification of 802.11b, with channel bit rate of 11 Mbps. Many scenarios were used in which important parameters have been modified, such as the number of nodes and their speeds. Thus, at first, nodes were assigned to random locations and they moved according the mobility model, named the Intelligent Driver Model including Lane Change (IDM-LC) [86], made available by VANET MOBIlity SIMulator (VanetMobiSim). IDM-LC implements road intersection supervising strategy: making vehicle nodes slow down, stop or move in accordance with traffic lights. They are also capable of overtaking to change lane in multi-lane roads.

Regarding failure, we consider in this step, those occurring from the model used for communications. Thus, ten percent of the nodes are randomly considered as failing. In the simulation phase, clocks were assigned random values and generated in accordance to Gauss law [87] with $average = 0$ and $\delta = 10$ ppm. We fixed the maximum drift ρ at the value 10^{-4}.

In our setting, the delegation of the transporter node has been made in a distributed manner by considering the main parameters, which make the cluster as stable as possible. These parameters include the velocity of the transporter node, its position, and the number of its neighbors. The velocity of the node should be averaged according to the environment and the speed the other vehicles are moving. Also, we should ensure that the transporter node will not leave its neighboring during the synchronization process. Therefore, the elected transporter must be located at the center of its neighboring nodes. The number of neighbors the transporter node has defines the cluster density and so affects the communication overhead. The number of neighbors of the transporter node depends on the transmission range. It is more suitable that this parameter should be average to minimize the communication overhead and to facilitate a reliable and fast broadcasting of the offsets table.

Table 4.3: Simulation Parameters.

Topology (m^2)	10000*10000
Nodes number	100
Average speed (m/s)	15, 20, 25
Traffic light	6
Mobility model	Randomly according to IDM_LC with 2 obstacles every 100 m^2
Range data transmission (m)	250
Nodes' failure	10% of nodes
ρ parameter	10^{-4}
ϵ parameter	$10 * 10^{-3}$, $5 * 10^{-3}$, $2 * 10^{-3}$
Simulation time (s)	3600

4.3.2.1 Message complexity

The results from the simulation showed that the number of messages needed to achieve the time synchronization relies on two main factors: number of moving vehicle nodes (including the transporter), which is also related to the communication range. Indeed, the number of required messages increases with the number of active nodes (see figure 4.11). This results from message replies sent out by neighboring nodes, which constitute the most messages produced by a phase in the time synchronization algorithm.

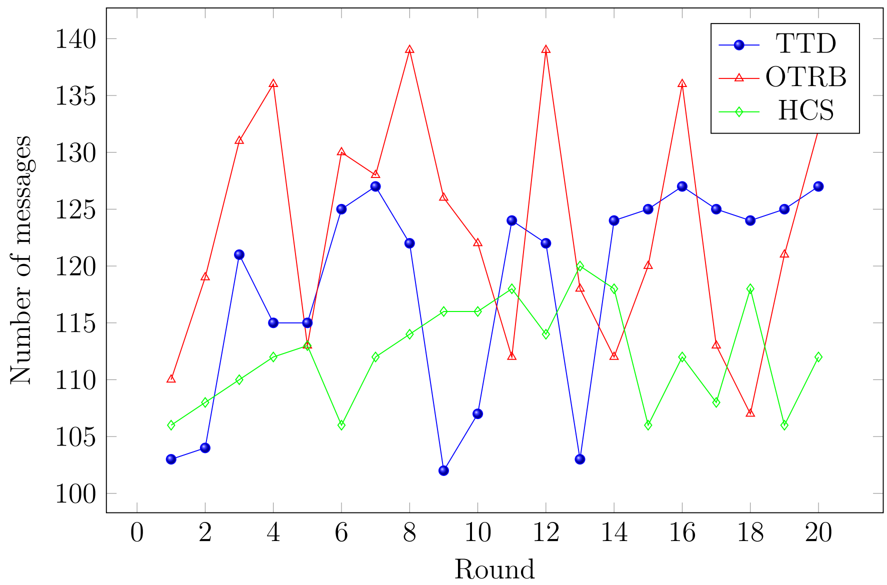


Figure 4.11: Message's number in different rounds.

4.3.2.2 Convergence time

The results provided by the simulation point out that when the number of nodes increases, it leads to the increase of the time convergence for OTRB. This situation is due to the number of involved nodes, which consequently generate a large number of neighbor replies (figure 4.12). The average convergence time is practically the same as in the TTD protocol (see [91]).

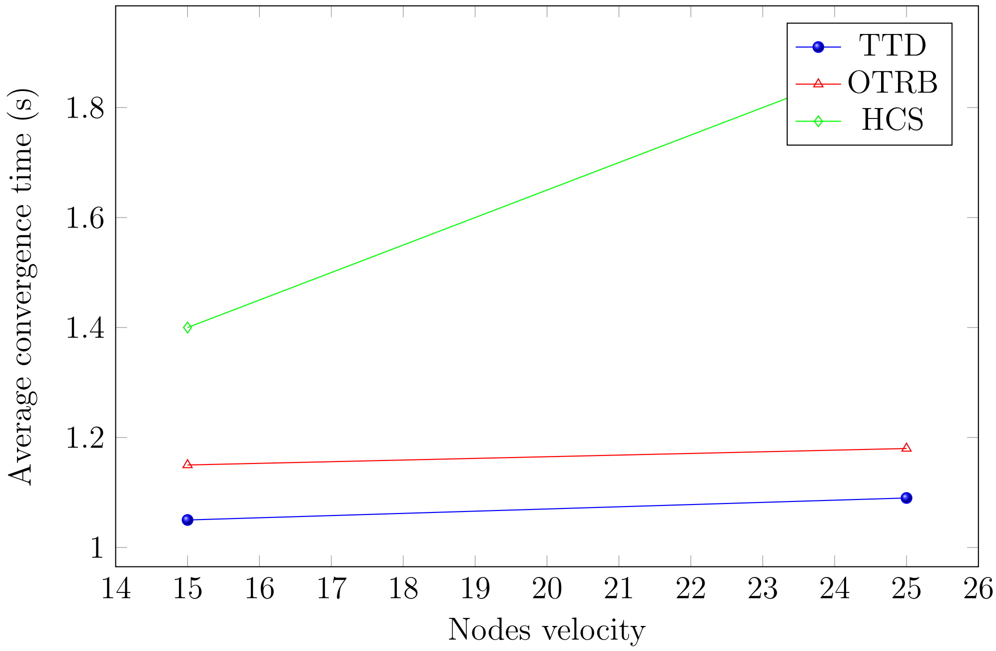


Figure 4.12: Average convergence time.

4.3.2.3 Synchronization rate

The synchronization rate is an important metric to evaluate the robustness of the proposed solution, which defines the ability of the protocol to perform time synchronization for the set of nodes in the network and to ensure reliable communication. As the protocol achieves the time synchronization per round, and the latter is calculated with respect to the clock drift, the synchronization rate is not incremental as our proposition hypothesized, but is calculated in each round separately. The synchronization rate in the TTD protocol can decrease in case of high node mobility and the instability of the cluster or the synchronized group (random transporter node). The simulation results show that the synchronization rate rises 100% in normal cases, but decreases to 90% with the injection of fault nodes, and to $(90 - x)\%$ if one of them is a transporter node with x neighbors. By contrast, in the OTRB protocol, the synchronization rate is improved by (i) the stabilization of the cluster construction; and (ii) the use of gateway members to synchronize the new arrival nodes.

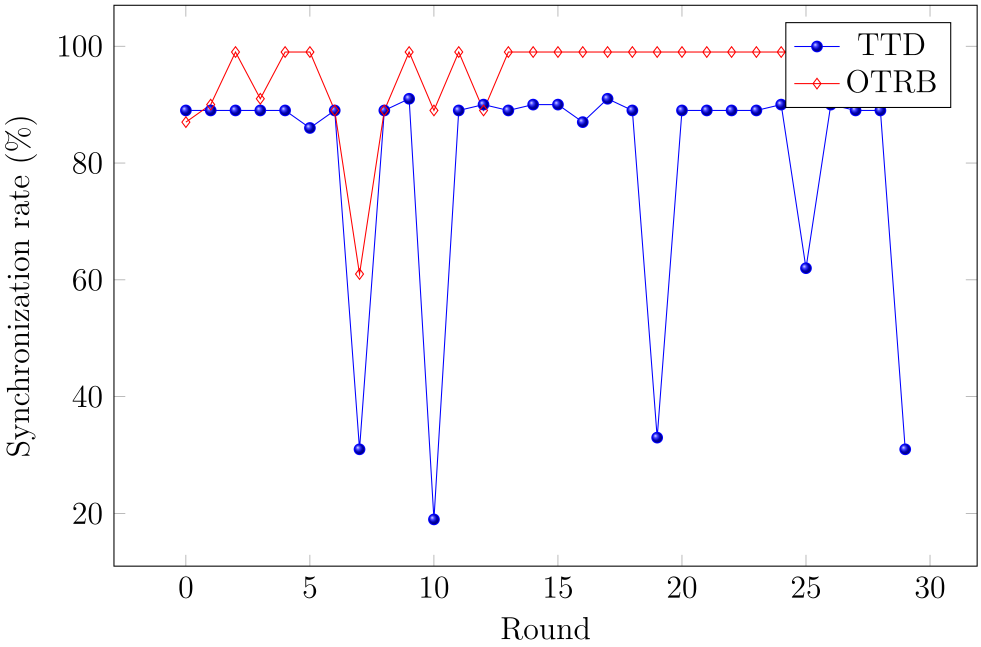


Figure 4.13: Synchronization rate.

4.3.2.4 The average synchronization error

The average synchronization error is given by averaging the time-difference between every pair of nodes. Figure 4.14 shows that the average synchronization error of the proposed protocol is bounded by the average error defined by the application.

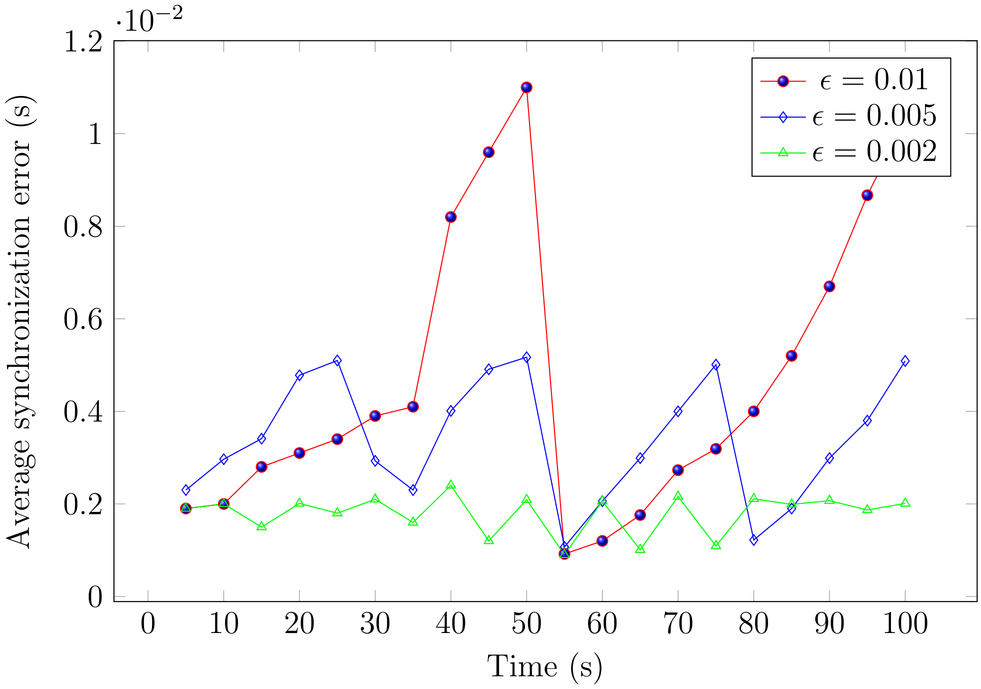


Figure 4.14: Average synchronization error.

4.4 Conclusion

Clock synchronization is a critical issue in VANETs for diverse crucial purposes including communication, data fusion, coordination, real-time safety applications, localization, and for many other applications. In order to face the technical challenges due to the constraints of unstable VANET environments, we have proposed in this chapter, a new time synchronization protocol referred to as Offsets Table Robust Broadcasting (OTRB). The proposed mechanism provides robust time synchronization with high accuracy. The robustness is owed, in one side, to the use of the broadcasting channel and, in the other side, to the acknowledgment message that ensures the offset table delivery by all nodes. The clock offsets are calculated using round-trip time mechanism, which offers a high accuracy with an average error of $16,9\mu s$. The performance evaluation of the proposal was carried out using an analytical model and different pertinent simulation scenarios. The results obtained by both approaches concur and demonstrate that OTRB outperforms similar protocols in terms of synchronization rate, convergence time and messages complexity, preventing so network congestion and communication capabilities degradation. The next chapter deals with network performance improvement based on clustering mechanism to reduce the number of communications caused by the high frequency of arrival/departure of vehicles while maintaining relevant clock synchronization adequately operating.

Chapter 5

Impact of Clustering Stability on the Improvement of OTRB Protocol

5.1 Introduction

The emergent Vehicular ad hoc networks (VANETs) tend to improve road safety and users infotainment during displacement. To this end, nodes in VANETs, as moving vehicles, exchange their information using wireless ad hoc communications. Due to the nature of the medium access, vehicles communicate directly with one another or with the available infrastructure using single hop communication. To reach a further destination, multi-hop communications can be established using relay nodes. However, the constraints created by the intrinsic characteristic of VANETs, such as the high density of nodes, make data routing a challenging task. To handle such a limitation, data should be spread among an optimized route to ensure reliable delivery with an optimized communication overhead. This can take place by splitting the network into small groups, clusters. For each cluster, a node is elected as a coordinator to communicate data with its members on the one hand and to spread the information to other clusters on the other hand. In addition, several issues depend on a strong assumption of reliable clustering mechanism.

The standard $IEEE802.11p$ has been proposed to deal with VANETs safety application. However, due to the high density of nodes, the standard suffers predictability, fairness, low throughput and high collision rate [126]. In addition, the high rate of transmitted data, such as safety packet dissemination, make network prone to face high congestion. So, collisions, especially when large transmission scope is used, lead to packet loss rate increases. To deal with these drawbacks and to improve reliability, many researchers suggest network partitioning into small clusters in order to allow better shared medium access control. Where clustering is used to limit the channel contention and provide fair channel access. Furthermore, clustering algorithms allow power control and packet delivery enhancement.

Time synchronization in VANETs is crucial. Several proposed algorithms, whatever derived from MANETs or not, rely on some sort of logical clustering view [127]. The aim is to provide local time synchronization among the neighboring nodes in order to reach global time synchronization. For instance, in HCS [96] and OTRB [127] protocols, nodes synchronize their clocks by the aid of a randomly selected node, which is considered as a reliable source of time. The time synchronization solution performances are related to the selected node. This elected node can be seen as the coordinator of its neighbors. Such protocols rely on the assumption that the algorithm provides reliable clustering with a minimized number of in / out moving vehicles.

In order to enhance its performances, the present paper carries a study case on the influence of clustering algorithms in time synchronization using OTRB protocol. OTRB algorithm requires the organized clusters to be stable for a period of time enough to achieve the time synchronization. The stability parameters, which refer to the average lifetime of the cluster, is related to the synchronization round period in OTRB. In turn, this latter is related to the synchronized group scale and the desired time precision [127]. Different clustering algorithms in the literature take only into account the distance between nodes and their directions as parameters to form the clusters. However, vehicles displacement is related to the road infrastructure. Even vehicles move roughly with same velocities starting from a nearby zone, they can be found in further ways after a while. Therefore, other parameters, like the velocity of nodes, should be considered in order to deal with the high rate of arrival/departure of nodes.

The main contribution of this chapter is to explore the impact of clustering on time synchronization process. To this end, we adopt an appropriate clustering algorithm to our previous solution OTRB. The best clustering algorithm should improve the synchronization rate and enhance the communication overhead.

The rest of the chapter is organized as follow: section II introduces the main clustering algorithm concepts and their requirement in VNAETs, while section III gives an overview of the proposed solution of clustering issue. Section IV is dedicated to the OTRB protocol. In section V, a comparison of the implemented clustering protocols under OTRB context is discussed in order to evaluate its performance. Finally, section VI concludes the chapter.

5.2 Basic clustering notions and requirements

The clustering is defined by the grouping of similar object items into clusters. In [128], a mathematical definition is given as follows: let $X \in R^{mxn}$ a set of items representing a set of m objects x_i in R^n. The goal of the clustering algorithm is to divide X into K groups C_k. Each one is called a cluster. Where all objects that belong to the same cluster are more alike than objects in different clusters. The result of the algorithm is an injective mapping $X \rightarrow C$ of object items X_i to clusters C_k.

The clustering can be exclusive, overlapping or fuzzy. In case of exclusive clustering, one object (node) can be found in one and just one cluster. When, in overlapping clustering, the node can belong to more than one cluster (≥ 1). Whereas, in fuzzy clustering, for each node is assigned a membership weight. Therefore, the node belongs to the cluster according to its associated weight. Fuzzy algorithms can be used to improve exclusive clustering by avoiding the arbitrary node assignment. In the case, the node will be assigned to the cluster within its membership weight is highest.

In VANETs, clustering is the act of grouping the vehicles into multiple clusters (groups of vehicles). In which, vehicles belonging to the same cluster have the same properties, such as velocity, position and direction.

Under each cluster structure, there is one node acting as a coordinator of the group. It is the cluster-head node (CH). The remaining nodes are classified to cluster-member (CM) and/or cluster-gateway (GW) nodes. In term of communication, a cluster-gateway node is able to communicate with all the clusters in which it belongs. Consequently, this latter represents the relay node of its clusters. In other words, it is the relationship between the neighboring clusters.

It is announced in [129] that "The clusters should reflect some mechanism at work in the domain from which instances or data points are drawn, a mechanism that causes some instances to bear a stronger resemblance to one another than they do to the remaining instances." Regarding to vehicular environments, several parameters have been considered, such as node density, position, speed and direction. These parameters define the requirement that should be employed by any clustering algorithm. From which, we define the following that affects clustering efficiency and network performance [41] [130] :

1. Cluster transmission overhead: The cluster transmission overhead refers as to the average number of packets exchanged to maintain the cluster structure. The more the transmission overhead is less, the more the clustering algorithm is desired.

2. Cluster convergence: Cluster Convergence refers as to the time needed for all the nodes in order to join a cluster. The clustering algorithm that takes large convergence time is less suitable.

3. Cluster connect time: this parameter refers as to the rate of connection of the vehicle to one cluster. The highest connect time rate is given, the more suitable clustering algorithm.

4. Cluster stability: the cluster stability refers as to the average lifetime of the cluster. The best clustering algorithm goes after high stability.

5.3 Related work

For the purpose of supporting Quality of Service (QoS) in VANETs networking, stable clustering of the moving vehicles is required. The design of clustering algorithms enables organized medium access control and simplified data communication task. Several works in MANETs literature focus on improving the generated communication overhead by minimizing the number of clusters.

In [131], Gerla et al propose to select the node having the highest number of neighbors as a CH in its vicinity. The proposed scheme minimizes the number of clusters. On the other hand, it deteriorates the network throughput due to the high rate of CMs within a cluster. This technique is not suitable for VANETs environment characterized by the high density of its nodes.

Other works, such as [8] selects the node with the lowest ID in its vicinity to be a CH of non-overlapping cluster. This protocol does not take nodes' mobility character into consideration. Then, CH selection process has to be frequently invoked especially in case of highly networks topology change, such as in VANETs. This will increase the cluster transmission overhead, which in turn reduces the network performances.

Su and Zhang [132] propose their algorithm which is a time and size based CH selection. The node that succeeds on sending its invite-to-join packet first and has more neighboring nodes is selected to be a CH. As the vehicles move in and out of the cluster frequently, there is high frequent of arrival and departure of nodes. Thus, maintaining the cluster stability becomes a hard task. Whereas, the mobility of vehicle in VANET systems is a major challenge to ensure clustering stability. Many researches in the current literature attempt to deal such a limitation.

In [133] the authors adopted the idea of [9]. In which, a node will announce itself as a CH in case it does not receive any Hello packet from its vicinity. Nevertheless, the node is allowed to be a CM in at least one cluster. The authors in the presented algorithm permit the merging of two adjacent clusters if their CHs come in direct communication range. The new CH is the one with the suitable weighed factor. This factor is calculated for each CH considering the mobility, connectivity and distance factors. However, CMs can be found out of the new elected CH, thus the possibility of frequent disconnection between the communicating nodes.

Kayis and Lichuan in [134] a speed-based classification in order to partition the vehicles into clusters. In which, seven groups of velocities are defined based on the min and the max speed boundaries. The "First Declaration Wins Rule" is adopted to select the set of the CHs. However, the nodes speed changes from node to node and can also change over the time. Therefore, nodes can frequently leave their group to join the more suitable. The algorithm may exhibit a frequent change in the network topology due to high cluster change rate. In addition, the CH can easily lose the connection with its member nodes.

A heuristic position-based clustering algorithm is proposed in [135]. The cluster formation is obtained by the partition of the network based on the geographical position and the priority associated with each vehicle. Each node calculates its priority value using its node ID, the current time and the eligibility value. The eligibility value increases with the travel time of the node and decreases when the node speed deviates largely from the average speed. The node with the highest priority in all its one-hop neighbors and one of its two-hop neighbors is elected to be a CH. Also, in the proposed technique, in order to optimize the cluster size, a maximum distance between the CH and its members is predefined. However, simulation results show that the formation of small cluster size increases the cluster reconfiguration rate. On the other hand, large communication range decreases the cluster transmission efficiency.

Wu et al [136] have proposed the Type-based Cluster-forming Algorithm (TCA) in order to reduce the update frequency of CHs in emergency ad hoc networks. Each node is assigned a stability factor, S, which is updated frequently. The node having a lower value of S has low mobility and reliable connectivity. The node with the lower stability factor is more likely to be elected as a CH. However, because of the assumption that the nodes have slow speeds, this algorithm generates more communication overhead for the cluster updating. This makes the algorithm not suitable to be implemented in VANETs environment.

A distributed CH selection is proposed in [137] to dynamically organize the network into clusters. The novel proposed algorithm is speed and distance based. The CH is selected according to its relative speed and the distance within its neighbors. Also, fuzzy logic inference system is used for predicting the future speed and position of all the CMs. As the maintenance phase is adaptable to the drivers' behavior, the proposed algorithm is highly stable. However, the distributed cluster overhead causes the number of messages received/transmitted to be decreased [41]. Also, the algorithm is implemented under medium vehicle density and low-speed conditions.

Rawashdeh and Mahmud [138] attempts to deal with the clustering issue with stability on highway VANET environments. The authors apart from the assumption that "the existence of VANETs nodes in the same geographic proximity does not mean that they exhibit the same mobility pattern", and try to improve the stability by making the network topology less dynamic. In addition to the velocity and the position, the proposed approach takes the location and the velocity difference between the nodes into consideration. Each node runs the algorithm in fully distributed manner. First, a separation of the vehicles into highly mobile and low mobile groups is made. The network partition should provide a minimized number of clusters and ensure that all the CMs are stable with respect to one another. Then, the set of nodes with the highest suitable value are elected as CH nodes. Each node calculates its suitable value, which is calculated based to the velocity and the distance with regard to its stable neighbors. The closer velocity to the average velocities of its stable neighbors the node has, the highest suitable value is given. Simulation results show that the proposed algorithm improves the average cluster lifetime, which increases the stability of network topology.

In [139] a new clustering algorithm in vehicular ad hoc networks is presented. The proposed scheme, called Hierarchical Clustering Algorithm (HCA), provides network two-hop clustering in fully distributed randomized fashion. Three roles can be defined to form the network hierarchy, in which a node can be a cluster head (CH), cluster relay (CR) or slave node. The CR nodes relay the transmitted data between the CHs and the slaves. The algorithm forms the clusters running four steps; cluster Relays Selection, ClusterHead Selection, Cluster Formation and Scheduling and Cluster Maintenance. Where it debates fast clusters construction in the first third steps and leave the mobility handling to the maintenance phase. However, the algorithm improves the clustering stability at the price of transmission efficiency and the average of clustering overhead. Also, the high rate of inter-cluster interferences causes frequent cluster changes and message loss due to message collisions. Another clustering algorithm for VANETs is proposed by Hassan abadi et al. in [140]. The authors have presented the Affinity PROpagation for VEhiclar networks (APROVE) protocol which attempts to produce high stability clustering taking into account the mobility parameter of nodes. In the proposed protocol, the CH selection is based on nodes' inter-distance. The closest node to its neighbors is selected. This scheme suffers long cluster convergence time in order to make all the nodes belong to a cluster.

In [10], the authors rely on neighbor vehicles' mobility to maintain VANETs stable clustering in rounds. In which, the node with the lowest neighbor vehicles mobility is elected to be the CH in its vicinity. Each node is assumed to have a unique identity, ID and maintain a neighbor vehicles table, which includes, for each neighbor, its ID and the neighbor vehicles mobility value. The neighbor vehicle table will be updated at each round and the neighbor vehicle mobility is calculated considering the number of vehicles entering/leaving the service convergence of the node.

In [141] a new oriented VANETs clustering scheme for urban city scenario is presented. In order to improve clustering stability, the proposed scheme considers the vehicles direction, position and the link lifetime estimation. Each node is able to estimate its speed and distance relative to its one-hop neighbors. The nearest vehicle into the central geographical position is likely chosen as a CH. The algorithm defines a safe distance threshold, D which D is less than the transmission range. The vehicles within this value are considered having more stable links with the CH. Therefore, the CMs are selected by the CH from its one-hop neighbors, in which the cluster size is defined by $L \leq 2D$.

An analytical study that evaluates the impact of the clustering mechanism on the QoS of VANET systems implementation must be carried out. By way of illustration, we analyze the influence of the clustering protocol performed while maintaining proper clock synchronization using OTRB protocol [127]. The analytical study is carried out using network simulator (NS2) to compare the impact of the rate of arrival/departure nodes on the performance of OTRB protocol. To this end, the following sections demonstrate the behavior of OTRB protocol to maintain synchronized clocks in VANET systems. Then the simulation results are presented.

5.4 OTRB protocol overview

For the purpose of time synchronization in distributed VANET environments, the OTRB protocol is proposed in [127] by Medani et al. The main idea is to set a number of nodes, named transporter nodes, to spread the time information over the entire network. For each transporter node, the time information broadcasted consists of the offsets related to all its neighbors. However, the transporter node is supposed to be a reliable source of time.

The OTRB protocol performs time synchronization relying on pair-wise time synchronization mechanism [63]. Initially, each transporter node broadcasts a synchronization packet to its neighbors. Upon receiving this synchronization packet, neighboring nodes respond by their time values as shown in Figure 4.2. Then, the transporter node calculates the clock offset relative to each neighbor i and broadcasts the estimated values, as a time table, allowing all of them to synchronize to one another.

The transporter node selection phase has an important impact on OTRB performance in terms of time synchronization rate and communication overhead. Thus, the transporter node should have a higher stability relative to its neighbors. That is, a node leaving its synchronized group before the synchronization will be achieved, generates more synchronization requests. In this case, time synchronization maintenance requires more communication overhead. The more the overhead is generated, the more the communication collisions. In turn, a high rate of collisions decreases the time synchronization rate. On the other hand, the transporter node selection phase should be as fast as possible in order to deal with the time synchronization requirements, delay latency for example.

It is stated in [127] that each transporter node with its synchronized group can be seen as a cluster. Where the transporter node represents the CH and its neighbors represent the CMs (see figure 4.8). Each node can join more than one transporter node. In the following, we refer to the transporter node as the CH, and to the synchronized group as the CMs.

To show the influence of the clustering algorithm on OTRB performance, in the next sections, we present the simulation and the comparison of three clustering algorithms (CBLR [9], LID [8] and NMCS [10]) working under OTRB protocol.

5.5 Simulation results

The proposed time synchronization OTRB protocol relies on a strong assumption of stable clustering mechanism. To evaluate its performance and show its better quality of functioning, we adopt three clustering protocols CBLR [9], LID [8] and NMCS [10]to work under the referred OTRB and the behavior of the protocol is observed for each one. The time synchronization rate and the average communication overhead needed to accomplish the synchronization process are measured with regard to the essential clustering stability parameters. In order to meet OTRB requirements, the implementation of these protocols is adapted to maintain overlapping clustering; where a node is permitted to be a CM in more than one cluster.

1. CBLR: CBLR is originally adopted on OTRB protocol upon its creation. A set of CHs are selected randomly, in which, a node will announce itself as a CH if this latter does not receive any synchronization packet from its vicinity. When receiving several synchronization packets, a node is allowed to be CMs in more than one cluster.

2. LID: LID protocol selects nodes with the lowest ID in their vicinity as CHs of non-overlapping clusters. As OTRB protocol requires overlapping clusters, LID is thus modified to allow nodes joining several clusters, if possible, in our contribution.

3. NMCS: In NMCS, the node having the lowest neighbor vehicles mobility is selected to be the CH in its vicinity. NMCS protocol allows overlapping clustering. For each node, a degree is defined as the sum of the number of nodes that have left/arrived the node's transmission range. The sum is then divided by the total number of neighbors to normalize the degree of node's mobility regarding its vicinity. The node having the lowest value for this metric is supposed to be located in a relatively stable environment, indicating that such node is a good candidate for the CH role.

The simulation of the three referred protocols holds under the parameters shown in Table 5.1. A scenario of 199 nodes with random positions is initially generated. The mobility model follows the Intelligent Driver Model with Lane Changing (IDM_LC) of VANET Mobility Simulator (VANETMobiSim), in which, vehicles regulate their velocity according to neighboring vehicle movements. This model also supports smart intersection and lane changing management.

Table 5.1: Simulation parameters

Topology ($m * m$)	Nodes number	Data transmission range (m)	Time of simulation (s)	Physical channel
3400*4900	199	300	200	IEEE 802.11b

In order to evaluate OTRB performances, we observe the changes in the synchronization rate and the communication overhead for each clustering protocol implemented under OTRB. On the other hand, to shed the light on the influence of the clustering protocol stability on OTRB performance, the average of arrival/departure and isolated nodes are captured in parallel.

As shown in Figure 5.1, NMCS protocol reduces the average of the nodes moving into new clusters during the synchronization process. That is, NMCS protocol favors nodes located in a stable environment relative to their vicinities to be CH nodes. In contrast, CBLR and LID protocols do not take into account nodes' mobility while performing the clustering operation. In addition, relying on the lowest ID and the random metrics to select the CH nodes in LID and CBLR respectively multiplies the risk of having so much number of CHs and a high number of isolated nodes (see Figure 5.2).

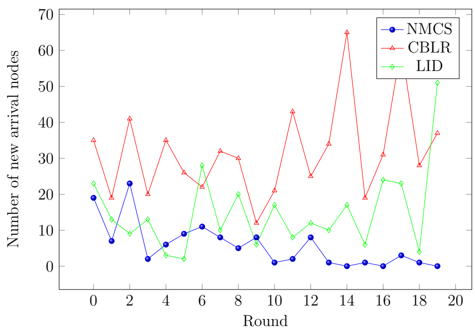


Figure 5.1: Average of new arrival nodes per round.

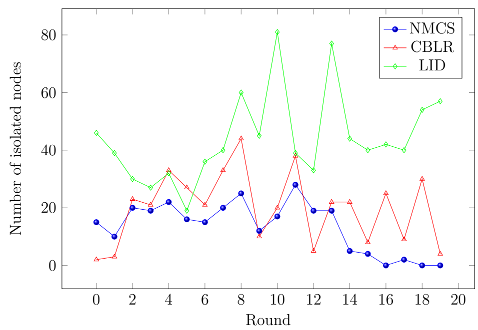


Figure 5.2: Number of isolated nodes per round.

Actually, the clustering stability metrics, arrival/departure, and isolated nodes affect the performance of OTRB protocol. Table 5.2 shows that the more stable is the clustering protocol, the more OTRB performances are improved.

Figure 5.3 exhibits that the synchronization rate of OTRB using NMCS outperforms OTRB using the two other protocols. This is due to the improvement in the average of arrival/departure nodes which reveals successful clock synchronization process in the overall of nodes. Therefore, the number of messages generated in order to accomplish the synchronization process is minimized. On the other hand, the more the number of isolated nodes, the more supplementary synchronization requests are generated. For that, using NMCS under OTRB protocol improves the communication overhead compared to CBLR and LID protocols under OTRB (see Figure 5.4).

Table 5.2: Comparison.

Implemented protocol	Arrival / departure nodes	Isolated nodes	Synchronization rate (s)	Communication overhead
CBLR under OTRB	High	Low	Medium	High
LID under OTRB	Medium	Medium	Medium	High
NMCS under OTRB	Low	Medium	Improved	Improved

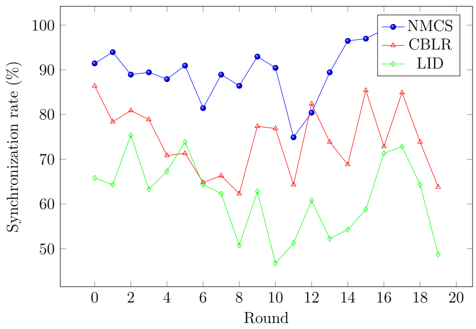


Figure 5.3: Synchronization rate comparison.

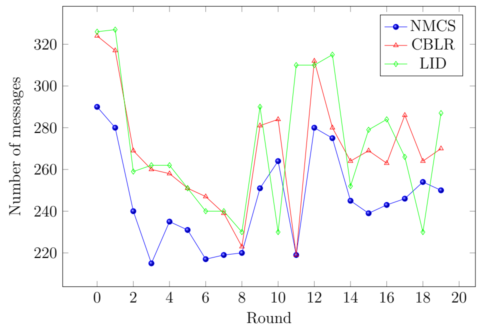


Figure 5.4: Communication overhead comparison.

5.6 Conclusion

Because time synchronization is an essential attribute of VANET networks, because VANETs are included in a larger area: the IoV which itself is part of the largest networks: the IoT, then the quality of service of the synchronization process must fulfill the expected objective in accordance with network importance. This chapter is dealing with the impact of the well know clustering nodes organization on time synchronization ORTB protocol in order to improve the latter capabilities. Several known clustering algorithms have been used by ORTB and the comparative analysis of simulation results reveals that better performance, in term of synchronization rate and average communication overhead, is offered by the high stability clustering algorithm. However, multi-metric algorithms can be used to further enhance the clustering stability and so the network performance. This can be the subject of future work.

Conclusion

Owed to the benefit gained from their applications in enhancing transportation systems, VANETs have emerged as a hot topic of research in the last decade. Especially, because they promise to provide a large amount of applications in order to improve the driving experience, facilitate secure traffic management, and enhance driving safety. Despite these advantages they may provide, VANETs come with their own set of constraining features that oblige the researchers to revise the solutions proposed in the current issues. In fact, the nature of the communication and the hostile environment, together make the deployment of dependable communications a real challenging task. In this thesis, we revised clock synchronization issue in the context of VANETs as a cornerstone to ensuring their dependability. Firstly, we analyzed the practical use of clock synchronization for improving reliability, fault-tolerant, maintainability, safety and security of vehicular communications. In other words, how clock synchronization supports VANET systems' dependability. To this end, we reviewed clock synchronization issue and its existing algorithms. As the characteristics of VANETs impose new challenges and requirements that should be considered and since there is a notable lack of oriented solutions, we gave a comparative analysis of a selective taxonomy of protocols in the current state of the art with respect to the key parameters of VANETs. The analytical study has shown that the evaluation of the protocols must be revised to adopt their practicability in the context of VANETs.

Secondly, we proposed a new timing synchronization protocol dealing with the technical challenges imposed by the constraints of unstable VANET environments. The proposal, referred to as TTD: Time Table Diffusion, then OTRB: Offsets Table Robust Broadcasting, aims to provide robust time synchronization with high accuracy. The proposal relies on the round-trip time mechanism to calculate the clock offsets which offers a high accuracy with an average error of $16,9\mu s$. In addition, the usage of the broadcasting channel speeds up the time information spreading over the whole networks and ensure reliable communications. The obtained results among validation demonstrate that OTRB protocol shows better performances in terms of synchronization rate, convergence time and messages complexity, preventing so network congestion and communication capabilities degradation.

5.7 Future work and perspectives

Although the proposed OTRB protocol is highly promising and may provide accurate time synchronization in highly distributed vehicular environments, there are other significant limits that should be carried out in order to improve its performance and so on improve network performance. We summarize them as follows:

- While the proposed protocol operates under strong assumption of reliable clustering algorithm, and as the results shown in paper C in the list of included publications demonstrate that the more the clustering algorithm is stable, the more the performances of OTRB are improved, our future work includes the implementation of a stable clustering algorithm dealing with the high mobility of nodes in vehicular environments. However, multi-metric algorithms can be used to further enhance the clustering stability and so the network performance.
- Improve the proposed solution by integrating/proposing a mechanism in order to identify byzantine clocks in the networks.
- Tackle security issue in the context of time synchronization in VANET systems.

Bibliography

Bibliography

[1] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33, 2004.

[2] Jean-Claude Laprie. Dependable computing and fault-tolerance. *Digest of Papers FTCS-15*, pages 2–11, 1985.

[3] Shizhun Wang, Anjum Pervez, and Maziar Nekovee. Converging time synchronization algorithm for highly dynamic vehicular ad hoc networks (vanets). In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pages 443–448. IEEE, 2010.

[4] Dahlia Sam, Cyrilraj Velanganni, and T Esther Evangelin. A vehicle control system using a time synchronized hybrid vanet to reduce road accidents caused by human error. *Vehicular Communications*, 6:17–28, 2016.

[5] Jun Liu, Zhaohui Wang, Jun-Hong Cui, Shengli Zhou, and Bo Yang. A joint time synchronization and localization design for mobile underwater sensor networks. *IEEE Transactions on Mobile Computing*, 15(3):530–543, 2016.

[6] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[7] Philipp Sommer and Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 37–48. IEEE, 2009.

[8] Chunhung Richard Lin and Mario Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected areas in Communications*, 15(7):1265–1275, 1997.

[9] RA Santos, RM Edwards, and NL Seed. Inter vehicular data exchange between fast moving road traffic using an ad-hoc cluster-based location routing algorithm and 802.11 b direct sequence spread spectrum radio. In *PostGraduate Networking Conference*, 2003.

[10] Jung-Hyok Kwon, Hyun Soo Chang, Taeshik Shon, Jai-Jin Jung, and Eui-Jik Kim. Neighbor stability-based vanet clustering for urban vehicular environments. *The Journal of Supercomputing*, 72(1):161–176, 2016.

[11] Saleh Yousefi, Mahmoud Siadat Mousavi, and Mahmood Fathy. Vehicular ad hoc networks (vanets): challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766. IEEE, 2006.

[12] Salim Bitam and Abdelhamid Mellouk. Qos swarm bee routing protocol for wireless sensor networks. In *Programming and Systems (ISPS), 2011 10th International Symposium on*, pages 60–66. IEEE, 2011.

[13] Asim Rasheed, Saira Gillani, Sana Ajmal, and Amir Qayyum. Vehicular ad hoc network (vanet): A survey, challenges, and applications. In *Vehicular Ad-Hoc Networks for Smart Cities*, pages 39–51. Springer, 2017.

[14] Marica Amadeo, Claudia Campolo, Antonella Molinaro, and Giuseppe Ruggeri. A wave-compliant mac protocol to support vehicle-to-infrastructure non-safety applications. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[15] Olivia Brickley, Martin Koubek, Susan Rea, and Dirk Pesch. A network centric simulation environment for calm-based cooperative vehicular systems. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 75. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.

[16] Yasser Toor, Paul Muhlethaler, and Anis Laouiti. Vehicle ad hoc networks: Applications and related technical issues. *IEEE communications surveys & tutorials*, 10(3), 2008.

[17] Panos Papadimitratos, Arnaud De La Fortelle, Knut Evenssen, Roberto Brignolo, and Stefano Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *IEEE communications magazine*, 47(11), 2009.

[18] Hannes Hartenstein, Bernd Bochow, André Ebner, Matthias Lott, Markus Radimirsch, and Dieter Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 259–262. ACM, 2001.

[19] Rupendra Nath Mitra and Dharma P Agrawal. 5g mobile technology: A survey. *ICT Express*, 1(3):132–137, 2015.

[20] Elias C Eze, Si-Jing Zhang, En-Jie Liu, and Joy C Eze. Advances in vehicular ad-hoc networks (vanets): Challenges and road-map for future development. *International Journal of Automation and Computing*, 13(1):1–18, 2016.

[21] Norman Abramson. The aloha system: another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285. ACM, 1970.

[22] Hamid Menouar, Fethi Filali, and Massimiliano Lenardi. A survey and qualitative analysis of mac protocols for vehicular ad hoc networks. *IEEE wireless communications*, 13(5), 2006.

[23] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. Rr-aloha, a reliable r-aloha broadcast channel for ad-hoc inter-vehicle communication networks. In *Proceedings of Med-Hoc-Net*, volume 2002, 2002.

[24] Lin Zhang, Zishan Liu, Rui Zou, Jinjie Guo, and Yu Liu. A scalable csma and self-organizing tdma mac for ieee 802.11 p/1609. x in vanets. *Wireless Personal Communications*, 74(4):1197–1212, 2014.

[25] Vandana Jayaraj, C Hemanth, and RG Sangeetha. A survey on hybrid mac protocols for vehicular ad-hoc networks. *Vehicular communications*, 6:29–36, 2016.

[26] Young-Bae Ko, Vinaychandra Shankarkumar, and Nitin H Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 13–21. IEEE, 2000.

[27] Yamen Y Nasrallah, Irfan Al-Anbagi, and Hussein T Mouftah. Distributed time synchronization mechanism for large-scale vehicular networks. In *Selected Topics in Mobile & Wireless Networking (MoWNeT), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[28] Juan Luo, Junli Zha, Yi Xiao, and Renfa Li. H-mac: a hybrid mac protocol for vanet. In *China Conference on Wireless Sensor Networks*, pages 346–356. Springer, 2012.

[29] R Brendha and V Sinthu Janita Prakash. A survey on routing protocols for vehicular ad hoc networks. In *Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on*, pages 1–7. IEEE, 2017.

[30] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68. IEEE, 2001.

[31] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[32] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[33] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM computer communication review*, volume 24, pages 234–244. ACM, 1994.

[34] Manuel Fogue, Piedad Garrido, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Evaluating the impact of a novel message dissemination scheme for vehicular networks using real maps. *Transportation Research Part C: Emerging Technologies*, 25:61–80, 2012.

[35] Hamid Menouar, Massimiliano Lenardi, and Fethi Filali. Improving proactive routing in vanets with the mopr movement prediction framework. In *Telecommunications, 2007. ITST'07. 7th International Conference on ITS*, pages 1–6. IEEE, 2007.

[36] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[37] Ram Shringar Raw, DK Lobiyal, Sanjoy Das, and Sushil Kumar. Analytical evaluation of directional-location aided routing protocol for vanets. *Wireless Personal Communications*, 82(3):1877–1891, 2015.

[38] Aakash Jasper, Rajeev Paulus, and Shubhanshi Rathore. Improvement of lar1 routing protocol. *Procedia Computer Science*, 57:77–83, 2015.

[39] Jaskaran Preet Singh and Rasmeet S Bali. A hybrid backbone based clustering algorithm for vehicular ad-hoc networks. *Procedia Computer Science*, 46:1005–1013, 2015.

[40] Samira Harrabi, Ines Ben Jaffar, and Khaled Ghedira. Novel optimized routing scheme for vanets. *Procedia Computer Science*, 98:32–39, 2016.

[41] Rasmeet S Bali, Neeraj Kumar, and Joel JPC Rodrigues. Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions. *Vehicular communications*, 1(3):134–152, 2014.

[42] N.V.S. Sree Rathna Lakshmi, S. Babu, and N. Bhalaji. Analysis of clustered qos routing protocol for distributed wireless sensor network. *Computers & Electrical Engineering*, 64:173 – 181, 2017.

[43] Andrea Gorrieri, Marco Martaló, Stefano Busanelli, and Gianluigi Ferrari. Clustering and sensing with decentralized detection in vehicular ad hoc networks.

Ad Hoc Networks, 36:450 – 464, 2016. Vehicular Networking for Mobile Crowd Sensing.

[44] Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 76–85. ACM, 2004.

[45] Mimoza Durresi, Arjan Durresi, and Leonard Barolli. Emergency broadcast protocol for inter-vehicle communications. In *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, volume 2, pages 402–406. IEEE, 2005.

[46] Ozan K Tonguz, Nawaporn Wisitpongphan, and Fan Bai. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. *IEEE Wireless Communications*, 17(2), 2010.

[47] Kashif Naseer Qureshi, Abdul Hanan Abdullah, and Jaime Lloret. Road perception based geographical routing protocol for vehicular ad hoc networks. *International Journal of Distributed Sensor Networks*, 12(2):2617480, 2016.

[48] Xinming Zhang, Xulei Cao, Long Yan, and Dan Keun Sung. A street-centric opportunistic routing protocol based on link correlation for urban vanets. *IEEE Transactions on Mobile Computing*, 15(7):1586–1599, 2016.

[49] Zhenghua Fu, Xiaoqiao Meng, and Songwu Lu. How bad tcp can perform in mobile ad hoc networks. In *Computers and communications, 2002. Proceedings. ISCC 2002. Seventh international symposium on*, pages 298–303. IEEE, 2002.

[50] Marc Bechler, Sven Jaap, and Lars Wolf. An optimized tcp for internet access of vehicular ad hoc networks. In *International Conference on Research in Networking*, pages 869–880. Springer, 2005.

[51] Ralf Schmitz, Alain Leiggener, Andreas Festag, Lars Eggert, and Wolfgang Effelsberg. Analysis of path characteristics and transport protocol design in vehicular ad hoc networks. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 528–532. IEEE, 2006.

[52] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel AF Mini, and Antonio AF Loureiro. Data communication in vanets: Protocols, applications and challenges. *Ad Hoc Networks*, 44:90–103, 2016.

[53] NAYANA P Vaity and DNYANESHWAR V Thombre. A survey on vehicular mobility modeling: flow modeling. *International Journal of Communication Network Security*, 1(4):21, 2012.

[54] Sabih ur Rehman, M Arif Khan, Tanveer A Zia, and Rashid H Khokhar. A synopsis of simulation and mobility modeling in vehicular ad-hoc networks (vanets). *IOSR Journal of Computer Engineering (IOSR-JCE)*, 15(2):1–16, 2013.

[55] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, 2014.

[56] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.

[57] Luciana Moreira Sá De Souza, Harald Vogt, and Michael Beigl. A survey on fault tolerance in wireless sensor networks. *Interner Bericht. Fakultät für Informatik, Universität Karlsruhe*, 2007.

[58] Min Ding, Dechang Chen, Kai Xing, and Xiuzhen Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 902–913. IEEE, 2005.

[59] Ricky W Butler. A survey of provably correct fault-tolerant clock synchronization techniques. *Technical Report NASA-TM-100553*, February 1988.

[60] Santashil PalChaudhuri, Amit Kumar Saha, and DB Johnsin. Adaptive clock synchronization in sensor networks. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pages 340–348. IEEE, 2004.

[61] Wanlu Sun, Fredrik Brannstrom, and Erik G Strom. On clock offset and skew estimation with exponentially distributed delays. In *Communications (ICC), 2013 IEEE International Conference on*, pages 1872–1877. IEEE, 2013.

[62] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[63] Bharath Sundararaman, Ugo Buy, and Ajay D Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad hoc networks*, 3(3):281–323, 2005.

[64] Amulya Ratna Swain and RC Hansdah. A model for the classification and survey of clock synchronization protocols in wsns. *Ad Hoc Networks*, 27:219–241, 2015.

[65] Djamel Djenouri and Miloud Bagaa. Synchronization protocols and implementation issues in wireless sensor networks: A review. *IEEE Systems Journal*, 10(2):617–627, 2016.

[66] Carlos H Rentel. *Network time synchronization and Code-based Scheduling for Wireless Ad Hoc Network*. Carleton University Ottawa, Canada, 2006.

[67] Yuzhuo Zhan, Weimin Lei, and Wei Zhang. Survey on time synchronization technology in networks. In *Control and Decision Conference (CCDC), 2015 27th Chinese*, pages 2157–2161. IEEE, 2015.

[68] Aneeq Mahmood, Reinhard Exel, Henning Trsek, and Thilo Sauter. Clock synchronization over ieee 802.11-a survey of methodologies and protocols. *IEEE Transactions on Industrial Informatics*, 13(2):907–922, 2017.

[69] Koshal Arvind. Probabilistic clock synchronization in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 5(5):474–487, 1994.

[70] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49. ACM, 2004.

[71] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149. ACM, 2003.

[72] Suyoung Yoon, Chanchai Veerarittiphan, and Mihail L Sichitiu. Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 3(2):8, 2007.

[73] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[74] Kyoung-lae Noh, Erchin Serpedin, and Khalid Qaraqe. A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization. *IEEE Transactions on Wireless Communications*, 7(9), 2008.

[75] André Ebner, Lars Wischhof, and Hermann Rohling. Aspects of decentralized time synchronization in vehicular ad hoc networks. *Proc. 1st Int. WIT*, pages 67–72, 2004.

[76] David L Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on communications*, 39(10):1482–1493, 1991.

[77] DM Anand, Deepak Sharma, Y Li-Baboud, and James Moyne. Eda performance and clock synchronization over a wireless network: Analysis, experimentation and application to semiconductor manufacturing. In *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on*, pages 1–6. IEEE, 2009.

[78] Aneeq Mahmood, Georg Gaderer, and Patrick Loschmidt. Clock synchronization in wireless lans without hardware support. In *Factory Communication Systems (WFCS), 2010 8th IEEE International Workshop on*, pages 75–78. IEEE, 2010.

[79] Steven E Butner and Scott Vahey. Nanosecond-scale event synchronization over local-area networks. In *Local Computer Networks, 2002. Proceedings. LCN 2002. 27th Annual IEEE Conference on*, pages 261–269. IEEE, 2002.

[80] Dong Zhou and Ten-Hwang Lai. A scalable and adaptive clock synchronization protocol for ieee 802.11-based multihop ad hoc networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[81] David Stanislowski, Xavier Vilajosana, Qin Wang, Thomas Watteyne, and Kristofer SJ Pister. Adaptive synchronization in ieee802. 15.4 e networks. *IEEE Transactions on Industrial Informatics*, 10(1):795–802, 2014.

[82] Lifei Huang and Ten-Hwang Lai. On the scalability of ieee 802.11 ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 173–182. ACM, 2002.

[83] Jui-Hao Chiang and Tzi-cker Chiueh. Accurate clock synchronization for ieee 802.11-based multi-hop wireless networks. In *Network Protocols, 2009. ICNP 2009. 17th IEEE International Conference on*, pages 11–20. IEEE, 2009.

[84] Sahar Ben Yaala, Fabrice Théoleyre, and Ridha Bouallegue. Cooperative resynchronization to improve the reliability of colocated ieee 802.15. 4-tsch networks in dense deployments. *Ad Hoc Networks*, 2017.

[85] Juan Li, Hao Zhang, Xuerong Cui, and Yongping Xiong. Analysis on the characteristics of timing synchronization samples of ieee 802.11 p. In *Identification, Information, and Knowledge in the Internet of Things (IIKI), 2015 International Conference on*, pages 219–222. IEEE, 2015.

[86] Jong-Kwan Lee, Min-Hwan Cheon, and Jaesung Lim. A time synchronization scheme with best reference selection for mobile ad hoc networks.

[87] Weilian Su and Ian F Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 13(2):384–397, 2005.

[88] Qun Li and Daniela Rus. Global clock synchronization in sensor networks. *IEEE Transactions on computers*, 55(2):214–226, 2006.

[89] Chafika Benzaïd, Miloud Bagaa, and Mohamed Younis. An efficient clock synchronization protocol for wireless sensor networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, pages 718–723. IEEE, 2014.

[90] Reza Khoshdelniat, Moh Lim Sim, Hong Tat Ewe, and Tan Su Wei. Time table transfer time syncronization in mobile wireless sensor networks. In *PIERS Proceedings*, pages 1724–1728, 2009.

[91] Khedidja Medani, Makhlouf Aliouat, and Zibouda Aliouat. High velocity aware clocks synchronization approach in vehicular ad hoc networks. In *IFIP International Conference on Computer Science and its Applications_x000D_*, pages 479–490. Springer, 2015.

[92] Philipp Sommer and Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 37–48. IEEE, 2009.

[93] Kittipat Apicharttrisorn, Supasate Choochaisri, and Chalermek Intanagonwiwat. Energy-efficient gradient time synchronization for wireless sensor networks. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, pages 124–129. IEEE, 2010.

[94] Kay Römer, Philipp Blum, and Lennart Meier. Time synchronization and calibration in wireless sensor networks. *Handbook of sensor networks: Algorithms and architectures*, 49:199, 2005.

[95] Walter Franz, Hannes Hartenstein, and Martin Mauve. *Inter-vehicle-communications Based on Ad Hoc Networking Principles: The FleetNet Project.* Universitätsverlag Karlsruhe, 2005.

[96] Dahlia Sam, Cyrilraj Velanganni, and T Esther Evangelin. A vehicle control system using a time synchronized hybrid vanet to reduce road accidents caused by human error. *Vehicular Communications*, 6:17–28, 2016.

[97] Matthias Függer, Thomas Nowak, and Bernadette Charron-Bost. Diffusive clock synchronization in highly dynamic networks. In *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, pages 1–6. IEEE, 2015.

[98] Habib Aissaoua, Makhlouf Aliouat, Ahcène Bounceur, and Reinhardt Euler. A distributed consensus-based clock synchronization protocol for wireless sensor networks. *Wireless Personal Communications*, pages 1–22, 2017.

[99] Branislav Kusy, Prabal Dutta, Philip Levis, Miklos Maroti, Akos Ledeczi, and David Culler. Elapsed time on arrival: a simple and versatile primitive for canonical time synchronisation services. *International Journal of Ad Hoc and Ubiquitous Computing*, 1(4):239–251, 2006.

[100] M Chitra and S Siva Sathya. Efficient broadcasting mechanisms for data dissemination in vehicular ad hoc networks. *International Journal of Mobile Network Communications & Telematics (IJMNCT)*, 3(3):47–63, 2013.

[101] Elena Fasolo, Andrea Zanella, and Michele Zorzi. An effective broadcast scheme for alert message propagation in vehicular ad hoc networks. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 9, pages 3960–3965. IEEE, 2006.

[102] Jung-Hun Kim and SuKyoung Lee. Reliable routing protocol for vehicular ad hoc networks. *AEU-International Journal of Electronics and Communications*, 65(3):268–271, 2011.

[103] Degan Zhang, Guang Li, Ke Zheng, Xuechao Ming, and Zhao-Hua Pan. An energy-balanced routing method based on forward-aware factor for wireless sensor networks. *IEEE transactions on industrial informatics*, 10(1):766–773, 2014.

[104] De-gan Zhang, Xiao-dong Song, Xiang Wang, and Yuan-ye Ma. Extended aodv routing method based on distributed minimum transmission (dmt) for wsn. *AEU-International Journal of Electronics and Communications*, 69(1):371–381, 2015.

[105] De-gan Zhang, Ke Zheng, Ting Zhang, and Xiang Wang. A novel multicast routing method with minimum transmission for wsn of cloud computing service. *Soft Computing*, 19(7):1817–1827, 2015.

[106] De-gan Zhang, Xiang Wang, Xiao-dong Song, Ting Zhang, and Ya-nan Zhu. A new clustering routing method based on pece for wsn. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–13, 2015.

[107] De-gan Zhang, Si Liu, Ting Zhang, and Zhao Liang. Novel unequal clustering routing protocol considering energy balancing based on network partition &

distance for mobile education. *Journal of Network and Computer Applications*, 88:1–9, 2017.

[108] Chan-Ki Park, Min-Woo Ryu, and Kuk-Hyun Cho. Survey of mac protocols for vehicular ad hoc networks. *SmartCR*, 2(4):286–295, 2012.

[109] Manju Khurana, Ranjana Thalore, Vikas Raina, and Manish Kumar Jha. Improved time synchronization in ml-mac for wsn using relay nodes. *AEU-International Journal of Electronics and Communications*, 69(11):1622–1626, 2015.

[110] Yamen Y Nasrallah, Irfan Al-Anbagi, and Hussein T Mouftah. Distributed time synchronization mechanism for large-scale vehicular networks. In *Selected Topics in Mobile & Wireless Networking (MoWNeT), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[111] Yamen Y Nasrallah, Irfan Al-Anbagi, and Hussein T Mouftah. Qos-based distributed time synchronization mechanism for high intensity vehicular networks. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 958–963. IEEE, 2016.

[112] Degan Zhang, Xiang Wang, Xiaodong Song, and Dexin Zhao. A novel approach to mapped correlation of id for rfid anti-collision. *IEEE Transactions on Services Computing*, 7(4):741–748, 2014.

[113] Degan Zhang, Chen-peng Zhao, Yan-pin Liang, and Zhao-jing Liu. A new medium access control protocol based on perceived data reliability and spatial correlation in wireless sensor network. *Computers & Electrical Engineering*, 38(3):694–702, 2012.

[114] De-gan Zhang, Ke Zheng, De-xin Zhao, Xiao-dong Song, and Xiang Wang. Novel quick start (qs) method for optimization of tcp. *Wireless Networks*, 22(1):211–222, 2016.

[115] De-gan Zhang and Xiao-dan Zhang. Design and implementation of embedded un-interruptible power supply system (eupss) for web-based mobile application. *Enterprise Information Systems*, 6(4):473–489, 2012.

[116] De-Gan Zhang. A new approach and system for attentive mobile learning based on seamless migration. *Applied Intelligence*, 36(1):75–89, 2012.

[117] De-gan Zhang and Yan-pin Liang. A kind of novel method of service-aware computing for uncertain mobile applications. *Mathematical and Computer Modelling*, 57(3):344–356, 2013.

[118] De-gan Zhang, Xiao-dong Song, Xiang Wang, Ke Li, Wen-bin Li, and Zhen Ma. New agent-based proactive migration method and system for big data environment (bde). *Engineering Computations*, 32(8):2443–2466, 2015.

[119] Degan Zhang, Xuejing Kang, and Jinghui Wang. A novel image de-noising method based on spherical coordinates system. *EURASIP Journal on Advances in Signal Processing*, 2012(1):110, 2012.

[120] Zhigang Wang, Lichuan Liu, MengChu Zhou, and Nirwan Ansari. A position-based clustering technique for ad hoc intervehicle communication. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):201–208, 2008.

[121] Mildred M Caballeros Morales, Choong Seon Hong, and Young-Cheol Bang. An adaptable mobility-aware clustering algorithm in vehicular networks. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1–6. IEEE, 2011.

[122] S Almalag Mohammad and C Weigle Michele. Using traffic flow for cluster formation in vehicular ad-hoc networks. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 631–636. IEEE, 2010.

[123] Weiwei Li, Ali Tizghadam, and Alberto Leon-Garcia. Robust clustering for connected vehicles using local network criticality. In *Communications (ICC), 2012 IEEE International Conference on*, pages 7157–7161. IEEE, 2012.

[124] Sheng-Shih Wang and Yi-Shiun Lin. Passcar: A passive clustering aided routing protocol for vehicular ad hoc networks. *Computer Communications*, 36(2):170–179, 2013.

[125] De-gan Zhang, Ya-nan Zhu, Chen-peng Zhao, and Wen-Bo Dai. A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the internet of things (iot). *Computers & Mathematics with Applications*, 64(5):1044–1055, 2012.

[126] Jijun Yin, Tamer ElBatt, Gavin Yeung, Bo Ryu, Stephen Habermas, Hariharan Krishnan, and Timothy Talty. Performance evaluation of safety applications over dsrc vehicular ad hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 1–9. ACM, 2004.

[127] Khedidja Medani, Makhlouf Aliouat, and Zibouda Aliouat. Fault tolerant time synchronization using offsets table robust broadcasting protocol for vehicular ad hoc networks. *AEU-International Journal of Electronics and Communications*, 2017.

[128] Thore Graepel. Statistical physics of clustering algorithms. *Technical Re-port*, 171822, 1998.

[129] Soumen Chakrabarti, Earl Cox, Eibe Frank, Ralf Hartmut Güting, Jiawei Han, Xia Jiang, Micheline Kamber, Sam S Lightstone, Thomas P Nadeau, Richard E Neapolitan, et al. *Data mining: know it all.* Morgan Kaufmann, 2008.

[130] Craig Cooper, Daniel Franklin, Montserrat Ros, Farzad Safaei, and Mehran Abolhasan. A comparative survey of vanet clustering techniques. *IEEE Communications Surveys & Tutorials*, 19(1):657–681, 2017.

[131] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.

[132] Hang Su and Xi Zhang. Clustering-based multichannel mac protocols for qos provisionings over vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 56(6):3309–3323, 2007.

[133] Yvonne Guenter, Bernhard Wiegel, and Hans Peter Großmann. Medium access concept for vanets based on clustering. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2189–2193. IEEE, 2007.

[134] Omer Kayis and Tankut Acarman. Clustering formation for inter-vehicle communication. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 636–641. IEEE, 2007.

[135] Zhigang Wang, Lichuan Liu, MengChu Zhou, and Nirwan Ansari. A position-based clustering technique for ad hoc intervehicle communication. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):201–208, 2008.

[136] Hao Wu, Zhangdui Zhong, and Lajos Hanzo. A cluster-head selection and update algorithm for ad hoc networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[137] Khalid Abdel Hafeez, Lian Zhao, Zaiyi Liao, and Bobby Ngok-Wah Ma. A fuzzy-logic-based cluster head selection algorithm in vanets. In *Communications (ICC), 2012 IEEE International Conference on*, pages 203–207. IEEE, 2012.

[138] Zaydoun Y Rawashdeh and Syed Masud Mahmud. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):15, 2012.

[139] Efi Dror, Chen Avin, and Zvi Lotker. Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks. *Ad Hoc Networks*, 11(7):2002–2015, 2013.

[140] Behnam Hassanabadi, Christine Shea, L Zhang, and Shahrokh Valaee. Clustering in vehicular ad hoc networks using affinity propagation. *Ad Hoc Networks*, 13:535–548, 2014.

[141] Mengying Ren, Lyes Khoukhi, Houda Labiod, Jun Zhang, and Véronique Vèque. A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (vanets). *Vehicular Communications*, 2016.

List of included publications

The thesis is based on the following appended papers:

A) MEDANI, Khedidja, ALIOUAT, Makhlouf, et ALIOUAT, Zibouda. High Velocity Aware Clocks Synchronization Approach in Vehicular Ad Hoc Networks. In : IFIP International Conference on Computer Science and its Applications_x000D_. Springer, Cham, 2015. p. 479-490.

B) MEDANI, Khedidja, ALIOUAT, Makhlouf, et ALIOUAT, Zibouda. Fault tolerant time synchronization using offsets table robust broadcasting protocol for vehicular ad hoc networks. AEU-International Journal of Electronics and Communications, 2017, vol. 81, p. 192-204.

C) MEDANI, Khedidja, ALIOUAT, Makhlouf, et ALIOUAT, Zibouda. Impact of Clustering Stability on the Improvement of Time Synchronization in VANETs. In : IFIP International Conference on Computational Intelligence and Its Applications. Springer, 2018. p. 472

D) MEDANI, Khedidja, ALIOUAT, Makhlouf, et ALIOUAT, Zibouda. Clock Synchronization in Vehicular Ad hoc Sensor Networks: A Survey. Submitted paper.

Abstract

Aiming to the main purpose of improving road safety and entertainment, Vehicular Ad Hoc Networks (VANETs) have emerged as an open and interesting research topic in the last decade. The deployment of VANET systems coupled with sensor technologies has increased the gained benefits from the development of these latter. Thus, they enable real-time data gathering and sharing, so that new applications, such as traffic reporting, relief to environmental monitoring and distributed surveillance are promoted. The design of reliable, fault-tolerant, maintainable, safe and secure applications and standard for realistic and large-scale deployment environment, like in VANETs presents extraordinary challenge, especially in the lack of global memory enabling global system state recognition. In this context, the requirement of clock synchronization remains one of the most significant issues that should be addressed to the extent of that dependable systems evolve. The focal point of this Doctoral dissertation is to give an analytical study of clock synchronization issue in vehicular communication systems. The intrinsic characteristics of the unstable vehicular environments, such as the high speed of nodes and the lack of permanent network connectivity have created new challenges and requirements, so that the solutions that have already proposed to synchronize nodes in classical networks are no longer appropriate. Consequently, new and adaptive clock synchronization mechanisms should be devised and implemented. Here, we propose a new mechanism for synchronizing clocks in vehicular environments, dealing so with communication and scalability issues. The proposal, named as "Offsets Table Robust Broadcasting" (OTRB), exploits the broadcasting channel to spread the time information over the entire network. This protocol is well-adapted to random network topology changes, high node velocity while offering good precision and robustness against nodes failure and packet loss. The analytical study and protocol simulation for evaluating the system performance, carried out by a combination of VanetMobiSim and NS2 simulators, have yielded convincing results, outperforming those exhibited by the basic referred protocols.

Keywords: Intelligent transportation system, ITS, VANET, WSN, dependability, clock synchronization, fault-tolerant clock synchronization.

ملــخص

عرفت الشبكات اللامركزية للمركبات المعروفة اختصارا بـ شبكات "فانيت" اهتماما بالغا في الأوساط العلمية في العشرية الأخيرة ضمن مجالات البحث المفتوحة، تهدف هذه الشبكات بالأساس لتحقيق أكبر قدر من الأمان و الترفيه خلال حركة سير العربة من خلال تفعيل خاصية تجميع و مشاركة المعلومات بصفة آنية. هذا و قد أدى الإستعمال المشترك لشبكات فانيت و تكنولوجيات أجهزة الإستشعار الإلكترونية إلى تحسين مزايا هذه التقنية بشكل واضح، حيث برزت من خلال ذلك تطبيقات جديدة تعد بالكثير ، نذكر منها: تطبيقات خاصة بإصدار تقارير عن حركة المرور , تطبيقات المساعدة على مراقبة المحيط و تطبيقات المراقبة في بيئة الأنظمة الموزعة.

غياب ذاكرة موحدة للنظام ككل في هذه التقنية يعتبر تحديا حقيقيا في سبيل إنشاء تطبيقات تتمتع بكافة المزايا المتعلقة بالنجاعة و الكفاءة في التعامل مع الأعطاب و كذا قابلية الصيانة و تحقيق السلامة و الأمان، ذلك لأن الذاكرة الموحدة كانت لتسهل عملية التعرف على الحالة الإجمالية للنظام و هذه تعتبر حجر الأساس لإبقاء أي نظام موزع قيد التشغيل، و بخاصة في الأنظمة الموزعة الواقعية واسعة النطاق. في هذا السياق برزت الحاجة لاستعمال ساعة تقوم بعمليات المزامنة في الأنظمة الموزعة، و من هذا المنطلق أصبح التفكير في تحسين أداء هذه الساعة من أهم القضايا الواجب التفكير فيها تماشيا مع التطور و الاستعمال المتزايد لهذه الأنظمة. إذا كان تواجد ساعة المزامنة مهما في أي نظام مهما كان نوعه، فإن تواجدها في أنظمة الفانيت و أنظمة أجهزة الإستشعار اللاسلكية يعد جزءا قاعديا ضمن بنية الاتصالات التي يقوم عليها عمل هذا النوع من الأنظمة، و ليس هذا فحسب، فالساعة توفر إمكانية كشف الحركة و الموقع، قياس المحاذاة مع المحيط، فضلا عن إمكانيات أخرى في محيط الشبكة. يواجه عمل هذه الساعة صعوبات تكمن في جوهر أنظمة الفانيت نفسها، حيث غالبا ما تعمل هذه الأنظمة في بيئات تعاني من عدم الاستقرار في شبكة الاتصال بسبب الانقطاعات المتكررة، كذلك قد يواجه النظام صعوبة في تتبع الهدف بسبب سرعته الكبيرة، هذه الصعوبات و أخرى فرضت تحديات جديدة لأن كل الحلول التي تم طرحها في السابق للشبكات الكلاسيكية أضحت غير مناسبة, لذلك ينبغي التفكير في آلية جديدة لعمل الساعة يمكنها التعامل مع الصعوبات المطروحة على مستوى الاتصال و على مستوى قدرة النظام على التكيف.

إذن، الموضوع المحوري لرسالة الدكتوراه التي بين أيدينا يتعلق بإجراء دراسة تحليلية حول المشاكل المرتبطة بعمل ساعة المزامنة ضمن بيئة نظام فانيت، نسعى من خلال ذلك لاقتراح آلية تعتبر كمساهمة مهمة في ضمان كفاءة عالية للنظام. هذا المقترح الذي يأتي تحت مسمى : "البث الناجع لجدول الإرسال" هو عبارة عن بروتوكول يتم فيه استغلال قناة البث لنشر المعلومة اللحظية عبر كامل الشبكة، هذا البروتوكول يعمل بكفاءة في البيئات التي تتغير فيها عناصر الشبكة بشكل عشوائي و ترتفع فيها سرعة الهدف، حيث يضمن دقة عالية و مرونة كبيرة في التعامل مع اختلالات الشبكة او في الهدف نفسه. إعتمدنا خلال هذه الدراسة التحليلة للبروتوكول المقترح على برامج محاكاة و هي VanetMobiSim و NS2 سمحت لنا بقياس و تقييم عمل النظام باستعمال البروتوكول و جاءت النتائج مقنعة لحد بعيد حيث تجاوزت البروتوكولات المرجعية التي اعتُمدت في الدراسة.

كلمات دلالية: أنظمة النقل الذكية, فانيت, الكفاءة, مزامنة الساعة.