UNIVERSITY OF THESSALY, GREECE

MESSAGE DISSEMINATION IN VEHICULAR AD HOC NETWORKS

by

Leandros A. Maglaras

A DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Leandros A. Maglaras, Candidate

Dissertation Committee:

Prof. Leandros Tassiulas, Chairman

Prof. George Stamoulis

Prof. George Liberopoulos

ACKNOWLEDGMENTS

First of all, I want to thank the supervisor of my thesis, Professor Leandros Tassiulas, for his valuable help and guidance during my work. I am also grateful to the other members of the selection committee of my thesis, Professors George Stamoulis, George Liberopoulos, Dimitrios Katsaros, Iordanis Koutsopoulos, Athanasios Korakis and Antonios Argyriou for careful reading of my work and for their valuable suggestions. I owe special thanks to Lecturer Dimitrios Katsaros who helped throughout all my research and played a key role in the completion of my PhD.

I would like to thank my colleagues for their invaluable help in different technical and procedural issues. Special thanks to Pavlos Basaras and Nikos Makris for their cooperation and help that they gave me in all the circumstances needed during the last 2 years. I want to thank Apostolos Apostolaras and Antonis Tzounis for their hospitality and their overall help when visiting Volos.

Special thanks to my family and friends who gave me courage all these years to continue my efforts for the completion of the dissertation. Many thanks to my friend Pavlina who had the courage to read and review the first draft of the dissertation. I would like to thank to my wife Zoe for her understanding, encouragement, quiet patience and love, especially during the last months of my trying. Above all, I am grateful to my parents, Athanasios and Eirini and my sister Tini for their wholehearted love and support over all these years.

I wish to acknowledge the EU-FP7 project REDUCTION "Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behaviour Adaptation"(# 288254), which supported the research related to this thesis

After all everything that has a beginning has an end.

Dedicated to my family and my son.

ABSTRACT OF THE DISSERTATION

Message Dissemination in Vehicular ad hoc networks
by

Leandros Maglaras

Doctor of Philosophy, Post Graduate Program in Electrical and Computer Engineering,
University of Thessaly, Greece.
June 2014
Prof. Leandros Tassiulas, Chairperson

A vehicular network is a challenging environment since it combines a fixed infrastructure (roadside units, e.g., proxies), and ad hoc communications among vehicles. Vehicular networks have the diverse range of applications that varies from safety applications to comfort applications. As the dissemination requirements (reliability, delay, coverage, etc.) may be different from one application to another, the dissemination mechanism could be implemented at the application layer where the interaction with the routing layer is limited. This thesis addresses several aspects of vehicular communications related to data dissemination.

This thesis concerns with centrality metrics in ad hoc networks. We present a centrality metric that characterize the network topology using only limited, local connectivity information one or two hop information. We quantify a nodes' centrality based on the number of neighbors a node belongs to and the ranking this node has in each one. That way nodes that belong to many clusters and participate in many communication paths is ranked high. No nodes remain unranked while for large networks the method makes similar to PageRank rankings, even though it is a localized metric whereas PageRank requires cumbersome computations and knowledge of the whole networks topology.

In the present work we also create novel clustering methods that work efficiently both in highway and urban environments. The methods combine different information such as current and future possible position of vehicles, relative mobility, vehicles' direction and height in order to create stable clusters. Historical data of drivers are also used in order to group those that share common habits and thus mobility patterns. Social aspects of driving are combined with mathematically measurable parameters for cluster stability, with the use of DSRC communication capabilities.

This thesis addresses the issue of delay constraints in packet scheduling routing in

both static and mobile ad hoc networks. We present novel methods for delay efficient routing of packets based on the well known Back-pressure routing protocol. Large delays that appear when network load is low or medium, are addressed with the combination of packets' travel history or with the segmentation of the network in clusters. Both methods manage to perform effectively in terms of mean delay, packet throughput and energy consumption.

A Push-based data dissemination mechanism that use fixed RSUs is also presented that addresses the issue of skewed access pattern. In real life applications, some items are more frequently accessed by clients than some others and the correct indexing of these data is very important. The method exploits the different access probabilities of data items, in order to improve the mean tuning time, while retaining the same access time low. The presence of air indexes would give an answer about the time of broadcast and is appropriate for helping the driver choose the correct driving behavior in order to retrieve this needed information.

We also present a pull-based congestion avoidance method for VANETs. We propose a real time system based on DSRC communication capabilities in order to reroute vehicles to the most ecological route, avoiding congested roads and balancing overall travel time and C02 emissions. The proposed method uses real time information broadcasted from passing vehicles to RSUs scattered along the road in order to assign weights to different road segments according to time, distance and CO2 emissions. The method can be used for traffic management situations that arise after an accident in a city.

ΠΕΡΙΛΗΨΗ

Μετάδοση πληροφορίας σε ασύρματα δίκτυα οχημάτων
Συγγραφή:
Λέανδρος Μαγλαράς

Πανεπιστήμιο Θεσσαλίας.
Ιούνιος 2014
Επιβλέπων καθηγητής. Λέανδρος Τασιούλας

Το δίκτυο οχημάτων είναι ένα απαιτητικό περιβάλλον καθώς συνδυάζει $ad - hoc$ επικοινωνίες μεταξύ των οχιμάτων με σταθερές υποδομές (μονάδες τοποθετιμένεσστις άκρες των δρόμων). Οι εφαρμογές που υπάρχουν για τα δίκτυα αυτα εκτείνονται από απλά κοινωνικά δίκτψα εως την διαμοίραση αρξείων με σψνεργατικέω μεθόδους για θέματα ασφαλείας Οι απαιτήσεις που υπάρχουν, όσον αφορά την μέση και μέγιστη καθυστέερηση στην αποστολή των πακέτων, στην αξιοπιστία, στην κάλυση του δικτύου κ.λ.π. διαφέρουν ανάλογα με το είδος της εφαρμογής. Αυτό το γεγονός οδηγεί στην ανάγκη για δημιουργία διαφορετικών μηχανισμών μετάδοσης της πληροφορίας.

Αυτή η διατριβή ασχολείται με θέματα υπολογισμού της κεντρικότητας των κόμβων. Παρουσιάζουμε ένα μέτρο που χαραχτηρίζει την τοπολογία του δικύου χρησιμοποιώντας μόνο περιορισμένη, τοπική πληροφορία. Η ποσοτικοποίηση της κεντρικότητας κάθε κόμβου γίνεται με βάση τις γειτονιές στις οποίες ανήκει ο εκάστοτε κόμβος και την κατάταξη που έχει σε καθε μια από αυτές. Με αυτό τον τρόπο, κόμβοι οι οποίοι ανήκουν σε πολλές γειτονιές ενώ ταυτόχρονα συμμετέχουν σε πολλά μονοπάτια επικοινωνίας, σε επίπεδο γειτονιάς, βαθμολογούνται υψηλά. Με το πέρας της μεθόδου, κανένας κόμβος δεν έξει μείνει αδιαβάθμητος, ενώ για μεγάλα δίκτυα η μέθοδος δίνει αποτελέσματα παρόμοια με την μέθοδο $PageRank$, παρόλο που είναι μια καθαρά τοπική μέθοδος σε αντίθεση με την $Pagerank$ που απιτεί την γνώση όλου του δικτύου.

Στην παρούσα δουλειά παρουσιάζουμε καινοτόμες μεθόδους ομαδοποίησης κόμβων που αποδίδουν το ίδιο καλά τόσο σε αυτοκινητόδρομους όσο και σε αστικά περιβάλλοντα. Οι μέθοδοι συνδυάζουν πληροφορίες όπως τωρινή και μελλοντική θέση των οχημάτων, σχετική ταχύτητα, κατεύθυνση, και ύψος ώστε να δημιουργήσουν σταθερές δομές. Χρησιμοποιούνται επίσης ιστορικά στοιξεία των οδηγών, ώστε να βελτιωθεί η σταθερότητα των μεθόδων, ομαδοποιώντας οδηγούς που έχουν κοινές σψνήθειες και πορείες μέσα σε ένα αστικό περιβάλλον.

Η παρούσα διατριβή καταπιάνεται με θέματα βελτίωσης της καθυστέτησης μετάδοσης

των πακέτων σε στατικά και δυναμικά δίκτυα. Παρουσιάζουμε καινοτόμες μεθόδου που επιτυγχάνουν αποδοτική δρομολόγηση των πακέτων όσον αφορά την μέση καθυστέρηση. Οι μέθοδοι βασίζονται στην μέθοδο δρομολόγησης *Backpressure*. Η σψγκεκριμένη μέθοδος παρουσιάζει μεγάλες καθυστερήσεις στην δρομολόγηση των πακέτων όταν ο φόρτος στο δίκτυο δεν είναι μεγάλος. Το πρόβλημα αυτό αντιμετωπίζεται με τον τεμαχισμό του δικτύου σε ομάδες κόμβων και με την χρήση επιπλέον πληροφορίας που τα ίδια τα πακέτα φέρουν σχετικά με την εως τώρα πορεία τους μέσα στο δίκτυο. Οι προτεινόμενες μέθοδοι επιτυγχάνουν υψηλή απόδοση σε συνδυασμό με μικρή καθυστερήση στην μετάδοση των πακέτων.

Παρουσιάζουμε επίσης έναν μηχανισμό διάδοσης πληροφορίας που βασίζεται σε $I2V$ επικοινωνίες. Ο συγκεκριμένος μηχανισμός έχει χρησιμότητα σε πραγματικές εφαρμογές όπου οι χρήστες κάποια δεδομένα τα προσπελαύνουν ποιο συχνά από τα άλλα. Η δημιουργία ενός κατάλληλου ευρετηρίου είναι πολύ σημαντική σε τέτοιες περιπτώσεις. Η προτεινόμενη μέθοδος εκμεταλλεύεται τις διαφορετικές πιθανότητες προσπέλασης των δεδομένων, με σκοπό να βελτιώσει τον μέσο χρόνο που ο χρήστης πρέπει να είναι συντονισμένος στο κανάλι επικοινωνίας ενώ ταυτόχρονα διατηρεί τον μέσο χρόνο πρόσβασης σε χαμηλά επίπεδα. Η παρουσία ευρετηρίων στον αέρα μπορεί να πληροφορήσει τον οδηγό για την συγκεκριμένη στιγμή που θα μεταδοθεί η πληροφορία που χρειάζεται, βοηθώντας τον οδηγό να επιλέξει την κατάλληλη οδηγική συμπεριφορά ώστε να παραλάβει την πληροφορία από το παρακείμενο σταθμό εκομπής RSU.

Παρουσιάζουμε τέλος μια μέθοδο μείωσης της κυκλοφοριακής συμφόρησης που βασίζεται σε μετάδοση πληροφορίας από ειδικούς κόμβους που έχουν τοποθετηθεί σε κομβικά σημεία στο δίκτυο ($RSUs$). Προτείνουμε ένα σύστημα που δουλεύει σε πραγματικό ξρόνο και που χρησιμοποιεί $DSRC$ επικοινωνίες ώστε να ανακατευθύνει τα οχήματα σε δρόμους με μικρότερη κυκλοφοριακή συμφόρηση εξισσοροπόντας τον ολικό χρόνο ταξιδιού και μειώνοντας τις εκπομπές διοξειδίου του άνθρακα. Η προτεινόμενη μέθοδος χρησιμοποιεί πληροφορίες που εκπέμπονται από διερχόμενα οχήματα σε πραγματικό χρόνο, ώστε να αποδόσει βάρη στους δρόμους, που υπολογίζονται από έναν συνδυασμό μέσου χρόνου που χρειάζεται ένα όχημα για να διασχίσει τον δρόμο αυτό, μέσες εκμομπές διοξειδίου του άνθρακα και το μήκος του δρόμου. Η μέθοδος μπορεί να χρησιμοποιηθεί σε περιπτώσεις όπου απαιτείται τοπική ρύθμιση της πληροφορίας λόγω ενός ατυχήματος.

Table of Contents

Related Publications

The ideas presented in this thesis appear in the following publications:

Journal submissions

- **[S.01]** Leandros Maglaras and Dimitrios Katsaros, **“Social Clustering of vehicles based on Semi-Markov processes”**. Submitted to IEEE transactions on Vehicular Technology, 2013 (under second round review).
- **[S.02]** Dimitrios Papakostas, Leandros Maglaras and Dimitrios Katsaros, **“A Rich Dictionary Markov Predictor for Trajectory Forecasting in Vehicular Environments”**, Submitted to IEEE Vehicular Technology Magazine 2013 (under second round review).
- **[S.03]** Leandros Maglaras and Dimitrios Katsaros **“Delay Efficient Backpressure Routing in Wireless Ad Hoc Networks”**. Submitted to Mobile Communications and Applications (EAI) 2014.
- **[S.04]** Pavlos Basaras, Leandros Maglaras and Dimitrios Katsaros, **“On the utility of V2V communications to CO2 emissions reduction in intelligent transportation systems”**, Submitted to IEEE Communications Magazine 2014

Journal publications

- **[J.01]** Leandros Maglaras and Dimitrios Katsaros, **“New measures for characterizing the significance of nodes in wireless ad hoc networks via localized path-based neighborhood analysis”**, Social Network Analysis and Mining (Springer), vol. 2, no. 2, pp. 97-106, June, 2012.

Conference publications

- **[C.07]** Leandros Maglaras, Pavlos Basaras and Dimitrios Katsaros. **”Exploiting Vehicular Communications for Reducing CO2 Emissions in Urban environments”’**, In Proceedings of the 2nd International Conference on Connected Vehicles and Expo (ICCVE) @ Track: Cooperative Driving, Intelligent and Autonomous Vehicles, Las Vegas, Nevada, USA, December 2-6, 2013.
- **[C.06]** Leandros Maglaras and Dimitrios Katsaros, **“Enhanced Spring Clustering in VANETs with Obstruction Considerations”**, In Proceedings of the 77th IEEE Vehicular Technology Conference - Spring (VTC-Spring), Dresden, Germany, June 2-5, 2013.
- **[C.05]** Leandros Maglaras and Dimitrios Katsaros, **“Clustering in Urban Environments: Virtual Forces Applied to Vehicles”**, In Proceedings of the IEEE Workshop on Emerging Vehicular Networks: V2V/V2I and Railroad Communications (ENV), In

conjunction with the IEEE International Conference on Communications (ICC), pp. 494-498, Budapest, Hungary, June 9-13, 2013.

[C.04] Leandros Maglaras and Dimitrios Katsaros, **"Distributed clustering in vehicular networks"**. In Proceedings of the International Workshop on Vehicular Communications and Networking (VECON), 2012.

[C.03] Leandros Maglaras and Dimitrios Katsaros, **"Distributed Skip Air Index for Smart Broadcasting in Intelligent Transportation Systems"**, In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), IEEE Intelligent Transportation Systems Society (ITSS), pp. 624-629, Alcala' de Henares, Madrid, Spain, June 3-7, 2012.

[C.02] Leandros Maglaras and Dimitrios Katsaros, **"Layered backpressure scheduling for delay reduction in ad hoc networks"** In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011.

[C.01] Leandros Maglaras and Dimitrios Katsaros, **"New metrics for characterizing the significance of nodes in wireless networks via path-based neighborhood analysis"**, In Proceedings of the Panhellenic Conference on Informatics (PCI), IEEE Press, pp. 143-147, Corfu Island, Greece, September 10-12, 2009.

List of Tables

List of Figures

List of Abbreviations

BFS	....	Breadth-First Search
C2X	....	Car-to-X
CH	....	Cluster Head
CPU	....	Central processing unit
CR	....	Communication Range
CAM	....	Cooperative Awareness Message
DSRC	....	Dedicated Short-Range Communication
DENM	....	Decentralized Environment Notification Message
DTN	....	Delay Tolerant Network
EnodeB	....	Evolved Node B
ETSI	....	European Telecommunications Standards Institute
FPLS	....	free-space path loss
GPS	....	Global Positioning System
IVC	....	Inter-Vehicle Communication
LOS	....	Line-of-Sight
LTE	....	Long Term Evolution
ID	....	identification number
ITS	....	Intelligent Transportation Systems
MAC	....	Media Access Control
MANET	....	Mobile Ad Hoc Network
NLOS	....	Non Line Of Sight
OBU	....	On Board Unit
OLOS	....	Obstructed Line Of Sight
QoS	....	Quality of Service
RSU	....	Roadside Unit
SAM	....	Service Announcement Message
SNA	....	Social network analysis
SPAT	....	Signal Phase And Timing Message
TCP/IP	....	Transmission Control Protocol/Internet Protocol
TMC	....	Traffic Message Channels
TraCI	....	Traffic Control Interface
VANET	....	Vehicular Ad-hoc NETwork
VSU	....	Vehicle Sensor Unit
V2I	....	Vehicle-to-Infrastructure
V2V	....	Vehicle-to-Vehicle
WAVE	....	Wireless Access in Vehicular Environment
WIMAX	....	Worldwide Interoperability for Microwave Access
WSN	....	Wireless Sensor Network

Chapter 1
Introduction

Next-generation telematics solutions are being driven by the maturation of recently deployed intelligent transportation systems, assisted by the integration of and rapid collaboration with information communication technology markets and the automotive industry. Inter-vehicle communication (IVC) has emerged as a promising field of research and development [1, 2, 3, 4, 5] , where advances in wireless and mobile ad-hoc networks can be applied to real-life problems (traffic jams, fuel consumption, pollutant emissions, and road accidents) and lead to a great market potential. Already, several major automobile manufacturers and research centers are investigating the development of IVC protocols, systems (e.g., DSRC, 802.11p) and the use of inter-vehicle communication for the establishment of Vehicular Ad-hoc NETworks (VANETs). Vehicles may utilize a variety of wireless technologies to communicate with other devices, but the dominant is Dedicated Short-Range Communication [6, 7] (DSRC), which is designed to support a variety of applications based on vehicular communication. Wireless Access in Vehicular Environment (WAVE) is a term used to describe the suite of IEEE P1609.x standards that are focused on MAC and network layers. WAVE is fairly complex and is built over the IEEE 802.11 standards by amending many tweaks to guarantee fast reliable exchange of safety messages; WAVE is the core part of DSRC. A vehicular network is a challenging environment since it combines a fixed infrastructure (roadside units, e.g., proxies), and ad hoc communications among vehicles. Vehicular networks have the diverse range of applications that varies from safety applications to comfort applications. Safety applications enhances the driving conditions and reduces the chances of accidents by providing enough time to the driver and applying the brakes automatically (eco-driving). These can be further divide into the following:

- Cooperative collision warning.
- Incident management.
- Emergency video streaming.

Intelligent transport applications aim at providing faster delivery of traffic information, and improving the efficiency and accuracy of traffic detection by allowing collaborative processing of information between vehicles. These applications focus on observing the traffic pattern and managing traffic accordingly (eco-routing). It can be further categorized into the following:

- Traffic monitoring.
- Traffic management.
- Platooning.
- Vehicle tracking.
- Notification services.

Comfort applications are the applications of VANET related to comfort level of the passenger moving in the vehicle. It can be further categorized into the following:

- Parking place management.
- Distributed games and/or talks.
- Peer-to-peer applications.

Consequently, the Quality-of-Service (QoS) required for the network varies from non-realtime, to soft real-time where a timing failure might compromise service quality, up to hard real-time where a timing failure might lead to a catastrophe. These applications can also be exemplified by their scope, i.e., whether they provide communication over a wide area, or are local only. Finally, such applications can vary in their networking approach: ad hoc, where vehicles communicate suddenly, or infrastructure-based, where communication is governed by fixed base stations. VANET has the communication type: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I).

In order to communicate with each other, vehicles exchange messages of different types. In the following a brief description of the individual message types is given:

- **Cooperative Awareness Message (CAM)**: Using CAMs, surrounding vehicles are informed about the host vehicles presence. The term host vehicle is used in this thesis to denote the reference vehicle. These messages are used to build up the Neighborhood Table and posses a generic structure. For each ITS station, i.e., basic vehicle, emergency vehicle, public transport vehicle, or RSU, an obligatory profile is chosen. This includes a list of tagged values like, e.g., vehicle type, dimensions, status of exterior lights and acceleration with respective confidence values. Depending on the situation also further safety parameters like crash status or emergency braking signals are transmitted. The list of optional tagged values is still under discussion and includes, e.g., the number of occupants or the current door status.

- **Decentralized Environment Notification Message (DENM):** DENMs are event based messages. Data elements for content type and subtype allow a fine grained addressing of respective applications. For DENMs, a destination area and location reference can be specified. The location reference refers to the geographical validity of the hazard and is defined as a circle, rectangle or as a trace, which consists of waypoints leading to the respective traffic situation.

- **Signal Phase And Timing Message (SPAT):** In order to enable optimized flow control on traffic light intersections, RSUs frequently broadcast the remaining green light phase for each lane. SPAT data may be further combined with messages describing the topology of the intersection for more precise referencing.

- **Service Announcement Message (SAM):** Services offered by RSUs are broadcasted via SAMs. The exact specification of SAMs is currently still an active work item of the ETSI standardization group. Note that these messages are not safety-related and consequently, will be sent on channels other than the three channels allocated in the G5A band.

1.1 Motivation

Vehicular Ad Hoc NETwork(VANET) applications are based on Car-to-X (C2X) communications. Vehicles become smarter with the installation of embedded systems and sensors. Sensors collect crucial data about the situation on the road and this information is exchanged in order to help the driver make appropriate decisions. The driver receives information about a local anomaly, a too short inter-distance with the leading vehicle, lane departure etc. Exchange of this information among neighboring vehicles is crucial for VANET applications to be efficient. Communication between vehicles can be used to inform drivers about congested roads ahead, a car accident, parking facilities etc. As a result, IVC communications may help drivers avoid dangerous situations, decrease driver time, fuel consumption and have an overall better driving satisfaction. Most of these applications demand data dissemination among vehicles.

Data dissemination generally refers to the process of spreading data or information over distributed wireless networks. From the networking point of view, it requires broadcast capabilities at the link layer, allowing a frame to be transmitted to all the vehicles in the radio scope. It also supposes implementation of network and transport mechanisms to disseminate the message in the whole network. This dissemination uses one of the two available communication modes. The message will be disseminated in a multi-hop fashion when the vehicle-to-vehicle (V2V) communication is enabled and will be broadcasted by all the roadside units (RSU) when infrastructure-to-vehicle (V2I-I2V) communications are

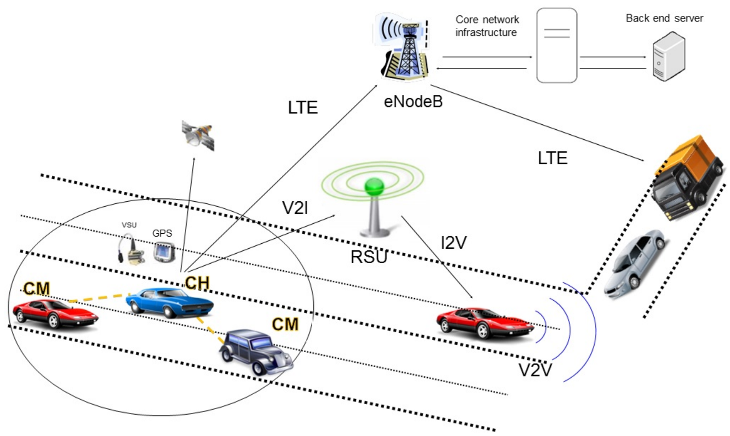


Figure 1.1: Data dissemination in VANETs

used instead. A hybrid version is also possible, RSUs broadcast the messages and, as they do not cover the whole network, some vehicles are selected to forward the message to complete the dissemination. These messages can be flooded at a certain number of hops or in a given area (geocasting) depending on the application purposes. In V2V mode, the tasks of a dissemination protocol consist in selecting a pertinent set of vehicles to disseminate the message, and defining retransmission procedures to ensure the entire applications requirements on reliability, delay, etc.[8]. Modern communication systems that exploit LTE * and DSRC communication capabilities are also investigated and developed mostly for safety and traffic management applications (Figure 1.1).

In IEEE 802.11p, the standard used to add wireless access in vehicular environments (WAVE), the sender does not know if its transmission has been received, and there is no retransmission in case of failure. However, safety and traffic management applications, require a reliable dissemination of the messages as they contain important information. Delivery delay is also an important factor. Messages must also be sent within the time specified by the application in order to be at any use. Therefore, protocols implemented at an upper layer is required to disseminate the message at several hops keeping the end to end delay low. These protocols must compensate the lack of reliability of the IEEE 802.11p and guarantee a fast and efficient delivery of the messages.

The basic forms of broadcasting in an ad-hoc network are:

- Flooding
- Probabilistic Broadcast

*http://uk.news.yahoo.com/google-audi-could-announce-android-car-next-week-092500470.html

- CounterBased Broadcast
- LocationBased Broadcast
- ClusterBased Broadcast
- Push-Based and Pull-Based Mechanism

There is a well-known problem in broadcasting in ad hoc networks, usually referred to as Broadcast Storm. This problem happens if we use a basic flooding also called blind flooding to disseminate a packet in the network. Basic flooding works as follows. When a node receives a packet which has to be disseminated in the network, it checks if it is the first reception of this packet. If yes, it broadcasts it; otherwise it silently discards it. Since each node forwards the packet, it leads to an important redundancy. This redundancy depends on the network density: a node will receive as many packets as it has neighbors in its radio range. A typical situation in a VANET when the broadcast storm problem occurs is a road accident (Figure 1.2). When an accident happens in a road segment at a point in time where traffic is intense the road is partially or totally blocked. In such a situation density of vehicles increases dramatically and a method is necessary for the proper dissemination of safety messages.

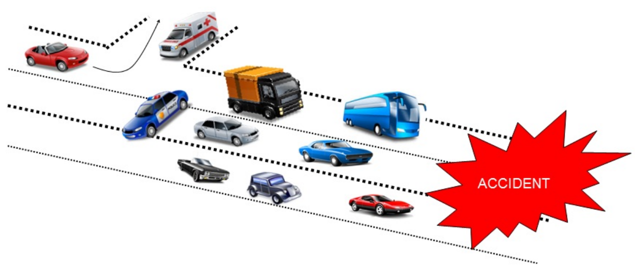


Figure 1.2: Broadcast storm problem may lead to safety-related service disruption in a road accident incident

In a VANET, a node may have up to 100 neighbors (the radio range of the IEEE 802.11p may reach up to 1 km and the density of vehicles may reach more than 100 vehicles per kilometer), such an approach will lead to 100 receptions for each vehicle. Such a scenario will significantly congest the network, causing packet transmissions to face heavy collisions, therefore wasting a lot of bandwidth and CPU resources[8].

Originating from the source node a message will be broadcasted through the relay nodes in order to reach the destination or cover the target area. This multi-hop broadcast procedure demands the appropriate selection of relay nodes, which are a subset of surrounding vehicles, in order to ensure that all vehicles in the target area are informed on time.

Since in VANETs ack/nack messages are not implemented, a certain level of redundancy, in terms of relay nodes, is required.

An appropriate protocol is required to ensure a good dissemination of the message. It is not possible to perform the optimal flooding, which minimizes the number of forwarders, because a complete and updated view of the topology is needed. This view requires a set of mechanisms not necessarily available: a link sensing mechanism allowing each node to discover its neighborhood, a link state routing protocol, etc. Moreover, even if such mechanisms are implemented in the VANET, it is not sure that routing information will be available for the dissemination protocol.As the dissemination requirements (reliability, delay, coverage, etc.) may be different from one application to another, the dissemination mechanism could be implemented at the application layer where the interaction with the routing layer is limited [9]

Detecting the most central nodes based on connection characteristics of the nodes is important for efficient data delivery in VANETs. The most important issue is to select a forwarding path with the smallest packet-delivery delay. Central nodes can create a backbone network which can be used for effective forwarding of messages. In order to address specific requirements of different applications, such as delay, area coverage etc knowledge of the topological characteristics of the VANET communication graph is required, where vehicles correspond to vertices and communication links to edges. Vehicles can be clustered into local groups based on these centrality metrics.

With the cluster-based scheme, nodes are supposed to be divided into a set of clusters. A cluster is a subset of vehicles forming a convex network. Clusters are supposed to be disjoint, i.e., a node can belong to only one cluster, while in some situations overlapping clusters are created where nodes that belong to more than one clusters are the gateway nodes. The basic idea is that of grouping network nodes that are in physical proximity, thereby providing the flat network a hierarchical organization, which is smaller in size, and simpler to manage. The subsequent backbone construction uses the induced hierarchy to form a communication infrastructure that is functional in providing desirable properties such as minimizing communication overhead, choosing data aggregation points, increasing the probability of aggregating redundant data, and so on. A lot of clustering protocols have been proposed for ad hoc networks and VANETs. Generally, a node in a cluster is classified as head, gateway, or member. The head, also called cluster head, is a particular node used to build the cluster. There is only one head for each cluster and it is often at the core of its cluster. Gateways are the nodes sharing a link with another cluster. Members are the nodes which are neither heads nor gateways. Most of the clustering schemes introduce additional messages which are translated into overhead. However, the clustering-specific messages are exchanged via the control channel (IEEE 802.11p) and this does not affect the dissemination of data.

Push-based data dissemination mechanisms on the other hand use fixed RSU or moving vehicles to periodically deliver data messages to other vehicles. These messages are managed by data centers which collect data from applications to deliver it to the vehicles. A computer with a wireless interface or an info-station can play the role of data center. This type of mechanism is useful for applications which need to advertise information to a set of vehicles. For example, it may be an application which delivers information about road and traffic conditions, estimated time to reach destinations, etc. Also, it may be interesting to advertise commercial information about restaurants, gas stations, etc.

Pull-based data dissemination mechanism is one form of request and response model. With this model, a vehicle sends query information to a specific location or target. For example, it can inquire about a gas station, parking lot, or any other service.

Packet routing mechanisms must be efficient in terms of delay, throughput and number of rebroadcasts. Routing of packets based on clustering of the network in smaller parts is crucial especially in situations where the density of vehicles is relatively high (traffic lights, peak hours, etc.). Proper scheduling/routing protocols must cope with the special characteristics of VANETs, i.e. Dynamic and dense network topology high mobility, frequent disconnected network, broadcast storm problem and hard delay constraints.

1.2 Synopsis

In this thesis we propose efficient methods for discovering central nodes, innovative clustering techniques, data dissemination and routing protocols that can be applied to VANETs. Particularly we investigate both centralized and distributed methods. We propose delay efficient routing methods that exploit partitioning of nodes by influencing the forwarding of packets. We also investigate how V2V communication can influence road congestion and overall CO2 reduction in Urban environments. An overview of the thesis is presented in Figure 1.2.

In Chapter 2 we present novel centrality metrics, *AWeNoR* and *AWeNoR–Reduced*, that characterize the network topology using only limited, local connectivity information one or two hop information. The objective of the proposed methods is to reward nodes not only according to how many neighbors they have but also according to the local rank they have in their communities. Due to this attribute no nodes (except from the isolated ones) remain unranked. These centrality metrics can be used as primitives in the design of networking protocols for both static and mobile networks.

In Chapter 3 we propose Distributed Clustering methods for Vehicular Networks. The initial method (*Spring clutering* [$Sp-Cl$]) forms stable clusters based on force directed algorithms. Using current and future possible position of vehicles, relative mobility and metrics such as direction(*Virtual Forces Vehicle Clustering* [$VFVC$]) and vehicles' height (*Enhanced Spring Clustering* [ESC]) the proposed distributed methods can create efficient clusters both in highways and urban environments.

In chapter 4 historical data of vehicles are used in order to further enhance the clustering technique. This data is collected from RSUs and are used in order to extract social patterns of vehicles/drivers. We developed a pair of algorithms, *Sociological Pattern Clustering (SPC)*, and *Route Stability Clustering (RSC)*, that exploit the social behavior of vehicles, i.e., their tendency to share the same/similar routes. The methods combine this macroscopic behavior with microscopic behavior in order to create stable and balanced clusters. A short term prediction mechanism that is based on two prediction mechanisms used simultaneously and complementary to each other in order to predict the next road segment that a vehicle will move, is also presented. Short term prediction mechanisms, can be used along with the the sociological patterns of vehicles in order to increase prediction accuracy and cluster stability.

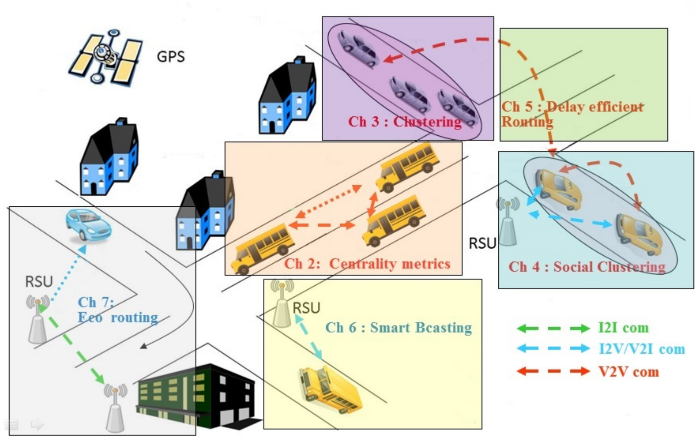


Figure 1.3: Thesis synopsis

Next, in Chapter 5 we propose Delay Efficient Backpressure Routing Protocols in Wireless Ad Hoc Networks. Both methods namely the Voting-based Backpressure ($\mathcal{V}o\mathcal{BP}$) and the Layered Backpressure ($\mathcal{L}ay\mathcal{BP}$) are throughput optimal. For the former method, taking a purely localized approach, we require the packets to carry their immediate travel history, so that the relays are prevented from sending the packets back from where they came. For the $\mathcal{L}ay\mathcal{BP}$ method, taking a less localized approach, we create layers of nodes and use the identities of these layers to forward the packets toward the destinations layer, and discouraging the packet from leaving the destination layer; these layers act as attractors for the packets. We also present Enhanced Layered Backpressure policy $\mathcal{EL}-\mathcal{BP}$ that improves the behavior of $\mathcal{L}ay\mathcal{BP}$ in case of moving nodes. The new protocol have the same characteristics with the $\mathcal{L}ay\mathcal{BP}$ and is robust to topology changes in terms of nodes that move from one layer to another.

In Chapter 6 we present a *Distributed Skip Air Index* (*DiSAIn*), for Smart Broadcasting in Intelligent Transportation Systems. In a push-based broadcast system, the RSUs (Road Side Units) can broadcast information concerning issues relevant to the moving vehicles. The presence of air indexes would give an answer about the time of broadcast thus helping the driver choose the correct driving behavior in order to retrieve this needed information. Such information is also useful for hybrid Geocast routing protocols, for use by dead drops and so on.

In Chapter 7 we present a Pull-based data dissemination mechanism for congestion avoidance in VANETs. We propose a real time system based on DSRC communication capabilities in order to reroute vehicles to the most ecological route, avoiding congested roads and minimizing the overall travel time and C02 emissions. The proposed *Ecological Routing of Vehicles* ($ErouVe$) method is a real time routing system that can be expanded from a city block to a whole city region.

Finally in Chapter 8, we study and summarize the main findings of our work. Additionally, we discuss possible future directions.

Chapter 2
Localized centrality metrics via path-based neighborhood analysis*

The synergy between social network analysis and wireless ad hoc network protocol design has recently created increased interest for developing methods and measures that capture the topological characteristics of a wireless network. Such techniques are used for the design of routing and multicasting protocols, for cooperative caching purposes, and so on. These techniques are mandatory to characterize the network topology using only limited, local connectivity information – one or two hop information. Even though it seems that such techniques can straightforwardly be derived from the respective network-wide techniques, their design presents significant challenges since they must capture rich information using limited knowledge. In this chapter we examine the issue of finding the most central nodes in neighborhoods of a given network with directed or undirected links taking into account only localized connectivity information. An algorithm that calculates the ranking, taking into account the N-hop neighborhood of each node is proposed. The method is compared to popular existing schemes for ranking, using Spearman's rank correlation coefficient. An extended, faster algorithm which reduces the size of the examined network is also described.

2.1 Introduction

During the last decade the advances in device miniaturization and in the respective system/application software, along with the tremendous growth of wireless networks have made the presence of ad hoc networks ubiquitous. A wealth of ad hoc networks is encountered today, such as mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), wireless mesh networks (WMNs) and so on. They have potential applications in disaster relief, battlefield environments, wireless Internet connectivity, intelligent vehicles. An ad hoc network consists of wireless hosts (nodes) that communicate with each other in the absence of a fixed infrastructure; each host acts as a relay that forwards messages towards their destination.

The lack of fixed infrastructure makes the nodes of an ad hoc network to be strongly interdependent on each other. This fact helped realize the significance of borrowing concepts from the field of social network analysis (SNA) [10] to the design of more efficient information

*The ideas presented in this chapter appear in the following publications [C.01,J.01] tranfer protocols. This borrowing was further enforced by the fact that many of the ad hoc networks are basically human-centered and they follow the way humans come into contact. Moreover, because of lack of infrastructure, it is rather challenging to develop more systematic design optimization approaches as for instance in cellular networks. Greedy, best-effort techniques are used primarily for opportunistic ad hoc networks and they may benefit significantly from the social networking perspective.

Informally, a *social network* is a collection of 'actors', a set of relational information on pairs of actors, and possible attributes of the actors and/or of the links. In our context, the actors are the ad hoc network nodes, and the relationship among pairs of actors is the existence (or not) of a wireless link among them. The attributes on the actors and links can model node energy and link quality metrics, respectively, are not investigated in the methods described in this chapter.

The notion of a social network and the methods of social network analysis (SNA) is a quite old discipline and they have attracted significant interest initially from the social and behavioral communities, later from the data mining [11, 12], and only recently from the networking community [13]. This interest stems from the focus of SNA to relationships among entities and on the patterns and implications of these relationships. In the networking community, SNA is viewed as another network measurement task, while the traditional tasks of network measurement deal with issues such as traffic monitoring, latency, bandwidth and congestion. The analysis of the 'social' aspects of a network is the study and exploitation of the structural information present in the network, such as existence and strength of communities [14], node centralities, network robustness to node removal, topology evolution over time [15], and so on.

Among the most significant tasks involved in SNA is the calculation of centrality measures [16]. Point centrality in communication is based upon the concept of betweenness, first introduced in [17]. According to betweenness centrality a node is central to the degree that it stands between others. PageRank [18] is another very popular method for measuring centricities in social networks, a spectral centrality measure; the basic idea behind PageRank is that a node is significant if it is connected to other significant nodes. Various other measures of centrality and ranking have been proposed to determine the importance of a vertex within a graph [19] (cf. Section 5.2).

These centrality indices are of great value in the understanding of the roles played by actors in social networks, and by the vertices in various webs (Web, Internet, Food/Sex web) but they are not appropriate to be used for protocol design in ad hoc networks. Network protocols for these types of wireless networks are based on localized algorithms, which means that they are allowed – for performance and scalability purposes – to use only local information, e.g., two or three hop connectivity information. Such 'localized' centrality metrics present a potential for control of communication, safety issues [20], routing protocols [21], information dissemination [22, 23] and so on.

Detecting the most central nodes based on social characteristics of the nodes is important for efficient data delivery in VANETs. The most important issue is to select a forwarding path with the smallest packet-delivery delay. Although geographical forwarding approaches such as greedy perimeter stateless routing (GPSR) [24], which always chooses the next hop closer to the destination, are very efficient for data delivery in ad hoc networks, they may not be suitable for sparsely connected vehicular networks.

When social contact patterns are exploited, data forwarding is divided into two stages. First, node centrality is evaluated at the global scope which includes all the nodes in the network, to ensure that data is carried and forwarded by relays with higher capability of contacting other nodes. Second, when a relay contacts a node within the same community of the destination, data is forwarded to that community. Afterward, node centrality is evaluated within the local community scope, and data is forwarded directly to the destination. In both stages, most of the current social-aware data forwarding schemes evaluate the centrality of nodes [25].

We study the problem of identifying the most central nodes in networks (graphs) by using only localized information, i.e., of a few hops. It is motivated by the design of protocols in wireless networks that seek for nodes "central" in the network to assign to them special roles, e.g., *mediator* nodes in cooperative caching for sensor networks [22, 26], *message ferrying* nodes in Delay Tolerant Networks [21], *rebroadcasting* nodes in vehicular networks [27], and so on. The relation of social networks and ad hoc networks is a well established relation in many works, e.g., [28] and the references therein. In this context, the work presented in this chapter makes the following contributions:

- argues for the inadequacy of network-wide centrality measures for use in ad hoc network protocol design, and explains the importance and challenges of designing *localized* centrality measures, initiating the relevant research.

- proposes two measures, namely *AWeNoR* and *AWeNoR–Reduced* that can be used for ad hoc network protocol design.

- evaluates the qualitative characteristics of these two measures comparing them with two popular network-wide centality measures, namely shortest-path betweenness centrality and PageRank centrality, using two real, well-known in the literature social networks and a medium graph of the network.

The remainder of this chapter is structured as follows: Section 2.2 briefly describes the related work on centrality measures. In Section 2.3, the network model, the assumptions and the *AWeNoR* ranking technique are described. Section 2.4 shows the results of the comparison of *AWeNoR* to other centrality measures. Section 2.5 introduces another faster technique for computing final rankings through local weight, and the chapter concludes with Section 2.6.

2.2 Relevant work

There are really no true ‘localized’ centrality measures, except from the degree centrality [10], which is loosely defined as the number of 1-hop neighbors of a node, and its variations, the lobby index [29] and the power community index [26]. Though, these indices are poor indicators of the local connectivity. The rest of the centrality measures are computed using knowledge of the complete (network-wide) connectivity information. Closeness centrality [10] is defined as the inverse of the sum of the distances between a given node and all other nodes in the network. We easily realize that this index is practically meaningless in a narrow, e.g., a two-hop neighborhood.

Shortest-path betweenness centrality [10] is defined as the fraction of the shortest paths between any pair of nodes that pass through a node. A similar technique that measures the extent of bridging capability of all nodes or links in the network is the bridging centrality [12, 30] . These centrality measures are relatively rich indicators of node ‘positioning’, but when these measures are to be used in ad hoc wireless networks, they suffer from several shortcomings. Betweenness centrality suffers from the fact that it leaves many nodes unranked, when these nodes don’t participate in any shortest paths computed. Moreover, the existence of bridge links in the network graph, result in increasing at an excessive amount the centrality value of the *articulation* node without this node being really “central”.

Other similar centrality measures are flow-betweenness [31] and betweenness centrality based on random walks [32]. Flow betweenness of a node i is defined as the amount of flow through node i when the maximum flow is transmitted from node s to node t, averaged over all s and t. The method requires the computation of all the maximum flows in the network and suffers from some of the same drawbacks as shortest-path betweenness. The random walk betweenness centrality on the other hand for a node i is the number of times a message passes through i on its journey, averaged over a large number of trials of the random walk and all pairs s, t. This measure is not localized and demands $O((L + V)V^2)$ in computational time.

Apart from the aforemetioned graph-theoretic measures, a very popular family of centralities are the spectral centralities [33]. There are various definitions of such measures, which are referred to as ‘spectral’ because they are based on the spectral properties of the matrix which represents the relationships among the nodes. These measures define the prominence of a node recursively, i.e., a node is prominent if it is pointed to by other prominent nodes. The most popular of the spectral centrality measures is the PageRank metric [18], which is one of the methods used by Google to rank Web pages. PageRank suffers from the fact that nodes may be ranked very high due to the fact that they are adjacent to significant nodes even though they play no specific role in packet forwarding (e.g., the *sink* nodes). Additionally, the PageRank produces meaningfull rankings when applied only to relatively large graphs and not in narrow, e.g., two-hop neighborhoods. Moreover, the computation of PageRank requires cumbersome calculations and knowledge of the whole network topology, which is not possible in ad hoc wireless networks that require *localized* algorithms.

In this chapter a novel measure for calculating the centrality of nodes in networks (static or semistatic) is proposed. The basic idea is that the centrality of a node is to be calculated over its neighborhood. In this subgraph, all the paths connecting the considered node with all the nodes of the neighborhood are found and a local weight is computed. Local weights are accumulated to give an aggregated measure of centrality and subsequently a node ranking. The new measure called *"Aggregated Weight N-hop Ranking"* (*AWeNoR*) not only rewards nodes that belong to many neighborhoods, but also, it rewards those ranked high in the neighborhoods they belong to. Due to this attribute no nodes (except from the isolated ones) remain unranked. To remedy the computational complexity of this measure, we also describe a second measure, namely *AWeNoR–Reduced*.

2.3 The *AWeNoR* node ranking method

As described earlier, the basic idea behind the proposed method is to create each node's neighborhood and compute the local weights in this subgraph. All these weights are then accumulated in order to give the final rank of each node. In subsection 2.3.1 we describe the algorithm for this method, subsection 2.3.2 shows how local weights are calculated, and subsection 2.3.3 demonstrates how the final rankings are computed by aggregating the local weights.

2.3.1 The N-hop neighborhood

We consider a network $G = (V, L)$, where V is the set of nodes and L is the set of links. Each link can be undirected or directed having weight equal to 1.[†] Each node is given a distinct ID; IDs start from the value one.

[†] Other weights can be assigned as well, when we want to model energy, latency issues, but these issues are not examined here.

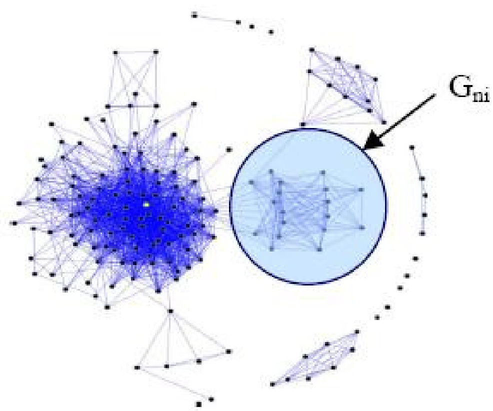


Figure 2.1: Graph G and a neighborhood $G_{N,i}$.

Definition 1. *A node j belongs to neighborhood $G_{N,i}$ of the node i, if there exists at least one path from the starting vertex i to the end vertex j, in at most N-hops away.*

In order to compute the ranking of each node, the proposed method operates as follows:

1. Find all the paths from node i to every other node j which is at most N hops away, thus creating the neighborhood $G_{N,i}$.

2. Calculate the local weight of all the nodes in $G_{N,i}$ (except from i) according to the *AWeNoR* method [explained later].

3. Accumulate local weights to obtain the final ranking of all the nodes.

Since ad hoc networks are relative sparse, the space requirements are not really large; with an average node degree equal to d, each node i needs to construct a $d^N \times d^N$ table (both d and N are not expected to be larger than 10).

2.3.2 Local weight

The *AWeNoR* algorithm aims at computing the local weights of nodes which belong to $G_{N,i}$ neighborhoods. There are two intuitions behind this algorithm. Firstly, the nodes closer to the starting node of a path are more crucial than the more distant ones, with respect to disseminating information to the rest of the network. Secondly, all paths can be used to pass data in a neighborhood and not only the shortest path, as used by the betweenness centrality when it calculates node rankings. The algorithm for computing local weight proceeds by deriving all paths with starting node i. The paths are specified as $P_k^i = \left(u_i^0, u_i^1, ..., u_i^N\right)$, where P_k^i is the k^{th} path from start node i, and u_i^j is a node at a j-hop distance from the start node i. For each hop j, a weight is computed for each node l using Equation 2.1.

$$W_l^j = \frac{a_l^j}{K_l}, \qquad K_l > 0 \tag{2.1}$$

where K_l is the total number of nodes that appear at the j-th hop and a_l^j shows the number of times that node l appears in hop j throught all the paths of the neighborhood $G_{N,i}$. The local weight for any node in neighborhood $G_{N,i}$ is computed using Equation 2.2.

$$b_l^i = \sum_{\forall j} \frac{W_l^j}{j}, \qquad \forall l \in G_{N,i} \tag{2.2}$$

The size of the neighborhood is a parameter that plays a significant role. Taking N equal to the network diameter, the neighborhoods coincide with the network graph G. In that case, in order to compute the ranking of a node, all paths between nodes have to be found, thus making the algorithm inappropriate even for medium-sized networks. On the other hand, giving to N a very small value, the obtained rankings may not be very representative at all.

As an example, consider the directed network shown in Figure 2.2 where a neighborhood $G_{N,i}$ of a directed graph is shown. A directed graph is used in order to be more clear for the reader to understand the steps of the method. Parameter N which is the size of the neighborhood has value 4 while the the initial node is node with id 1. Thus the neighborhood is $G_{4,1}$. The paths that exist in this neighborhood are shown at Table 2.1.

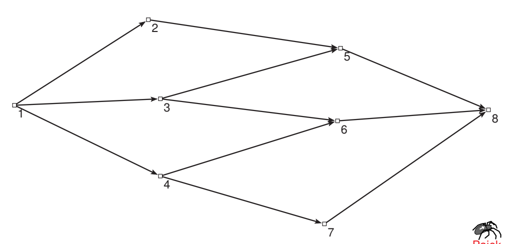


Figure 2.2: Example neighborhood $G_{N,i}$.

Values of parameter a_l^j are shown at Table 2.2 for every node except node 1 since it is the starting node of all paths and the value a_l^j for that node would be equal to one.

After computing parameter a_l^j for all nodes of the neighborhood, Equation 2.1 is used. Parameter's K_j value for every hop is 5,5,5,1 respectively. In the fourth hop only one path has a node thus K_4 equals one. It can be verified that $W_2^1 = 1/5, W_3^1 = 2/5, W_4^1 =$

intial node	1_{st} *hop*	2_{nd} *hop*	3_{rd} *hop*	4_{rth} *hop*
1	2	5	8	-
1	3	6	8	-
1	3	6	8	-
1	4	6	8	-
1	4	7	6	8

Table 2.1: Neighborhood's paths.

1_{st} *hop*	2_{nd} *hop*	3_{rd} *hop*	4_{rth} *hop*
$j \mid a_l^j$	$j \mid a_l^j$	$j \mid a_l^j$	$j \mid a_l^j$
2 | 1	5 | 2	6 | 1	8 | 1
3 | 2	6 | 2	8 | 4	
4 | 2	7 | 1		

Table 2.2: Parameter a_l^j.

$2/5, W_5^2 = 2/5, W_6^1 = 2/5, W_7^2 = 1/5, W_6^3 = 1/5, W_8^3 = 4/5, W_8^4 = 1$.

Final local weights of all nodes of $G_{4,1}$ according to Equation 2.2 are $b_2^1 = \frac{1}{5}, b_3^1 = \frac{2}{5}, b_4^1 = \frac{2}{5}, b_5^1 = \frac{2}{5} * \frac{1}{2}, b_6^1 = \frac{2}{5} * \frac{1}{2} + \frac{1}{5} * \frac{1}{3}, b_7^1 = \frac{1}{5} * \frac{1}{2}, b_8^1 = \frac{4}{5} * \frac{1}{3} + \frac{1}{1} * \frac{1}{4}$.

2.3.3 Final rankings

The algorithm *AWeNoR* computes local weights for all nodes that belong to a neighborhood $G_{N,i}$. Since nodes may belong to multiple neighborhoods, the local weights have to be accumulated in order to obtain the final ranking of the node using Equation 2.3.

$$b_l = \sum_{\forall G_{N,i}} b_l^i, \qquad \forall l \in G \tag{2.3}$$

It must be stated that only acyclic paths are used from *AWeNoR* in order to compute local weights. Also, in every neighborhood $G_{N,i}$, the local weights are calculated for every node that belongs to $G_{N,i}$, except from i itself, since its weight, using Equation 2.1, would be equal to one.

The time complexity of the method is $O((|L| + |V|)|V|)$, since every node and every link will be explored in the worst case, for each neighborhood created. Parameter $|L|$ is the cardinality of the set of links (the number of links), and $|V|$ is the cardinality of the set of nodes. This is the total time consumed if the computation of local weights is performed in a sequential manner in all nodes. *AWeNoR* can be conducted either centrally or indepedently at every node. In the latter case time complexity of the method is $O((|L| + |V|))$.

2.4 Evaluation of the proposed method

In order to evaluate the proposed ranking technique, since there is no "ground truth", we used two real networks. These graphs represent real social networks, with large connectivity among nodes. We used networks with both undirected and directed links. The visualization of the networks was performed with Pajek (http://vlado.fmf.unilj.si/pub/networks/pajek/) and the calculation of the shortest-path betweenness and PageRank centrality values of the network nodes was done with the aid of CentiBiN (http://centibin.ipk-gatersleben.de/) package.

The real graphs are the following:

- Zachary's karate club: a network of friendships between 34 members of a karate club at a US university in the 1970 [34].
- Dolphin network: an undirected network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [35].

We also examined the performance of the method in a medium network of routers of the internet called Autonomous Systems (AS).The network we used consists of 103 nodes. (http://snap.stanford.edu/data/as.html).

Except from the *AWeNoR* centrality metric, the betweenness and the PageRank centrality values were also computed for every graph in order to compare them. For every graph ranking, we measure the number of ties that each ranking algorithm produces and also we compute the Spearman's rank correlation coefficient (Equation 2.4) between pairs of ranking algorithms. The more ties an algorithm produces, the less useful the ranking is for use in wireless networks, because it does not discriminate among network nodes. This is also a crucial requirement for Web rankings when they are going to be used in search engines. Spearman's is a non-parametric measure of correlation widely used to describe the relationship between two variables that is used to report the difference in ranking produced by two methods. Differences $d_i = \mid x_i - y_i \mid$ between the ranks of each observation on the two variables are calculated. In our case, this metric is used to evaluate the proposed metric in relation to PageRank and shortest-path betweenness centrality values.

$$p = 1 - \frac{6\sum d_i^2}{n(n^2-1)}, \qquad \forall i \in G \tag{2.4}$$

For all the networks used to evaluate our method the average distance D is computed and this parameter is used to create neighborhoods (Equation 2.5).

$$N = \lceil D \rceil \tag{2.5}$$

2.4.1 Undirected experimental graphs

The first real graph, the Zachary's karate club is shown in Figure 2.3 and the Dolphins graph is depicted in Figure 2.4. The visualization is used here as a means to confirm the obtained results via human intuition.

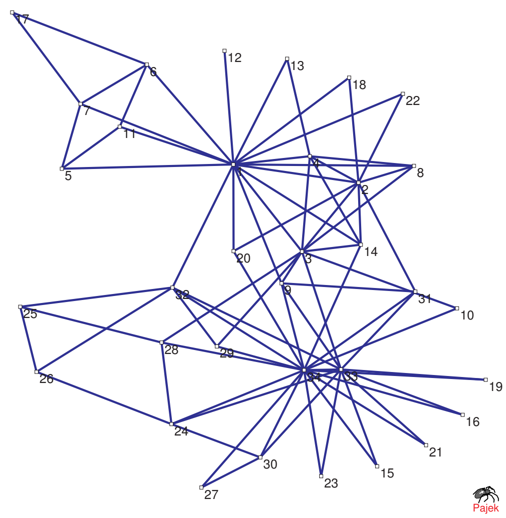


Figure 2.3: Zachary's karate club undirected graph.

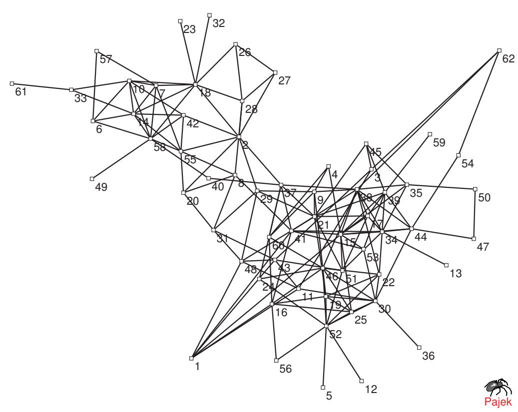


Figure 2.4: The Dolphins network.

Table 2.3 shows the total number of ties that each of the three methods produces for the two networks. The numbers in parentheses represent the number of nodes with zero centrality value (non-ranked). It can be seen that the betweenness centrality measure produces a significant amount of non-ranked nodes, which is a non desirable effect when the centrality measures is used in wireless networks for characterizing the significance of nodes in the network topology.

graphs	*Betweenness*	*Pagerank*	*AWeNoR*
Zachary's karate club	16(12)	11(0)	11(0)
Dolphin social network	9(9)	4(0)	4(0)
Autonomous system	45(43)	31(0)	30(0)

Table 2.3: The number of ties produced by each competitor.

Table 2.4 shows the Spearman's rank correlation coefficient computed for every pair of rankings. In the Dolphins dataset, we can observe significant discrepancy in the rankings produced by *AWeNoR* with those produced by PageRank.

graphs	*Betweenness - AWeNoR*	*Pagerank - AWeNoR*	*Betweenness - Pagerank*
Zachary's karate club	0,8442	0,8512	0,8747
Dolphin social network	0,7712	0,9457	0,8171
Autonomous system	0,7196	0,8712	0,8566

Table 2.4: Spearman's rank correlation coefficient.

Table 2.5 shows the biggest rank difference observed between the three methods. For the Dolphin and the Autonomus networks, where the number of nodes is relatively large, it is observed that the *AWeNoR* gives results close to PageRank. The numbers in parentheses represent the node where the biggest difference is observed. Table 2.6 depicts the highest ranked nodes for each graph. We observe that *AWeNoR* makes similar to PageRank rankings for the top-ranked nodes, even though it is a localized measure whereas PageRank requires cumbersome computations and knowledge of the whole network's topology.

graphs	*Betweenness - AWeNoR*	*Pagerank - AWeNoR*	*Betweenness - Pagerank*
Zachary's karate club	14(26)	12(10)	13(10)
Dolphin social network	38(40)	13(17)	41(40)
Autonomous system	56(67)	62(19)	50(67)

Table 2.5: Biggest difference observed.

In order to evaluate the proposed method against the competing ones for the Autonomus network, we focus on the biggest differences observed since the graph is too large to be displayed. In the comparison of *AWeNoR* and Pagerank we see that node with ID 19

RANK POSITION	*Pagerank*	*AWeNoR*	*AWeNoR Reduced*	*Betweenness*
1^{st}	34	34	34	1
2^{nd}	1	1	3	34
3^{rd}	33	33	33	33
4^{th}	3	3	2	2
5^{th}	2	2	1	32
RANK POSITION	*Pagerank*	*AWeNoR*	*AWeNoR Reduced*	*Betweenness*
1^{st}	15	15	5	37
2^{nd}	18	38	58	2
3^{rd}	52	46	18	41
4^{th}	58	34	34	38
5^{th}	38	52	44	8
RANK POSITION	*Pagerank*	*AWeNoR*	*AWeNoR Reduced*	*Betweenness*
1^{st}	60	60	60	60
2^{nd}	11	11	11	11
3^{rd}	40	40	40	40
4^{th}	16	16	31	16
5^{th}	1	1	16	31

Table 2.6: Highest ranked nodes for Karate (top), Dolphins(middle) and Autonomous(bottom).

is ranked in position 95 by our method while Pagerank puts it in place 33. Observing the Figure 2.5 we see that node 19 is rather isolated in the graph but is two hops away from node 3 which is ranked high in both methods (Pagerank : 16_{th}, $AWeNoR$ 17_{th}). Pagerank tends to reward such nodes with high score even though their significance in data dissemination is rather contradictory. Betweeness centrality on the other hand leaves too many nodes unranked (in the autonomus network 43 out of 103 nodes are unranked) making it rather impractical.

2.4.2 Directed experimental graphs

The new ranking technique was also tested in directed graphs. Taking the Karate club real graph and converting each edge to a directed arc, the network of Figure 2.6 is created. The direction of each arc is selected randomly.

Table 2.7 shows that $AWeNoR$ incurs significantly fewer ties than PageRank does. (Betweenness centrality is not possible to be computed for this network due to the lack of strong connectivity.)

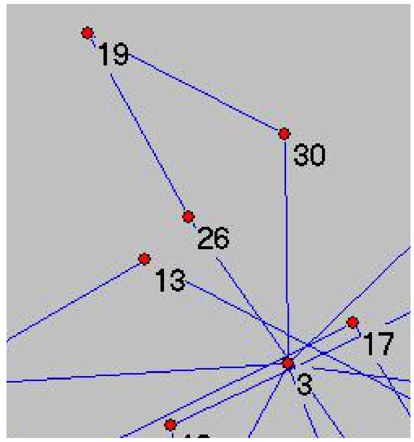


Figure 2.5: The Dolphins network.

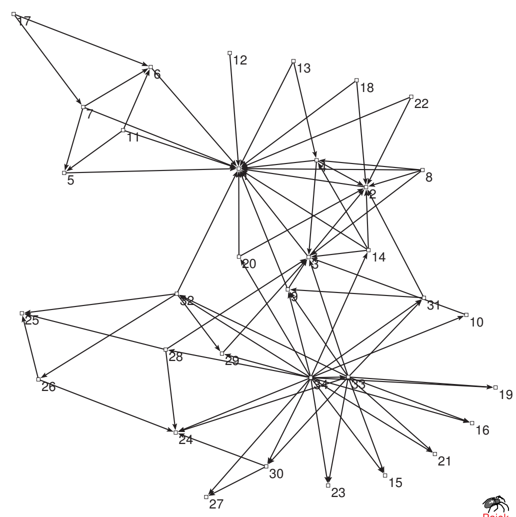


Figure 2.6: Zachary's karate club directed graph.

graphs	*Pagerank*	*AWENOR*
$Zachary's\, karate\, club$	21	15
$Dolphin\, social\, network$	23	22
$Autonomous\, system$	70	56

Table 2.7: Number of ties incurred by each algorithm.

Table 2.8 depicts the ids of the five highest ranked (top-5) nodes for this directed network. The numbers in parentheses represent the position of the node in the ranking produced by the competitor method, in the cases where this node does not appear in the top-5 list of the competitor. From Figure 2.6 and Table 2.8, we gain an insight why the *AWeNoR* algorithm is a more accurate algorithm in determining the most significant nodes compared to PageRank in terms of data dissemination: for instance, node 33 (ranked 4th) is more crucial in terms of routing than node 19, which is a sink node. The three highest ranked nodes are the same for both methods in the Karate club graph. Of course, such an observation is not a proof of the superiority of the algorithms, but it is a strong evidence that produces more meaningful rankings for the considered application scenaria.

RANK POSITION	*Pagerank*	*AWeNoR*
1^{st}	1	1
2^{nd}	3	3
3^{rd}	2	2
4^{th}	$19(24^{th})$	$33(7^{th})$
5^{th}	$4(6^{th})$	$9(8^{th})$
RANK POSITION	*Pagerank*	*AWeNoR*
1^{st}	1	1
2^{nd}	15	15
3^{rd}	$16(4^{th})$	$38(23^{th})$
4^{th}	$4(6^{th})$	$16(3^{RD})$
5^{th}	$19(8^{th})$	$46(33^{th})$
RANK POSITION	*Pagerank*	*AWeNoR*
1^{st}	1	60
2^{nd}	11	11
3^{rd}	60	9
4^{th}	9	1
5^{th}	$5(8^{th})$	$16(6^{th})$

Table 2.8: Highest ranked nodes for Karate (top), Dolphins(middle) and Autonomous(bottom).

In the Dolphins network (Figure 2.4), the nodes with id 38 and 46, which are among the highest ranked by *AWeNoR*, have very low ranking position in PageRank measure. This is due to the fact that *AWeNoR* ranking rewards nodes that belong to many neighborhoods, though PageRank only those connected to significant nodes. PageRank may rank in high position those nodes that have few (even just one) neighbor, which is significant to the network, without examining if they play any role in larger neighborhoods, which is desirable by policies applied to ad hoc wireless networks. In the Autonomous network due to the connectivity of the graph the two methods give similar results. The network could not be displayed due to space limitations (the graph is too large to be able to distinguish node's ids) but the reader can download the connectivity matrix from (http://snap.stanford.edu/data/as.html) and use pajek or any similar program to obtain a visual representation.

2.5 The *AWeNoR–Reduced* centrality measure

As described in Section 2.4, in order to compute the aggregated weights, the *AWeNoR* algorithm has to add local weights of all neighborhoods in the network. So, for a K-hop 'long' network, the *AWeNoR* algorithm has to run K times, one for each node. Computing local weights for every neighborhood can be a very time consuming task even for medium sized networks. In order to improve the total running time of the proposed *AWeNoR* algorithm, we further describe here the *AWeNoR–Reduced* ranking method. The *AWeNoR–Reduced* algorithm creates neighborhoods only for some nodes, according to a parameter q_i and a threshold A. Parameter q_i is used to count the times that node i participates in paths of all the neighborhoods created by the algorithm in every step. The *AWeNoR–Reduced* runs only cenrtally or with an excessive change of information between nodes, in contrast to *AWeNoR* that can be executed indepedently at every node. At the first itteration of the algorithm a node i (node with $id = 1$ is chosen) creates its neighborhood $G_{N,i}$ by detecting all paths of lenght N. Every node j that participates in any path of the node update the parameter q_j (q_j++) for every instance. At every next iterration of the algorithm another node is selected randomly and if its parameter q_i is below threshold A the procedure follows the same steps. If q_i is over A which means that node i has already participated in many other neighborhoods the node i is discarted and algorithm moves to next selected node.

The *AWeNoR–Reduced* ranking algorithm is described below in pseudocode:

1. Initiate algorithm. Set $i = 1$.
2. If $q_i < A$ then find all the paths from node i to every node j which are at most N hops away, thus creating the neighborhood $G_{N,i}$.
3. For every path $P_k^i = \left(u_i^0, u_i^1, ..., u_i^N\right)$ update parameter $q_{ui} \forall u \in P_i^K$ except u_i^N and u_i^0 .
4. Calculate the local weight of all the nodes in $G_{N,i}$ (except from node i) according to the *AWeNoR* algorithm.
5. Set $i = i + 1$. If the last node of graph is reached, then go to step 6 else go to step 2.
6. Accumulate local weights to obtain the final ranking of all the nodes.

The *AWeNoR–Reduced* algorithm, according to Table 2.6 and Table 2.9, achieves the same performance in terms of finding the most important nodes in a graph, while requires fewer neighborhoods to be created (Table 2.10).

undirected graphs	*Betweenness-AWeNoR Reduced*	*Pagerank-AWeNoR Reduced*	*AWeNoR-AWeNoR Reduced*
Zachary's karate club	0,8105	0,8438	0,9175
Dolphin social network	0,7925	0,9207	0,8782
Autonomous system	0,7296	0,8517	0,9728

Table 2.9: Spearman's rank correlation coefficient.

The parameter A is used as a threshold in order to choose whether a node's neighborhood is created or not. Choosing the value of parameter A is an important issue. Giving A a rather big value, the *AWeNoR–Reduced* algorithm degenerates to *AWeNoR*, since all neighborhoods are created. Setting A equal to zero, a risk of creating disjoint neighborhoods arises, letting some nodes unranked. In the experiments conducted, a value close to zero was used in order to avoid these situations.

undirected graphs	*AWeNoR Reduced (A=1)*	*AWeNoR Reduced (A=3)*	*AWeNoR*
Zachary's karate club	3	7	34
Dolphin social network	14	17	62
Autonomous system	31	38	103

Table 2.10: Neighborhoods created.

Figure 2.7 shows the effect of parameter A to method's results compared to *AWeNoR*, along with the number of neighborhoods created for every such choice. A strict relation between method's accuracy and cost, in terms of time consumption, is observed. The preferred policy is to have a value that changes according to the size or connectivity of the network, but its development is a subject of future work.

2.6 Chapter Conclusions

The issue of discovering which nodes in a wireless ad hoc network are central to the topology is of fundamental importance, since it can be used as a primitive to perform routing [21, 27], cooperative caching [22] and contamination detection.

There exist several centrality measures in the literature, like shortest-path betweenness centrality, PageRank, closeness centrality. Betweenness is based on shortest paths between nodes. nodes that lie on many shortest paths between other nodes are given a high centrality value. In many cases though, this measure is not useful due to the fact that it counts only a small subset of all the paths. Moreover, it creates hotspots in the

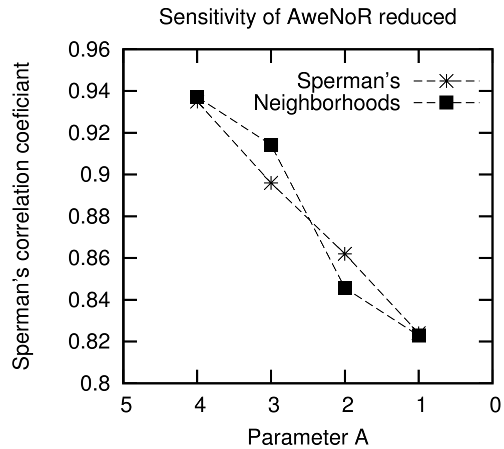


Figure 2.7: Sensitivity of AweNoR reduced ranking to parameter A (Zachary's karate club undirected graph)

communications because it consistently uses very few nodes [36]. When PageRank is used, the significance of a node comes from the significance of its 1-hop neighborhoods, leading many times to misleading results. A sink node may be ranked very high just because it is adjacent to a very significant node, even though its contribution to communication is of no importance. Additionally, these measures need to take into consideration the whole network topology – they are "centralized", which is not acceptable when these centrality measures are to be used for ad hoc network protocol design.

We propose a new measure, namely *AWeNoR*, for determining significant nodes. For each node i a neighborhood is created, and all paths with starting node i are created. For every "cluster" created, a local weight is computed, and a final ranking measure is created by adding these local weights. The new *localized* centrality measure rewards nodes that belong to many neighborhoods, and lie in many paths between nodes of the neighborhood. This measure was compared to shortest-path betweenness and PageRank centrality, and achieved to provide meaningful rankings with few ties, and leave no nodes unranked for both directed and undirected networks. The *AWeNoR–Reduced*, a faster algorithm for finding localized centrality values, was also presented.

These centrality measures can be used as a primitive in the design of networking protocols, like cooperative caching for ad hoc wireless networks [26], and routing in DTN networks where other attributes like energy of nodes or link quality could be combined. In VANETs special characteristics of vehicles like position, direction, relative mobility and social behavior of drivers can be incorporated in the centrality metric in order to better represent significance of actors in real life.

Chapter 3
Force Directed Distributed Clustering*

Clustering in VANETs is of crucial importance in order to cope with the dynamic features of the vehicular topologies. Algorithms that give good results in Manets fail to create stable clusters since vehicular nodes are characterized by their high mobility and the different mobility patterns that even nodes in proximity may follow. We propose a distributed clustering algorithm which forms stable clusters based on force directed algorithms. The simulation results show that our Spring-Clustering (Sp-Cl) scheme has stable performance in randomly generated scenarios on a highway. It forms lesser clusters than Lowest-ID and it is better in terms of Cluster stability compared to Lowest-ID and LPG algorithms in the same scenarios.

3.1 Introduction

For exchanging information about the current driving situation - traffic or weather conditions, hazard areas or road conditions - vehicles form a spontaneous network, known as a vehicular ad hoc network (VANET), even though the aid of fixed infrastructure [37] can be used. Due to the distributed network nature many messages are generated describing the same hazard event, hence, these messages can be combined to a single aggregate message through clustering. Since VANETs have a very limited capacity, it is desired that the number of messages can be reduced e.g. using aggregation. To reduce the number of aggregators, single messages are not broadcasted through the whole network, however, they are contained in a given area around the hazard event location. Only vehicles inside this area receive single messages and aggregate them. The vehicles outside this area are informed about the hazard event by the aggregate messages only. To further reduce the number of messages in a network, aggregate messages can be aggregated also.

In order to perform aggregation, several clustering techniques have been introduced, while other clustering algorithms for MANETS are also used (Figure 3.1). Cluster leaders also called clusterhead are assigned special operations like regulation of channel use, data aggregation and message routing between cluster members and between clusters.

Exchange of information between vehicles can be either V2V or vehicle-to-roadside (V2R). Forming V2V-based VANETs has some advantages as compared with the V2R-based VANETs. First, the V2V-based VANET is more flexible and independent of the roadside

*The ideas presented in this chapter appear in the following publications [C.04,C.05,C.06]

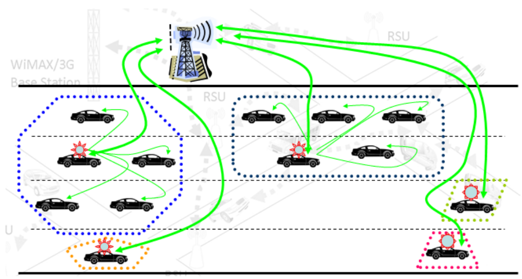


Figure 3.1: Vehicle clustering.

conditions, which is particularly attractive for the most developing countries or remote rural areas where the roadside infrastructures are not necessarily available. Also, V2V-based VANET can avoid the fast fading, short connectivity time, high frequent hand-offs, and so forth caused by the high relative-speed difference between the fast-moving vehicles and the stationary base stations. However, the link qualities in V2V communications can also be very bad due to multi path fading, shadowing, and Doppler shifts caused by the high mobility of vehicles. V2V communication can be used as the basic means of communication between vehicles and Roadside units may help in places of high vehicle density.

In our clustering scheme only V2V communication between vehicles is considered. The combination of V2V with Roadside units in urban areas with high traffic, where RSUs take control of nearby clusters acting as clusterhead is a matter of future enhancement for our Spring-clustering scheme.

3.2 VANET clustering algorithms

One of the many challenges for VANETs is the dynamic and dense network topology, resulting from the high mobility and high node-density of vehicles [38] especially in urban environments. This dynamic topology causes routing difficulties. A clustered structure can make the network appear smaller and more stable in the view of each vehicle.

A well-known mobility-based clustering technique is Mobility Based Clustering (MOBIC) [39], which is an extension of the Lowest-ID algorithm [40]. In Lowest-ID, each node is assigned a unique ID, and the node with the lowest ID in its two-hop neighborhood is elected to be the cluster head. This scheme favors nodes with lower identifiers to become clusterheads (CHs) without taking in mind mobility patterns of the nodes.

In MOBIC, an aggregate local mobility metric is the basis for cluster formation instead of node ID. The node with the smallest variance of relative mobility to its neighbors is elected as the cluster head. The relative mobility for a certain node is estimated by comparing the received power of two consecutive messages from each neighboring node which is not a so easy task in high dense environments. If we consider certain scenarios where attenuation of radio signals inevitably exists, the power of signals for estimating mobility can be very limiting and may provide inaccurate measurements. In cluster maintenance also a clusterhead is not guaranteed to bear a low mobility characteristic relative to its members. As time advances the mobility criterion between cluster members is somewhat ignored. If mobile nodes move randomly and change their speeds from time to time, the performance of MOBIC may be greatly degraded.

Many clustering methods have been introduced lately which aim at establishing stable clusters, where clusterhead reelection is reduced. In Dynamic Doppler Value Clustering (DDVC) [41] the Doppler shift of communication signals is used in order to create clusters. Affinity propagation is an algorithm for image processing, and APROVE has proved that its distributed case can be utilized for VANETs [42]. In Distributed group mobility adaptive clustering (DGMA) [43] group mobility information which contains group physical center's coordinates, group size, group velocity is used for clustering. Density based clustering is based on a complex clustering metric which takes into account the density of the connection graph, the link quality and the road traffic conditions [44]. Blum et al. [45] proposed a Clustering for Open IVC Networks (COIN) algorithm where cluster-head election is based on vehicular dynamics and driver intentions; Zhang in [46] proposed a DSRC multi-channel-based clustering scheme.

Finally, a large number of sensor node clustering algorithms have been proposed in the literature, e.g., [47] and the references therein, but these are inappropriate for our environment since they assume stationary nodes.

3.3 System overview and assumptions

The idea is based on force-directed algorithms. The force-directed assign forces among the set of edges and the set of nodes in a network. The most straightforward method is to assign forces as if the edges were springs and the nodes were electrically charged particles. The entire graph is then simulated as if it were a physical system. The forces are applied to the nodes, pulling them closer together or pushing them further apart.

Every node applies to its neighbors a force F_{rel} according to their distance and their velocities. Vehicles that move to the same direction or towards each other apply positive forces while vehicles moving away apply negative forces. Components of the vector F_{rel} along the east-west F_x and north-south F_y axes are calculated. In order to form stable clusters only vehicles that move to the same direction or towards each other are considered as candidate cluster members.

For a specific vehicle that the total magnitude of forces applied to it is negative no

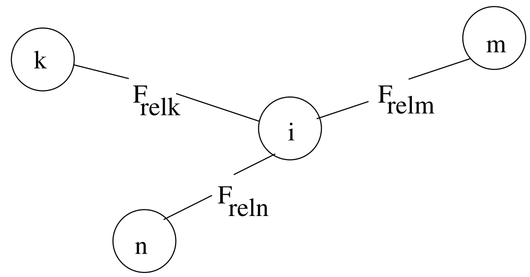


Figure 3.2: Relative forces applied to vehicle i.

clustering procedure is triggered since all the surrounding nodes tend to move away from it. Calculating total force F helps to avoid re-clustering in many situations - for example, when groups of vehicles move away from each other.

To illustrate this, consider figure 3.3 where three snap shots of a specific scenario are presented. It is assumed that nodes participating to a cluster are determined by nodes shape. Vehicles participating to a group are represented by a square and a free node is represented by a circle. In the left snapshot a group of vehicles that move along the east - west axis move towards a vehicle i which is moving along the north-south axis. In the center snapshot the vehicles meet and in the right snapshot the group of vehicles move away from vehicle i. At the top of the figure nodes are reclustered when the vehicles meet. The total force applied to node i the different time slots is positive for the two first moments and negative for the third. Vehicle i according to $Sp-cl$ is encouraged to form a cluster with the group of vehicles that move along different directions, in order to exchange useful information.

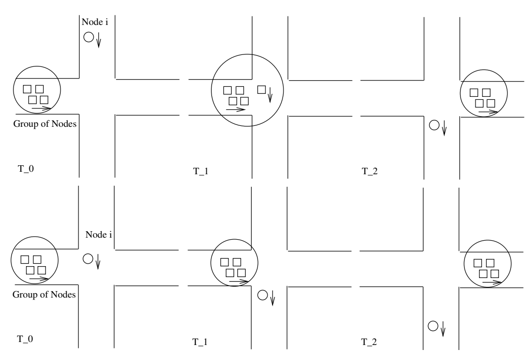


Figure 3.3: Reclustering procedure and its avoidance thanks to Relative forces.

In the occasion where vehicle i meet the group of nodes as shown in figure 3.3 at the bottom of the figure no reclustering from free node i would be triggered since the total force applied to it would be negative for all the time period. The negative relative force that is applied to node i, represents the fact that node i is moving away from the group of nodes and thus the moment of meeting will be very short and changing cluster structure at this moment may lead to another re-clustering, immediately after nodes move outside their transmission range. Also this short time of period will not be enough for the exchange of useful data.

All nodes are equipped with GPS receivers and On Board Units (OBU). Location information of all vehicles/nodes, needed for clustering algorithm is collected with the help of GPS receivers. The only communications paths available are via the ad-hoc network and there is no other communication infrastructure. Maximum Transmission Range (R) of each node in the vehicular network environment is 250 meters.

3.3.1 Neighborhood identification

Neighborhood identification is the process whereby a vehicle/node identifies its current neighbors within its transmission range. For a particular vehicle, any other vehicle that is within its radio transmission range is called a neighbor. All vehicles consist of neighbor set which holds details of its neighbor vehicles. The neighbors set is always changing since all nodes are moving. The neighbor set N_i of vehicle i is dynamic and is updated frequently. Every moving node keeps track of all current neighbors (their id's) the current and the past distance.

Generally,neighbor node identification is realized by using periodic beacon messages. The beacon message consists of node Identifier (ID), node location, speed vector in terms of relative motion across the axes of x and y (dx, dy) total force F, state and time stamp. Node location is used in order to calculate the distance between the nodes. Each node informs other nodes of its existence by sending out beacon messages periodically.

All nodes within the transmission range of source/packet carrier node will announce their presence by sending beacon messages frequently. After the reception of a beacon, each node will update its neighbor set table. For a neighbor that already exists in its neighborhood only the current and the past distance are updated. If a node position is changed, then it will update its position to all neighbors by sending a beacon signal. If a known neighbor, times out, it will be removed from the neighbor set table. The total number of neighbors of a given vehicle is called the “active neighborhood set” N_i of the vehicle

3.3.2 Clustering process and protocol structure

As described in the previous section the beaconing thread is responsible for exchanging informations between neighboring nodes. Another task of this thread is processing and proper use of the messages received from other nodes. Each node constantly updates knowledge about neighboring nodes. Each node i using the information of the beacon messages calculates the pairwise relative force F_{relij} for every neighbor applied to every axes j using the coulomb law.

$$F_{relijx} = k_{ijx}\frac{q_i q_j}{r_{ij}^2}, \quad F_{relijy} = k_{ijy}\frac{q_i q_j}{r_{ij}^2} \tag{3.1}$$

where r_{ij} is the current distance among the nodes k_{ijx} (k_{ijy}) is a parameter indicating weather the force among the nodes is positive or negative depending on whether the vehicles are approaching or moving away along the corresponding axis and q_i and q_j may represent a special role of a node (e.g. best candidate for Cluster head due to being close to an RSU, or due to following a predefined route (bus)).

In coulombs law a positive force implies it is repulsive, while a negative force implies it is attractive. In our implementation a positive force symbolizes the fact that the specific pair of nodes is approaching or is moving towards the same direction while a negative force is applied to nodes that move to different directions. Every node computes the accumulated relative force applied to it along the axes x and y and the total magnitude of force F. According to the current state of the node and the relation of its F to neighbor's F, every node takes decisions about clustering formation, cluster maintenance and role assignment.

A node may become a clusterhead if it is found to be the most stable node among its neighborhood. Otherwise, it is an ordinary member of at most one cluster. When all nodes first enter the network, they are in non-clustered state. A node that is able to listen to transmissions from another node which is in different cluster can become a gateway. We formally define the following term: relative mobility parameters k_{ijx} and k_{ijy}.

Definition 2. *Relative mobility parameters k_{ijx} and k_{ijy} between nodes i and j, indicate whether they are moving away from each other, moving closer to each other or maintain the same distance from each other. To calculate relative mobility, we compute the difference of the distance at time, t and the possible distance at time, $t + dt$ for every axis.*

Relative mobility at node i with respect to node j is calculated as follows:

We calculate the distance at every axes between the nodes at time t and the possible distance at time $t + dt$ according to,

$$D_{cxij} = x_i - x_j, \quad D_{fxij} = x_i + dx_i - x_j - dx_j \tag{3.2}$$

$$D_{cyij} = y_i - y_j, \quad D_{fyij} = y_i + dy_i - y_j - dy_j \tag{3.3}$$

The relative movement dx and dy of every vehicle along the axes x and y are calculated by the vehicles OBU according to previous data received from the GPS with respect to the traffic ahead (figure 3.3.2). According to mobility in every axis relative mobility k_{ijx} and k_{ijy} are calculated according to :

$$if \quad D_{cxij} \leq D_{fxij} \quad then \quad k_{ijx} = -a_x dt. \tag{3.4}$$

$$if \quad D_{cxij} \geq D_{fxij} \quad then \quad k_{ijx} = a_x dt. \tag{3.5}$$

where a_x and a_y are given by

$$if \quad D_{cxij} \leq D_{fxij} \quad then \quad a_x = D_{fxij} - D_{cxij} \tag{3.6}$$

$$if \quad D_{cxij} \geq D_{fxij} \quad then \quad a_x = \frac{1}{D_{cxij} - D_{fxij}} \tag{3.7}$$

Parameters a_x and a_y indicate the significance of the force applied between the vehicles by reflecting the ratio of divergence or convergence among moving nodes. In equation 3.6 a_x is proportional to the divergence among nodes, since the faster it takes place the more negative the force must be. In equation 3.7 a_x is proportional to the reverse difference of the distance among the nodes, since nodes that approach each other in a fast pace wont probably stay in contact for a sufficient amount of time in order to form cluster and exchange information.

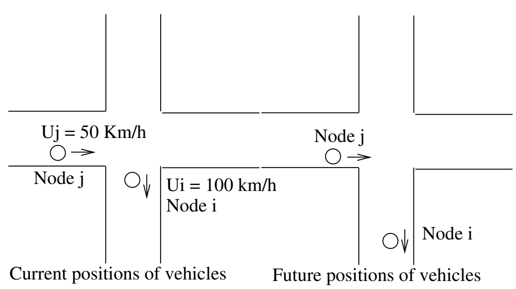


Figure 3.4: Relative mobility at node i with respect to node j.

3.3.3 Special role of vehicles

The pairwise relative force $Frel_{ij}$ for every pair of nodes depends on the relative mobility $Krel_{ij}$, the current distance and parameters q_i and q_j which indicate a special role for the vehicles. In the cluster creation procedure, nodes that driver intentions can be predicted like truck drivers that keep an almost constant velocity must be favored to become clusterheads. Also in urban areas vehicles that follow the same routes constantly may act as clusterheads in such a dynamic environment. In a street with many lanes, cars that follow the non turn lane are also best candidates for clusterhead since they are expected to stay longer on the street (Figure 3.5).

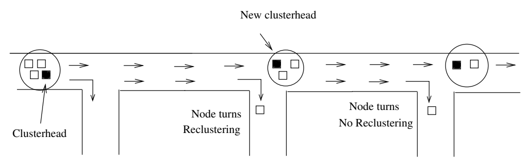


Figure 3.5: The correct choice of the clusterhead plays significant role.

Historical data about driver behavior may be used in order to select the appropriate clusterheads. In Sp-Cl parameter q_i are used to favour vehicles to become clusterheads. Using equation 3.1 to compute the relative force between two nodes, parameter q_i is used as follows:

$$\text{if } k_{ijx} \geq 0 \;\; \text{then } q_i = 2, \; \text{if } k_{ijx} \leq 0 \;\; \text{then } q_i = 1/2. \tag{3.8}$$

Positive forces applied to these nodes are strengthened while negative are weakened, in order to facilitate this node to become a clusterhead. The correct choice of the clusterhead is very important for the stability of the method, the cluster lifetime and the overhead involved in forming and maintaining these clusters.

3.3.4 Cluster-head election parameters

Vehicles use beacon messages in order to broadcast information to neighboring nodes such as Identifier (ID), node location, speed vector in terms of relative motion across the axes of x and y (dx, dy) total force F, state and time stamp. Using this information as stated above nodes calculate the forces applied to each other according to position and relative mobility. The mobility information of the neighbors is needed for the vehicle to initiate the cluster formation request, while cluster-head election information for any node is limited to the nodes that are within range. After receiving information of all neighboring vehicles, node i calculates

$$F_x = \sum_{j \in N_i} F_{relijx} \quad and \quad F_y = \sum_{j \in N_i} F_{relijx} \tag{3.9}$$

which is the total force along axes x and y applied to it. The total value of forces (norms) is calculated for every node according to :

$$F = |F_x| + |F_y| \tag{3.10}$$

Total force F is used to determine the suitability of a vehicle to become clusterhead according to the following criteria:

- The suitability value of the vehicle is calculated by considering the mobility information of its neighbors (parameters k_{ijx} and k_{ijy})
- Nodes having higher number of positive neighbors ($F_{relijx} \geq 0 \quad F_{relijy} \geq 0$), maintaining closer distances to their neighbors, should have be qualified to be elected as cluster-heads.

3.3.5 The cluster formation algorithm

In order to execute the algorithm, each vehicle is assumed to maintain and update the N_i set. At any time each vehicle i recalculates total F and according to total non-clustered members within range try to form a cluster and become the clusterhead.

If the node has the biggest positive force applied to it and there exist at least one free node in its neighborhood, it declares itself to be a clusterhead. In the opposite situation, where there exist a free node j with biggest total F in range the vehicle becomes a cluster member of j. This algorithm leads to the formation of clusters which are at most two hops in diameter.

3.3.6 Cluster maintenance

The cluster maintanance procedure follows the following rules:

- For every free node.

  When a standalone (non-clustered) vehicle comes within R distance from a nearby cluster-head, the cluster-head and the vehicle compare the total force f applied to them. If the relative force F of the clusterhead is bigger than that of the free node then the cluster-head will accept the vehicle and will add it to the cluster members list. If the clusterhead has smaller F then no action is triggered.

  If there other nearby standalone vehicles it compares the values of F in order to form a new cluster.

- For every member node.

  If a member node at a certain time finds itself to have bigger F than any of the surrounding clusterheads then it becomes a free node and tries to form its own cluster.

When a cluster member moves out of the clusterhead's transmission range. it is removed from the cluster members list maintained by the cluster-head and it becomes a free node again.

- For every clusterhead.

  when two cluster heads come within each other's transmission ranges and they stay connected over a time period CCI the cluster merging process takes place. The clusterhead with the lower F gives up its cluster-head role and becomes a cluster-member in the new cluster.

3.4 Simulation and performance evaluation

An extensive simulation study was conducted to evaluate the performance of our protocol using a custom simulator. In our simulation, we consider different road traffic and different network data parameters. The simulation environment is a one direction 5-lane highway with a turn in order to evaluate the performance of the scheme.

The total length of the highway is 2 Km. The stationary LPGs created in each scenario are of size comparable to communication range of nodes, i.e. if the communication range of the vehicles is 80 meters each LPG is of 160 meters long as if the RSU has range of 80 meters.

3.4.1 The mobility model

The arrival rate of the vehicles follows the Poison process with parameter λ. The speed assigned to the vehicles is according to the lane it chooses to follow according to Table 3.1.

Lane	*Speed km/h*
1	80
2	100
3	120
4	140
5	160

Table 3.1: Speed per lane.

The density of the vehicles depends on parameter λ. The number of vehicles per lane is between (2 -15 v/km/Lane) depending on the speed being used and the value of parameter λ according to Table 3.2.

parameter λ	ν*/km/Lane*
3	8-15
5	5-9
7	3-6
9	2-5

Table 3.2: Density per lane.

3.4.2 Evaluation criteria

To show the performance of our proposed Spring Clustering Algorithm (Sp-Cl), we compare it with the Lowest-id (Low-id) and the stationary Local Peer Group (LPG) architecture proposed in [40] and [48] respectively. The Lowest-ID algorithm forms of clusters which are at most two hops in diameter.

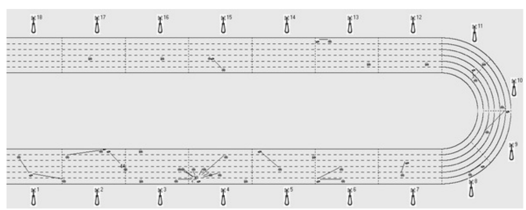


Figure 3.6: Simulation environment.

The basic concepts of Lowest id are the following. Each node is given a distinct ID and it periodically broadcasts the list of its neighbors (including itself). A node which only hears nodes with ID higher than itself is a “clusterhead” (CH). The Lowest-ID node that a node hears is its clusterhead, unless the Lowest-ID specifically gives up its role as a clusterhead (deferring to a yet lower ID node). A node which can hear two or more clusterheads is a “gateway”. Otherwise, a node is a free node.

The basic architecture feature of the stationary LPG is to use a GPS-based grid to partition roadways into zip code areas that define LPGs. In stationary LPG all LPG areas are location based and well defined. Members of LPG dynamically change as vehicles move along the highway.

We compare the three methods under the same environment variables. Each simulation run repeated times with different random seeds and the collected data was averaged over those runs.

Snapshots of the simulation of the three methods are shown in Figures 3.7, 3.8 and 3.9.

Cluster stability

Figure 3.7: Simulation highway Spring-Clustering.

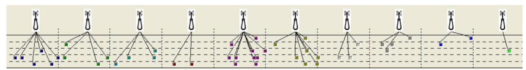


Figure 3.8: Simulation highway LPG.

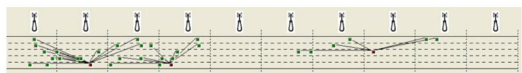


Figure 3.9: Simulation highway Lowest-id.

In order to evaluate the stability of the algorithm, we measure the stability of the cluster configuration against vehicle mobility. In a high dynamic VANET, nodes keep joining and leaving clusters along their travel route. Good clustering algorithms should be designed to minimize the number of cluster changes of the vehicle by minimizing reclustering. This transitions among clusters are measured in order to evaluate the performance of the algorithm. The basic transition events the vehicle encounters during its lifetime:

- A vehicle leaves its cluster and forms a new one (becomes a clusterhead).
- A vehicle leaves its cluster (due to communication range) and joins a nearby cluster or becomes free.
- A cluster-head merges with a nearby more stable cluster.

We compare the average transition events of the vehicles for the Sp-Cl,Low-Id and LPG methods when different speeds and different transmission ranges are used. From Figure 3.10, we can see that the average transitions produced by our Sp-Cl technique is smaller compared to that produced by the Low-ID and LPG methods. This means our technique causes less number of cluster transitions for all different density topologies. Similar figures were produces for other transmission ranges.

The figures show that the average transitions of the vehicle decreases as the transmission range increases. This is because increasing the transmission range, increases the probability that a vehicle stay connected with its cluster-head.

Number of clusters

Due to high dynamics of the VANET, clusters are created (new clusters added to the system) and dissolved over time. The total number of clusters created over a period of

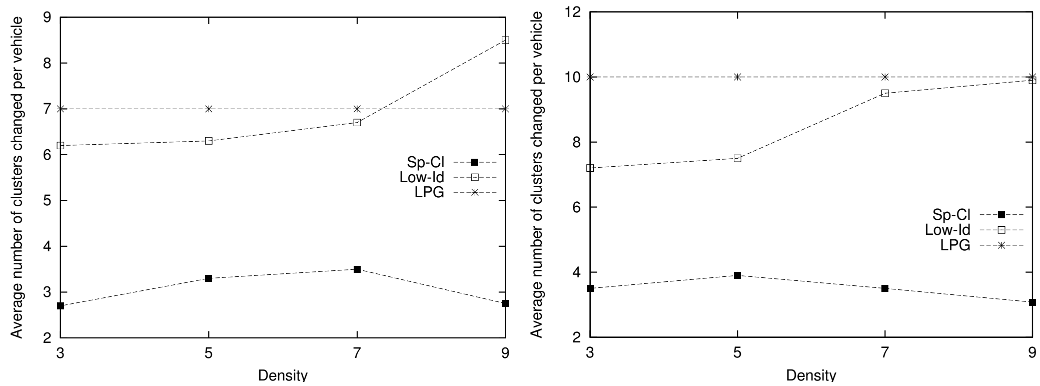


Figure 3.10: Average cluster change per vehicle for Sp-Cl, LPG and Low-Id methods for different transmission ranges (125m, 80m).

time is a metric of the stability of the clustering method used. Good clustering algorithms should be designed to reduce the rate at which clusters are created and added to the system due to the mobility of the nodes. The ability of the clustering method to maintain cluster structure despite vehicles mobility defines its performance. In this simulation we counted the new clusters which are added to the system.

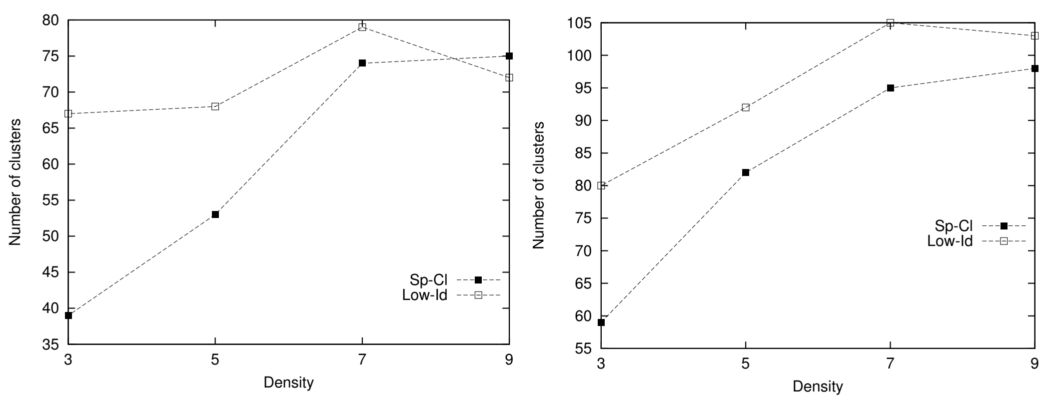


Figure 3.11: The average total number of formed clusters for Sp-Cl and Low-Id methods for different transmission ranges (125m, 80m).

Figure 3.11 shows that the total number of clusters created by Sp-Cl is always smaller compared to that produced by the Lo-Id method and this number decreases as the transmission range increases. This is because the Sp-Cl method uses the accumulated forces among as a parameter to create the clusters. Thus, the clusters are more stable and have longer lifetime.

Cluster lifetime

The average cluster lifetime is an important metric that shows the performance of the clustering algorithm. The cluster lifetime is directly related to the lifetime of its cluster-head. The cluster-head lifetime is defined as the time period from the moment when a vehicle becomes a cluster-head to the time when it is merged with a nearby cluster.

The average cluster lifetime produced by the Sp-CL and the Low-Id methods is evaluated in various topologies with different transmission ranges, and the results are illustrated in Figure 3.12.

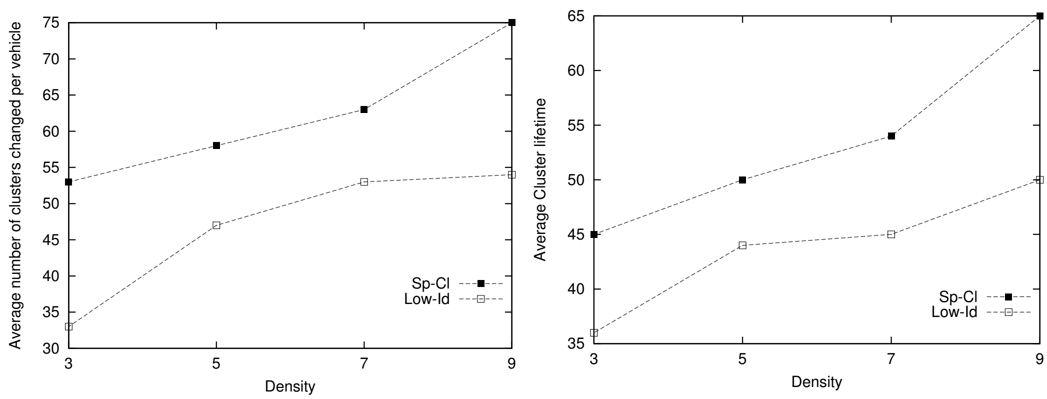


Figure 3.12: The average cluster lifetime Sp-Cl and Low-Id methods for different transmission ranges (125m, 80m).

3.5 Enhanced Spring clustering

In Figure 3.5 an accident happens in a highway at a point in time where traffic is intense, and we also suppose that the vehicles approaching the place of accident are able to "detect" the accident. The accident results in the highway being blocked. In such a situation density of vehicles increases dramatically and a clustering method is necessary for the proper dissemination of safety messages.

Spring Clustering which was introduced forms stable clusters based on force directed algorithms. That protocol suffers from the fact that it treated vehicles only as senders, relays and receivers and not as part of the environmental obstacles. A number of V2V measurements have been performed to study the statistical properties of V2V propagation channels [49, 50]. In [51] it is observed that in rush hours the received signal strength gets worse compared to no traffic hours for the same part of an open road. These observed differences can only be related to other vehicles obstructing Line-of-Sight (LOS), since the system parameters remained the same during the measurements. Recent work reported in [52, 53] showed that vehicles as obstacles have an important influence on the behavior of a VANET system.

Highly realistic channel models [54] gives results that are in very good agreement with the real world. However, these models are computationally too expensive making them impractical for extensive simulation studies. In [52] it is shown that the vehicles as obstacle have a significant impact on LOS obstruction in both the dense and sparse vehicular networks, therefore, shadow fading effects due to other vehicles are very important to be included in channel models.

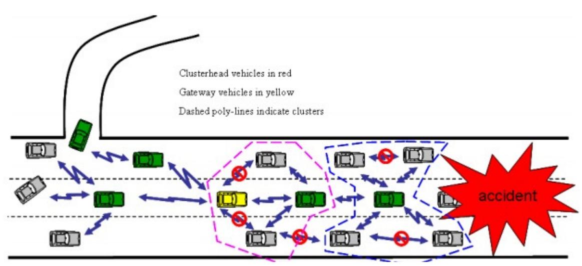


Figure 3.13: Illustration of an example where vehicle clustering is important.

3.5.1 Special role of vehicles – Enhanced Spring Clustering

In the Spring Clustering method, vehicles that follow predefined routes or keep a relative constant velocity are favored to become clusterheads due to being more stable in terms of mobility. Except from being best candidates based on mobility criteria, tall vehicles like trucks and busses suffer less from vehicle obstruction. The maximum distance over which communication is still possible is significantly larger than when neither the transmitter nor the receiver is a tall vehicle. Selecting tall vehicles as clusterheads increases the probability that members stay for a longer time connected to them. The correct choice of the clusterhead is very important for the stability of the method, the cluster lifetime and the overhead involved in forming and maintaining these clusters.

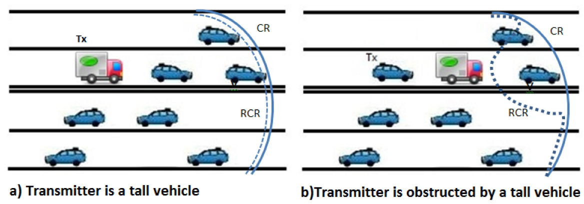


Figure 3.14: Reliable Communication Range (RCR) is significantly larger for tall vehicles

In Figure 3.14 when a tall vehicle is the transmitter the Reliable Communication Range (RCR) is almost the same to the ideal communication range (CR) when vehicle obstruction is neglected (a). When a car being a transmitter is obstructed by a tall vehicle the RCR is strongly affected in the area around the obstacle (b). Taking this characteristics in mind we incorporate height as a criterion for the clusterhead election. When a node i finds itself to be among the tallest in its one-hop neighborhood then parameter q_i is used to favor it become a clusterhead. In order for this new criterion to be incorporated to our method an extra byte is added to the beacon message. The beacon message now except from node Identifier (ID), node location, speed vector, total force F, state and time stamp, also includes vehicle height or, for simplicity, a vehicle category in terms of tall or short. The new method is called Enhanced Spring Clustering (ESC).

3.6 VANET diffraction models

Several propagation models applied in VANET research can be used to quantify the impact of vehicles as obstacles on the electromagnetic wave propagation. Geometry based deterministic models are used to analyze particular situations. A highly realistic model, based on optical ray tracing was proposed in [54]. The accuracy of the model is achieved at the expense of high computational complexity and location-specific modeling. There are simplified geometry based deterministic models [52, 55]. In particular, the research work proposed by Boban et al. in [52] derive a simplified geometry-based deterministic propagation model, in which the effect of vehicles as obstacles on signal/wave propagation is isolated and quantified while the effect of other static obstacles (i.e., buildings, overpasses, etc.) is not considered. The research work in [52] focuses on vehicles as obstacles by systematically quantifying their impact on line of sight and consequently on the received signal power.

For the received power level, the impact of obstacles can be represented by signal attenuation. This increase in attenuation is due to the diffraction of the electromagnetic waves. To model vehicles obstructing the line of sight, we use the knife-edge attenuation model.

If only one obstacle is located between Tx and Rx, then the single knife-edge model described in ITU-R recommendation [56] is used. If the direct line-of-sight is obstructed by a single knife-edge type of obstacle, we use the following diffraction parameter v:

$$\nu = h * \sqrt{\frac{2}{\lambda} * (\frac{1}{d1} + \frac{1}{d2})} \tag{3.11}$$

where h is the height of the top of the obstacle above the straight line joining the two ends of the path. If the height is below this line, h is negative. d1 and d2 are distances of the two ends of the path from the top of the obstacle.

The diffraction loss can be closely approximated by

$$A_d = \begin{cases} 6.02 + 9.00\nu + 1.65\nu^2 & \text{if} \quad -0.8 \leq \nu < 0 \\ 6.02 + 9.11\nu - 1.27\nu^2 & \text{if} \quad 0 \leq \nu < 2.4 \\ 12.953 + 20\log\nu & \text{if} \quad \nu \geq 2.4 \end{cases}$$

When two or more vehicles exist between the transmitter and receiver horizons then diffraction loss is calculated using the Epstein-Peterson method [57]. The diffraction loss is taken as the sum of the individual knife edges according to Figure 3.15.

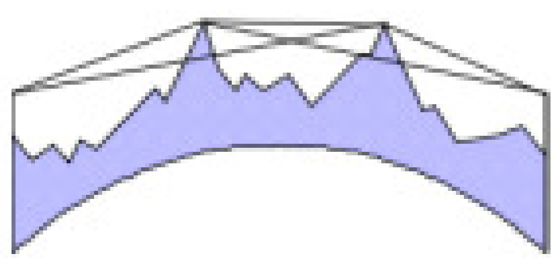


Figure 3.15: Epstein-Peterson method.

3.6.1 Placement of antennas

The antennas in vehicles are usually mounted either on the roof or inside the vehicle (e.g., under the windshield, in rear view mirror, on a seat or near a dashboard). The effect of the antenna placement in vehicles is significant [53]. The differences in the cumulative link packet error rates are until $25-30$ % depending on the antenna locations under LOS conditions. This is due to the fact that the height of the antenna affects the attenuation caused by obstructing vehicles. In order to make the environment more realistic we incorporated the placement of the antenna in our simulation environment. The model used has cars that either have the antenna mounted on the roof at a height of 10cm above the car or under windshield at 50cm below the height of the car.

3.6.2 Obstructed Line Of Sight

To separate the Line Of Sight from OLOS (Obstructed Line Of Sight) and NLOS (No Line Of Sight) cases at every time instance we use a simple algorithm to detect intersections between line segments. Every node in the simulator is represented as a rectangle occupying an area of the simulation space. For every pair of nodes we draw a line connecting their antennas. If the line intersects with one rectangle then the single knife model is used to calculate the additional loss. In case when the line crosses more than one rectangle then the Epstein-Peterson model is used.

In our simulator the vehicles are produced by our generator. In case where the vehicles are coming from a mobility simulator like SUMO the same procedure is used. When the cars are blocked by a building then we have a NLOS situation. In this case it is impossible to make a straight line between the two vehicle positions without being obstructed by a building and the transmission range is usually limited to a few meters. In the scenarios investigated in this research we are focused to find the impact of vehicles in RCR and for that reason when we have NLOS situation we consider the range to be zero.

When the line connecting two vehicles is obstructed by another car then we say that we have a OLOS situation (Obstructed line of sight). According to the number of obstructing vehicles the corresponding diffraction method is used to calculate the power loss and determine the RCR among the nodes. OLOS situations are very important since vehicles, due to diffraction, have different communication ranges with their neighbors according to their environment. The implementation of the simulator that incorporates the vehicles as obstacles is presented in simple steps in the next section.

3.7 VANET implementation

- Every vehicle and building is modeled as a rectangle with proportional dimensions.
- For every instance and pair of nodes a straight line is drawn from antenna position of each TX vehicle to the antenna position of each RX Vehicle.

- If the line does not touch any other rectangle, TX/RX has LOS.
- If the line passes through one rectangle at least , LOS is obstructed by a vehicle or by a building, the two cases can easily be distinguished by using the geographical information available in simulator or by the sizes of the obstacles.
- Once the propagation condition is identified, the simulator can simply use the relevant model to calculate the power loss.
- According to power loss the Reliable Communication Range (RCR) between each pair of nodes for every time instance is calculated.
- Enhanced Spring Clustering performance is evaluated according to Reliable Communication Range.
- Enhanced Spring Clustering is evaluated according to proper clusterhead election.

3.8 Simulation and performance evaluation

A simulation study was conducted to evaluate the performance of our protocol using a custom simulator. The competitor is the Spring Clustering method, since it is shown there that it is superior to many current high-performance VANET clustering protocols. In our simulation, we consider various road traffic and network data parameters. The simulation environment (Figure 3.16) is a two direction, 3-lane per direction highway with a turn in order to evaluate the performance of the scheme. All nodes are equipped with GPS receivers and On Board Units (OBU). Location information of all vehicles/nodes, needed for the clustering algorithm is collected with the help of GPS receivers. The only communications paths available are via the ad-hoc network and there is no other communication infrastructure. The power of the antenna is P_{tx} = 18dBm and the communication frequency f is 5.9 Ghz.

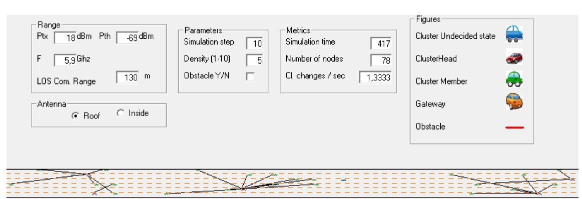


Figure 3.16: Simulation environment.

The communication range of the vehicles is calculated according to Table 3.3. In our simulations, we use a minimum sensitivity (P_{th}) of -69 dBm which gives a transmission range of 130 for LOS.

Data Rate (Mb/sec)	*Minimum Sensitivity(dBm)*
3	-85
4.5	-84
6	-82
9	-80
12	-77
18	-70
24	-69
27	-67

Table 3.3: Minimum sensitivity in receiver antenna according to data rate.

In a situation of OLOS the Reliable Communication Range (RCR) is calculated according to:

$$P_{tx} - FPLS - A_d < P_{th} \tag{3.12}$$

where the value of d is the biggest one that makes Equation 3.12 true and FPLS is the loss in signal strength of an electromagnetic wave that would result from a line-of-sight path through free space (usually air), with no obstacles nearby to cause reflection or diffraction. FPLS is calculated according to Equation 3.13

$$FPLS = 92.45 + 20 * Log(RCR) + 20 * Log(f). \tag{3.13}$$

3.8.1 The mobility model

The arrival rate of the vehicles follows the Poison process with parameter λ. The speed assigned to the vehicles is according to the lane it chooses to follow according to Table 3.4.

Lane	*Speed km/h*
1	80
2	100
3	120

Table 3.4: Speed per lane for both directions.

The density of the vehicles depends on parameter λ. The number of vehicles per lane is between (2 -15 v/km/Lane) depending on the speed being used and the value of parameter λ according to Table 3.5.

Parameter λ	ν*/km/lane*
3	8-15
5	5-9
7	3-6

Table 3.5: Density per lane.

3.8.2 Evaluation criteria

In order to evaluate the effect of vehicles as obstacles in the performance of Enhanced Spring Clustering method we conducted several simulation scenarios. The vehicles that enter the simulation are of 25 % trucks with average height of 2.5 meters, when the rest of the cars have height of 1.5 meters. The placement of the antenna is in the roof of the car at a height of 10 decimeters above the car or inside the car at a height of 0.5 meter below the roof of the vehicle(e.g., 2 meters for trucks and 1 meter for cars).

We measured how density of the cars affects the performance of our method when cars are treated as obstacles. We observed that the placement of the antenna inside the car gives a big difference in RCR and to the overall performance of the clustering.

We also measured how the ratio of tall vehicles affect the Reliable Communication Range (RCR). We investigated how favoring tall vehicles to become clusterheads (Enhanced Spring Clustering) in a realistic scenario, where RCR is computed among each pair of nodes, makes the method more stable.

Clustering performance. In order to evaluate the effect of OLOS in stability of our clustering method we created scenarios of different car densities. The results clearly indicate that vehicles as obstacles have a significant impact on the formation of clusters in a typical VANET clustering method compared to those acquired when we neglect this phenomenon. This is due to the fact that the medium contention is overestimated in models that do not include vehicles as obstacles in the calculation and that the RCR is considered to be significantly bigger than in the real world (Figure 3.17).

Also, the number of mean cluster change (Figure 3.18) and the average lifetime (Figure 3.19) of clusters are also affected when the cars are treated as obstacles in the simulation.

The obstructing vehicles decrease RCR which leads to what is called a *lost link event*, causing the presence of often transition events to vehicles. Nodes that loose communication with their neighbors, due to attenuation by obstructing vehicles, leave their clusters, join other nearby ones or form new clusters degrading the overall performance of the clustering method.

Antenna placement. The placement of the antenna inside the car or at the roof

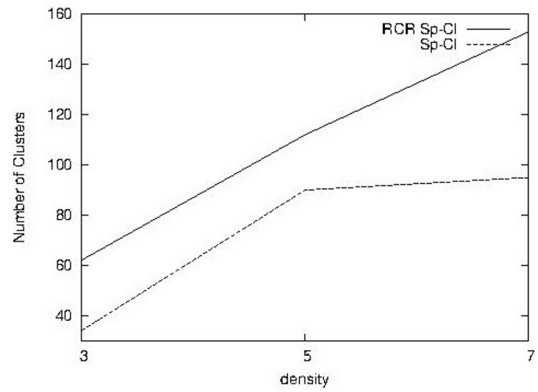


Figure 3.17: Impact of OLOS in average number of clusters

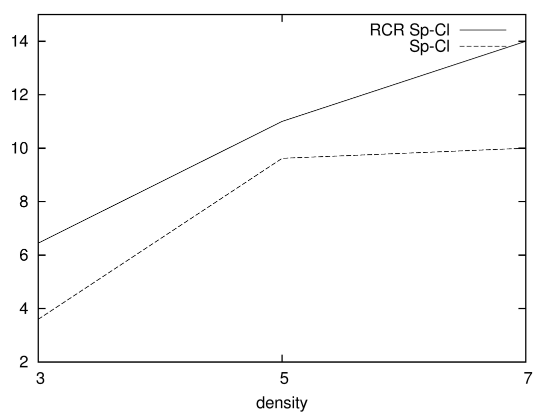


Figure 3.18: OLOS influence average cluster changes / vehicle

of it, plays a significant role at the attenuation caused by the obstructing cars. Table 3.6 shows mean RCR for different antenna placements. When the antenna in mounted inside the car it is under windshield at 50cm below the height of the car and the cars enter the simulation with parameter $\lambda = 5$.

placement	RCR	A_d
roof	80	10
inside	45	25

Table 3.6: Antenna placement.

Tall vehicles. When the percentage of the tall vehicles (trucks) increases the mean RCR is strongly affected. This is due to the fact that the attenuation caused by this

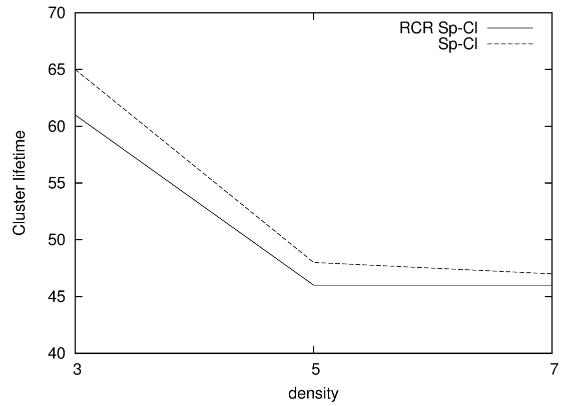


Figure 3.19: Average cluster lifetime under OLOS

vehicles when in the middle of the communication range of other cars is bigger. The reliable communication range, when there is an obstacle between transmitter and receiver, drops to almost half of the static communication range when 50 % of the cars are tall vehicles and increases again as this percentage increases.

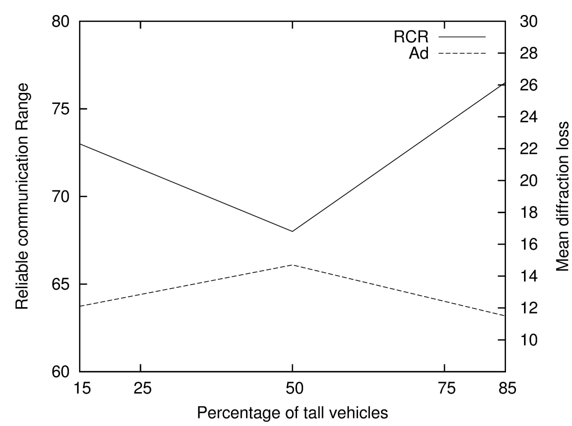


Figure 3.20: Number of Trucks (tall vehicles) influence reliable communication range

Special role of vehicles. In order to evaluate how favoring tall vehicles to become clusterheads affects Enhanced Spring Clustering, we created scenarios of different vehicle distributions. The average density is of $\lambda = 5$, the percentage of tall vehicles is 15% and all vehicles move to the same direction. The simulations showed that favoring tall vehicles in a realistic scenario, where reliable communication range is computed among each pair of nodes, makes Enhanced Spring Clustering more stable (Figure 3.21).

This is due to the fact that tall vehicles have biggest average reliable communica-

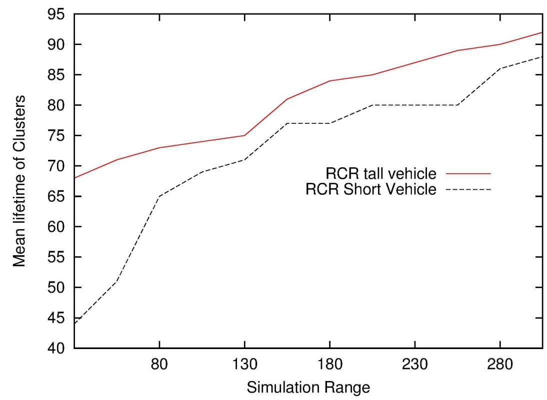


Figure 3.21: Tall vehicles play significant role in Spring Clustering.

tion range (RCR) with their neighbors compared to short vehicles, since the latter are more vulnerable to diffraction losses. This phenomenon doesn't show up when obstructing vehicles are neglected, since in that situation all vehicles have static communication ranges (CR) with all their neighbors, independently of their height or other obstructing cars (Figure 3.22).

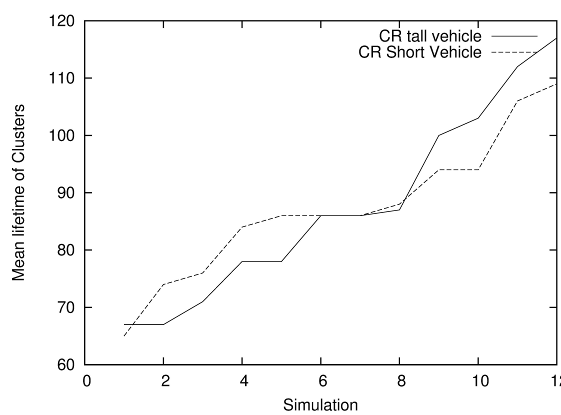


Figure 3.22: Height of clusterhead does not affect Spring Clustering's performance, when static communication range is used

Average number of clusters and mean cluster transitions per vehicles also follow the same patterns, showing that Enhanced Spring Clustering which assigns special roles to tall vehicles is more stable than the initial one.

3.9 Vitrual Forces Vehicle Clustering

In our proposed scheme $VFVC$, we extend the meaning of special roles on vehicles used in $Sp - Cl$. The charge of every vehicle, is proportional to many parameters that affect its behavior in the network. All vehicles are assigned an initial electric charge Q. Vehicles according to their status (e.g. lane they belong to, car height, public transport etc.) are assigned a different amount of load ($Q(i)$) at each time step.

The characteristics that give vehicles extra charge are:

- Vehicles that follow predefined routes like a bus (Q_p)
- Tall vehicles like trucks (Q_T)
- Vehicles that follow non-turning lanes in a multi lane main street (Figure 3.5) ($Q_d(t)$).
- Vehicles that their driver behavior is statistically smooth (Q_b).
- Vehicles that based on historical data, mobility can be predicted (Q_h)

The total charge $Q(i)$ that is given to every vehicle at each time step according to the parameters described above, is given by equation 3.14. The only parameter that is dynamic is $Q_d(t)$ since the vehicle as it moves along the street may change lanes and follow turning or non turning lanes at different time steps.

$$Q_i = Q * Q_p * Q_T * Q_d(t) * Q_b * Q_h \tag{3.14}$$

In the simple scenario where all vehicles have the same characteristics and direction is not taken in mind all vehicles are equally charged

$$Q_i = Q \ \forall i, \ time \ step \tag{3.15}$$

and Virtual Forces Vehicular Clustering $VFVC$ performs like the original $Sp - Cl$.

In order to enhance the performance of the method in Urban environments we incorporate in every beacon message one additional byte of information about the lane the vehicle belongs to. Positive forces applied to these nodes are strengthened while negative are weakened, in order to facilitate this node to become a clusterhead. The correct choice of the clusterhead is very important for the stability of the method, the cluster lifetime and the overhead involved in forming and maintaining these clusters.

3.9.1 Direction matters, parameter $Q_d(t)$

In urban environments where vehicles change directions often, parameters concerning direction need to be taken in mind in order to perform clustering. The $VFVC$ method that we propose uses parameter $Q_d(t)$ in order to favor vehicles to become clusterheads according to the lane they belong to. There are three main traffic flows at an intersection: Left Turn (LT), Right Turn (RT), and No Turn (NT). The intersection may have all three types of traffic flows or only some of them. LT is applied to the leftmost lane(s) if it splits the traffic to the left, RT is applied to the rightmost lane(s) if it splits the traffic to right, while NT is applied to the lane(s) in the middle if traffic goes straight.

In a multi-lane street vehicles that follow a non turning lane are better candidates to become clustrheads since they are going to stay longer on the street. If a vehicle that is going to leave the street soon, is elected as a clusterhead then major re-clustering is going to take place when it turns to another road segment, since it leaves all of its members orphans. In case a member node i leaves the street in order to follow another edge of the network, only this vehicle tries to find a nearby cluster to enter. Charges are assigned to cars according to the lane they belong according to the following rules.

- If the car follows a non turning lane then $Q_d(t)$=2
- If the car follows a turning lane then $Q_d(t)$=1
- If lane the car belongs to is going straight or turns then the $Q_d(t)$=1.5

In most of the cases this method increases the performance of spring clustering since expect from the most stable node in terms of relative mobility and velocity also a sense of future direction is used in order to perform clustering.

3.9.2 Lane Detection

Virtual Forces Vehicular clustering is based on the assumption that each vehicle knows its exact lane on the road via a lane detection system and a digital street map [58]that includes lane information for every road segment. Localization of vehicles in mainly conducted through GPS either as a standalone system or combined with a wheel odometer [59] for better detection of lane changes.

Also a beacon network using infrastructure to triangulate vehicle position can be used [60]. Other algorithms do not use GPS, and instead use techniques such as vision [61], LIDAR (Light Detection and Ranging) [62] etc. In case a vehicle isn't equipped with any localization mechanism, relative positions of its one hop neighbors could be used in order to detect its lane with a good precision.

3.10 Simulation and performance evaluation

Our proposed clusterhead selection algorithm was evaluated through detailed simulation on an urban environment. We simulated an area from city of Volos in Greece that is shown in figure 3.23 and is 2km x 600m.

Figure 3.23: Urban area of Volos

After aggregating the road segments that have the same attitudes we have simulated the area in SUMO as shown in figure 3.24

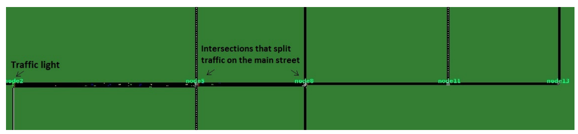


Figure 3.24: Simulated Urban area of Volos

The area consists of thirteen intersections. Only intersections 5 and 8 (see figure 3.24) split the traffic of the main street of interest. The intersections split the traffic into three different directions. The first intersection has four lanes, dividing the traffic into two directions: one lane to the right (TR) and three lanes going straight (NT). The second intersection has three lanes, dividing the traffic into two directions: one lane to the left (TL) and two lanes going straight. We focused only on one traffic direction.

Vehicles follow three different route distribution according to table 3.7. These distributions are used in order to favor vehicles follow the main street or turn in the intersections in a probabilistic way and not follow deterministic routes.

Route	*Intersection 5*		*Intersection 8*	
	NT	*TR*	*NT*	*TL*
1	80%	20 %	90%	10%
2	80%	20 %	10%	90%
3	20%	80 %	50%	50%

Table 3.7: Route distributions

The vehicle type ratio used for this simulation was 15% trucks and 85% sedans. We inject vehicles on the road and for the first 110 seconds of simulation time we use a traffic light in the beginning of the area of interest. A traffic light is used in order to have all vehicles injected as a group in the area. We follow them until they leave the straight section of the road turning left or right. In that way we are focusing on what happens on a central road, where cars enter and leave it all the time, if we favor cars that follow the non turning lane to become clusterhead. The parameters of $VFVC$ used in the simulation scenarios are listed in table 3.8.

VFVC parameter	*Simulated*	*Parameter value*
$Predefined\ routes$	No	1 (default)
$Vehicle's\ Height$	Yes	2(Tall), 1(Short)
$Vehicle's\ Lane$	Yes	2(NTL), 1.5 (TL & NTL), 1(TL)
$Driver\ behavior$	No	1 (default)
$Mobility\ prediction$	No	1 (default)

Table 3.8: Parameters of Virtual Forces Vehicular Clustering.

The traffic simulation is conducted with SUMO [63] and the trace files are injected to our custom simulator in order to perform clustering. We ran 10 different runs for each scenario of different communication ranges and speed limits. The vehicles were given different maximum speeds to provide a realistic highway scenario. Random maximum speeds were assigned to the different vehicles by providing SUMO with a probability distribution input.

Scenario	*Transmission Range*	*Max Speed limit*
1	130 m	80 - 36 Km/h
2	200 m	80 - 36 Km/h
3	250 m	80 - 36 Km/h
4	300 m	80 - 36 Km/h

Table 3.9: Scenarios tested during the simulation.

All nodes are equipped with GPS receivers and On Board Units (OBU). Location information of all vehicles/nodes, needed for the clustering algorithm is collected with the help of GPS receivers. The only communications paths available are via the ad-hoc network and there is no other communication infrastructure. The power of the antenna is $P_{tx} =$ 18dBm and the communication frequency f is 5.9 Ghz.

The reliable communication range of the vehicles is calculated according to Table 3.10. The reliable communication range is calculated for every pair of nodes at every instance based on the diffraction caused by obstructing vehicles [64]. In our simulations, we use a minimum sensitivity (P_{th}) of -69 dBm to -85 db which gives a transmission range of 130 to 300 meters.

We compare the average transition events of the vehicles for the Virtual Forces Vehicular clustering ($DFVC$), $Sp-Cl$ [65],$Lowest-ID$ [40] and $Mobic$ [39] methods when

Data Rate (Mb/sec)	*Minimum Sensitivity(dBm)*
3	-85
4.5	-84
6	-82
9	-80
12	-77
18	-70
24	-69
27	-67

Table 3.10: Minimum sensitivity in receiver antenna according to data rate.

different transmission ranges are used. From Figure 3.25, we can see the average cluster lifetime is bigger compared to that produced by the $Sp - Cl$, $Lowest - ID$ and $Mobic$ methods. The average clusterhead duration is the average length of time that a node remains a clusterhead, once it has been elected. Long clusterhead duration is important for MAC schemes where the clusterhead is the central controller and scheduler. Frequent changes to the clusterhead will degrade the performance of these cluster-based MAC schemes.

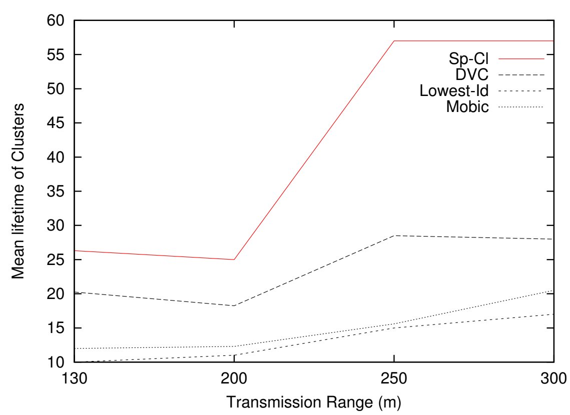


Figure 3.25: Average cluster lifetime vs transmission range

In this simulation we counted the new clusters which are added to the system. To effectively decrease network contention, fewer clusters is desirable. Typically, clustering algorithms strive to have only one clusterhead within a given broadcast range. Figure 3.26 shows that the total number of clusters created by $VFVC$ is always smaller compared to that produced by the other methods and this number decreases as the transmission range increases. The number of clusters is decreased compared to $Sp - Cl$ due to the fact that except current and future position and relative velocities among vehicles, also direction expressed in terms of the street lane occupied by the vehicle is used in order to select the more stable clusterhead.

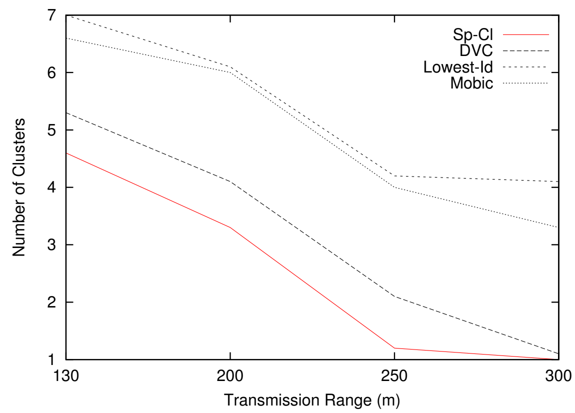


Figure 3.26: Average number of clusters for different transmission ranges.

From Figure 3.27, we can see that the average transitions produced by our $VFVC$ technique is very small compared to the other methods. The average rate of clusterhead change is the overall average number of clusterhead changes per second. The more clusters that are present, the greater the number of clusterhead changes; therefore this metric conveniently considers both clusterhead duration and the number of clusters formed. Similar figures were produced for different speed limits.

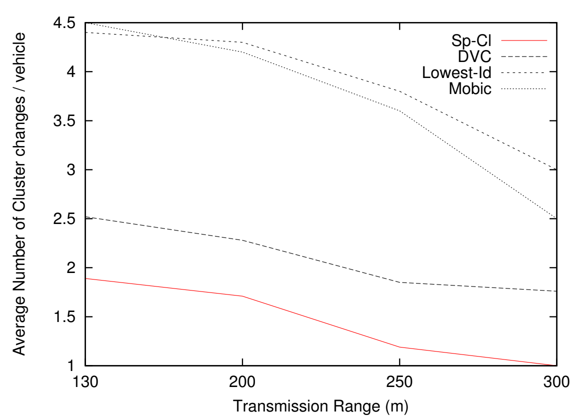


Figure 3.27: Clusterhead changes vs transmission range

The results clearly indicate that favoring vehicles that follow non turning lanes has a significant impact on the formation of clusters in a typical VANET clustering urban scenario where a major city area is simulated.

3.11 Chapter Conclusions

Clustering can provide large-scale VANETs with a hierarchical network structure to facilitate routing operations. We proposed a distributed clustering algorithm which forms stable clusters based on force directed algorithms. We proposed a mobility metric based on forces applied between nodes according to their current and their future position and their relative mobility.

The force applied between the vehicles reflects the ratio of divergence or convergence among them. We have simulated Sp-Cl and the results show that the performance of Sp-Cl is better than other existing algorithms. It also creates lesser and more stable clusters in order to achieve high scalability. The clusterhead change is relatively low and the overall performance of the method is stable to different topologies and transmission ranges

Treating vehicles as obstacles, has a significant impact on the reliable communication range. Reliable communication range that is calculated for every pair of nodes for every time instance according to the attenuation caused by obstructing vehicles is exploited for the design of a new VANET clustering protocol, namely the Enhanced Spring Clustering algorithm. It is shown that significant benefits are observed when the height of vehicles is taken into account when electing clusterheads; this feature is in the heart of the Enhanced Spring Clustering. This behavior is observed only in a realistic scenario where reliable communication range is computed among each pair of nodes instead of using a static communication range for all pairs of vehicles. The incorporation of heights in clusterhead election makes the method more stable.

Finally for Urban environments we proposed $VFVC$ that incorporates vehicle direction in clusterhead election mechanism. The results of simulations conducted show that $VFVC$ algorithm outperforms the other investigated methods, in terms rate of cluster-head changes (lower), total number of clusters (lower), average cluster lifetime(higher), translated in increased cluster stability, lower percentage of orphan nodes and larger cluster sizes. The stable clusters created by $VFVC$ can be used as a base so typical VANET routing algorithms can be applied for intra-cluster routing. Microscopic information of vehicles, such as velocities, current and future positions and vehicle's physical characteristics can be combined with macroscopic information from vehicles history in order to create trajectory-based schemes for clustering of vehicles. Social behavior of drivers can be used in order to make the clustering methods more stable.

Chapter 4
Clustering of vehicles based on social patterns of drivers*

Vehicular networks can bring great benefit on driving safety, traffic regulation, infotainment, and in many other practical applications. These applications require effective and efficient packet exchange between vehicles, which is a very challenging problem. VANETs, especially in urban environments, a node may have up to 100 neighbors (the radio range of the IEEE 802.11p may reach up to 1 km and the density of vehicles may reach more than 100 vehicles per kilometer) which situation will cause severe wireless network congestion, causing packet collisions and thus losses, which leads to bandwidth and CPU resources waste. Vehicle clustering is an established technique to alleviate this broadcast storm problem.

Though drivers tend to follow the same or similar routes, social behavior of vehicles moving in a city are ignored from previous clustering methods. A clustering method that uses macroscopic information from vehicles' history in order to create trajectory-based schemes for clustering of vehicles in VANETs is needed. This information can be combined with the microscopic information that vehicles exchange through periodic V2V messages, such as velocities, current and future positions and vehicle's physical characteristics (vehicle's height) in order to make the method robust to the dynamic mobility that vehicles exhibit in an urban environment.

4.1 INTRODUCTION

Mobility prediction is a thoroughly investigated, but still open to advances, topic. In the past the techniques of learning automata, Kalman filtering, pattern matching, and Markov modeling have been used. Learning automata [66] are simple, but they are not considered very efficient learners, because of the need to devise appropriate penatly/reward policies, and due to their slow convergence to the correct actions. Kalman filtering-based methods [67] construct a mobile motion equation relying on specific distributions for its velocity, acceleration and direction of movement; their performance largely depends on the stabilization time of the Kalman filter and knowledge (or estimation) of the system's parameters. Pattern matching techniques have been used for location prediction [67]. They compile mobility profiles, and perform approximate similarity matching, using the "edit

*The ideas presented in this chapter appear in the following publications [S.01,S.02] distance", between the current and the stored trajectories, in order to derive predictions. For the edit distance, it is hard to select the meaningful set of edit operations, to assign weights on them, and so on. The most effective and efficient algorithms are those based on Markov chains [68] and they can be appplied to any problem domain as long as we can convert the state-space of the prediction problem into a discrete-sequence prediction problem.

The investigation of social aspects in ad hoc networking is a topic of intense research in the past few years. Several studies have confirmed the existence of communities in ad hoc networks' nodes [69] or friendships among nodes [70] in mobile social networks. Similarly, the tendency of vehicles to move along the same routes have been exhibited in [71] and in [72]. Finally, road community finding has been used for efficient routing in vehicular environments [73]. For a survey of other social aspects in ad hoc networks, the reader may refer to [74].

4.1.1 Motivation and contributions

The technique of clustering is widely investigated in the context of mobile ad hoc networks [75], and in sensor networks [76]. For both types of networks, and for any kind of ad hoc network, it brings significant benefits that can be summarized as follows: a) alleviates the broadcast storm problem [77] which results in reduced congestion, and packet losses, b) decreases packet delivery latency, c) provides better spectrum utilization in time and space, d) allows for data aggregation, and e) increases network longevity. A search for clustering protocols for these two types of networks will reveal the existence of several hundreds of articles. Thus arises the question of whether VANETs need new clustering techniques. The answer is affirmative because VANETs are characterized by unlimited power, high but constrained (due to the road network) mobility, and human sociological factors.

Collectively, the existing proposals for vehicle clustering suffer from one or more of the following shortcomings: a) they are not generic enough to be used for both urban and highway scenarios, b) they are based on sophisticated and unpractical data mining procedures with many hard-to-set administrative parameters, c) they do not exploit the road network's and/or the VANET's topology at all, and d) they exploit at a very localized manner the "intention" of the mobility (i.e., speed, direction) which may present significant variations thus confusing the clustering protocols and making suboptimal clustering decisions which harm both the cluster stability and effectiveness.

The present work proposes a novel vehicle clustering protocol that avoids the aforementioned shortcomings and tries to incorporate the best features of the major vehicle clustering families. At the heart of the protocol are the social aspects of vehicles moving in a city or in highway; their tendency to follow the same routes because their drivers have some final destination in mind. Such sociological aspects have reported in several studies [69, 13, 71]. The implementation of this idea is based on the simple, solid mathematical theories.

In particular, the present research work develops two clustering policies, namely the *Sociological Pattern Clustering (SPC)* and its specialization, namely the *Route Stability Clustering (RSC)*. Statistical information gathered by Road-Side Units (RSUs) located in the boundaries of the area of interest are used in order to build the sociological profile of every vehicle. This profile is used in order to create clusters with neighbors that will (high probability) have similar behaviors. The research work presented in this chapter makes the following contributions:

- It exploits the "macroscopic" social behavior of vehicles, for the first time in the clustering literature.
- It combines this macroscopic behavior with "microscopic" behavior which is based on an earlier proposal by the authors under the concept of *virtual forces* [78] in order to create stable and balanced clusters.
- Based on this two-level behaviors, it develops the *Sociological Pattern Clustering (SPC)*, and the *Route Stability Clustering (RSC)* clustering protocols.
- It evaluates the performance of the proposed clustering techniques against the most popular clustering methods for VANETs. The evaluation is done for a large range of parameters and values:
  - for both urban and highways scenarios,
  - for different transmission ranges and vehicles speeds,
  - for varying social behaviors.

The rest of this chapter is organized as follows: Section 4.2 describes the network model. In Section 4.3 the semi-Markov model is described; Section 4.4 presents the I2V and V2I communication part of the scheme used in order to create the sociological profile of the vehicles; Section 4.5.1 describes the Sociological Pattern Clustering (*SPC*); Section 4.5.2 describes the Route Stability Clustering algorithm; Section 4.6 presents the simulation environment and results. Section 4.7 concludes the chapter.

4.2 Network model

4.2.1 Definition of the system

Definition 1 Let $S = \{S_1, S_2, ..., S_M\}$ represent the set of road segments in a given geographical area or map. A road segment S_i is represented by a unidirectional edge between two consecutive junctions.

Definition 2 Let $V = \{V_1, V_2, ..., V_N\}$ be the set of vehicles that are traveling in the given geographical area during a certain time period.

Definition 3 Let $TP = \{TP_1, TP_2, ..., TP_K\}$ be the set of Time Periods that the investigated system is segmented.

4.2.2 Road network communities - subnetworks

In the past few years, complex networks [13, 47, 69] have been studied across many fields of science. A number of network features have been discovered, among which the properties of hierarchical topology and community structure have attracted a great deal of interests recently. Communities are groups of vertices within which connections are dense but between which they are sparser. Networks often show a hierarchical structure of communities nested within each other. Identification of these communities can provide insight into better understanding and visualizing the structure of networks, and applications have ranged from technological networks to biological networks and social networks. In a road network where streets are mapped as edges and intersections as vertices, the intersections that locate close in a small region are more likely to form a community. The network is then decomposed, with adjacent subnetworks being loosely connected by the intergroup edges.

Many approaches focus more on how to partition and manage a network such that the number of boundary – border nodes for each subnetwork is uniform and minimized, the subnetworks are approximately of equal size, and so on.We use partitioning based on [73] where each subnetwork forms an isolated part, and different parts are connected together via intergroup arcs (arcs that are incident to/from boundary nodes and do not belong to any subnetwork). The city is then partitioned in small areas that can be investigated in isolation. These areas are called *subnetworks*. RSUs are assumed to be located at every entry/exit of each subnetwork in order to collect driving behavior of every vehicle that leaves the subnetwork and assign a social number when it enters the area based on previous historical data of the specific vehicle (Figure 4.1).

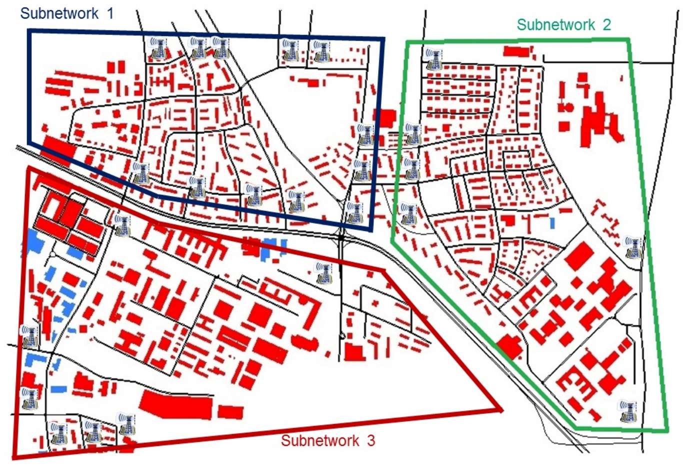


Figure 4.1: City is divided in subnetworks. RSUs are located in entries/exits of each subnetwork.

4.2.3 Collection of personalized data

As we described in the previous section,RSUs are assumed to be located at fixed locations at the borders of the region of interest. As vehicles move through the network, they record every road segment S_j they traverse in the order of arrival to those segments. Every second, each RSU will broadcast a short message ($DENM$) to all vehicles in its vicinity, which will query each vehicle to send their collected set of segments. Upon receipt, the vehicles will create a packet containing the partial path collected as well as other attributes. Each vehicle has a unique identifier V_i. More analytical description is at section 4.4.

Privacy preservation is critical for vehicles. In the vehicular context, privacy is achieved when two related goals are satisfied: untraceability and unlinkability [79]. First property states that vehicle's actions should not be traced and second that it must be impossible for an unauthorized entity to link a vehicle's identity with that of its driver/owner. On the other hand no traffic regulation or congestion avoidance can be achieved if this privacy protection is not removed. Data concerning owner's identity of a given vehicle and the path followed along a period of time are crucial for building that vehicle's social profile. Security mechanisms should prevent unauthorized disclosures of information, but applications should have enough data to work properly [80].

4.3 Mobility of nodes and semi-Markov Model

We model the mobility of vehicle i with a time homogeneous semi-Markov, with discrete time. The states are represented by the road segments.

The reason for using semi-Markov processes (rather than continuous-time Markov chains) for modeling user mobility is because the sojourn time during which a user is traveling along a road segment does not always follow the exponential distribution. A semi-Markov process allows for arbitrary distributed sojourn times and can be viewed as a process with an embedded Markov chain, where the embedded points are the time instants when a user travels along a road segment. A node that moves between two road segments, transitions in the Markov process between the corresponding states. We assume the transition probabilities between states have the Markov memoryless property, meaning that the probability of a node i transiting from state V_j^i to state V_{j+1}^i is independent of state V_{j-1}^i

$$A = \begin{pmatrix} 0 & a_{12} & 0 & a_{14} & 0 \\ 0 & 0 & a_{23} & a_{24} & 0 \\ 0 & 0 & 0 & 0 & a_{35} \\ 0 & 0 & a_{43} & 0 & a_{45} \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As a vehicle enters a state (road segment) j, it stays there for a time called state holding time T_{j,TP_k}, and then leaves to the next state j'. Note that this sojourn time does not include the time when the nodes are in transit between road segments. In order to avoid having absorbing states we perform wrap around, connecting each exit state with every entry state with connections that have equal probabilities.

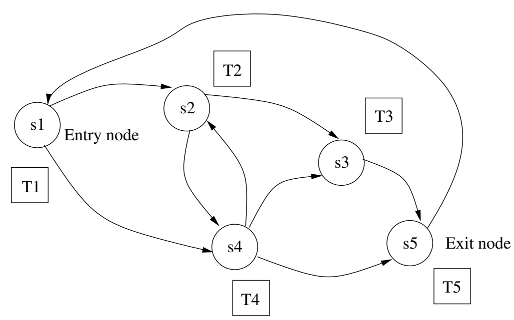


Figure 4.2: Semi-Markov model of vehicle v_i in a simple network topology with one entry and one exit.

The state holding time effectively depends on the vehicle speed and road condition (e.g. congestion) that we assume is constant in our application for every time period TP_k.

4.3.1 Transition probability matrix

A is the transition probability matrix of the embedded Markov chain for vehicle V_i for Time Period TP_k. Figure 4.2 shows an example transition probability matrix for vehicle V_i that moves around. At any of the road segments move to another road segment according to its preferred probability.

For example, if the node is at s_2, it can then

- move to the s_4 with probability a_{24}
- or stay in state S_2 for time T_{2,TP_k}
- or go to the s_3 with probability a_{23}

Those mobility probabilities constitute the transition probability matrix A. Note that each node has its own transition probability matrix that reflects its trajectory preference for the investigated time period TP_k.

4.3.2 Equilibrium probabilities

The equilibrium probabilities of the embedded Markov process can be calculated according to Equations 4.1 and 4.2

$$\pi_j = \sum_{i=0}^{M} \pi_i a_{ij}, \quad j \geq 0 \tag{4.1}$$

$$\sum_{j=0}^{M} \pi_j = 1 \tag{4.2}$$

If T_{j,TP_k} is the mean sojourn time at state j for investigated time period k then the equilibrium probability of the semi-Markov process at that state is calculated using the probabilities of the embedded process using Equation 4.3

$$p_j = \frac{\pi_j T_{j,TP_k}}{\sum_i \pi_i T_{i,TP_k}} \tag{4.3}$$

4.4 Communication phase

4.4.1 Vehicle leaving the subnetwork

As vehicle V_i leaves the region of interest, goes into the control range of the intersection, RSU device near the boundary of the control range impels this vehicle to send the collected set of segments, also known as partial path PP_i and travel time T_j for each road segment j traversed also known as partial time path PT_i. Upon receipt, the vehicles will create a packet containing the partial time path collected as well as other attributes as shown in Figure 4.3.

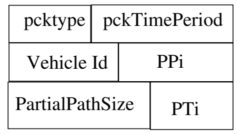


Figure 4.3: Vehicle packet design.

Each RSU has a database that contains all the partial paths that the vehicles traversed in the investigated area. The database consists of a separate table PPT_{ik} for every vehicle i and for every Time Period k. That means that for a single vehicle there may be several tables one for each Time Period according to the segmentation of time.

At the RSU side, received collected partial paths will be added to the RSUs' databases according to the nature of the received packet. If the received packet contains a vehicle ID that does not exist in the RSU database, the RSU will create a new table for this vehicle ID. However if the vehicle ID in the packet already exists, the partial path is appended to the existing vehicle table PPT_{ik}. Every β seconds, the RSUs can provide each other with the information collected.

The investigation time is segmented to time periods TP. It has been observed [81] that in practice, weekdays and weekends usually exhibit significantly different traffic conditions while it is shown that all weekdays (weekends) have similar congested and congestion-free traffic patterns; therefore we group the days and treat different groups separately. The time periods are pre-dawn: (up until 8AM), morning rush-hour (8-10AM), late morning (10-noon), early afternoon (noon-4), evening rush-hour (4-7PM), and night time (after 7PM). The Paths table PPT_{ij}, shown in Table 4.1, represents the set of vehicles movement patterns during their trips in that monitored area. As shown below, each vehicle V_i will have a table in this database describing its movement paths for each Time Period. RSU uses this data to compute the mean holding time T_{S_j,TP_k} for each time road segment S_j and for each time period TP_k (see subsection 4.4.3).

V_{ID}	*Partial Paths*
V_1	$[S_1, S_2, S_3, S_5]$
V_1	$[S_1, S_2, S_4, S_4]$
V_1	$[S_1, S_4, S_3, S_5]$

Table 4.1: Path table of vehicle i for time period j.

According to these paths the transition probabilities of table A are updated for vehicle V_i for the Time Period TP that the vehicle entered the region of interest. That way every vehicle has a unique semi-Markov model for every time period called SM_{V_i,TP_k}.

4.4.2 Vehicle entering the subnetwork

As vehicle V_i enters the region of interest, goes into the control range of the intersection, RSU device near the boundary of the control range sends to the vehicle data ($DENM$ message) that is extracted from its semi-Markov model for the specific time period SM_{V_i,TP_k}. Since all the computations take place off the vehicle and on the RSU the information that the vehicle gets from the RSU is limited. This information is either the social number SN of the vehicle or the Route stability Number RN according to the clustering method that the vehicle is going to follow. If the vehicle enters the subnetwork for the first time then RSU assigns to the vehicle a social number SN or a Route stability Number RN metric which is the mean value for the specific Time Period. This data is just one single floating number that is embedded in a simple beacon message.

4.4.3 Mean sojourn times

Each vehicle packet that is transmitted to the RSU contains the time that the vehicle spent on each road segment traveled for the specific time period that it was in the subnetwork. These partial time paths (PT_i) are used in order to calculate mean travel time for each road segment j and for each time period k. For each new message new values are added to the table RST. One new row is added for each distinct road segment, time period and travel time as shown in Table 4.2.

Road Segment j	*Travel Time*	*Time Period k*
S_j	T_1	TP_k
S'_j	T_2	TP'_k
S''_j	T_3	TP''_k

Table 4.2: Road segment travel times table RST.

Mean travel time for a specific road segment j for a time period k is the sum of all values of the second column from Table 4.2 according to Equation 4.4.

$$T_{S_j,TP_k} = \frac{\sum_i RST(i,2)}{L}, \quad RST(i,1) = s_j \quad and \quad RST(i,3) = TP_k \tag{4.4}$$

, where L is the number of rows of Table 4.2 that satisfy the above constraints.

4.4.4 Sociological patterns

As it was described earlier as a vehicle V_i leaves the subnetwork informs the RSU in range, and through it the central database, about the path it followed during its stay in the area.

This path in inserted in the table PPT_{ik} that contains all the paths of the vehicle for the specific time period.

Using these paths, the transition table of Vehicle is created and from it the social patterns of the vehicle are extracted (Figure 4.4). As shown in Figure 4.4 the Social Pattern in created by starting at each entry segment in the network and by following the most possible next transition of the transition table of vehicle V_i for the specific time period.

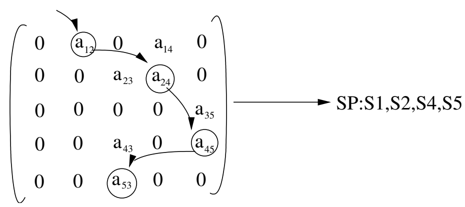


Figure 4.4: Transition table → social pattern.

Once the social patterns are extracted, a table SP_k than contains all the social patterns for the specific time period in updated:

- If V_i has previous Social patterns then these values are deleted from SP_k.

- The Social Pattern is compared to all the social patterns that exist in the central database and in the table for the specific Time Period. If the Social Pattern already exists in the database in the Table SP_k the Vehicle ID is appended in the corresponding line. If this social pattern is new, the social Pattern is appended to the table and a new number is assigned to it.

This procedure is represented in Figure 4.5.

After the end of this procedure several social patterns for each vehicle V_i are created, depending on the road segment that the vehicle used in order to enter the subnetwork and the corresponding time period. Every social pattern is matched to a unique social number SN. It is important to note here that a vehicle may have more than one social numbers, in order to better represent different social behavior of the same vehicle/driver. This behavior is related to time, representing driving to work for the morning and hobbies in the evening and to the entry point in the subnetwork, which probably means a different final location. The next time that the vehicle V_I reenters the subnetwork the SN value that best matches current time and entry point to the subnetwork is assigned to it. This number is used in order to perform clustering and create groups with vehicles that share common habits and behaviors.

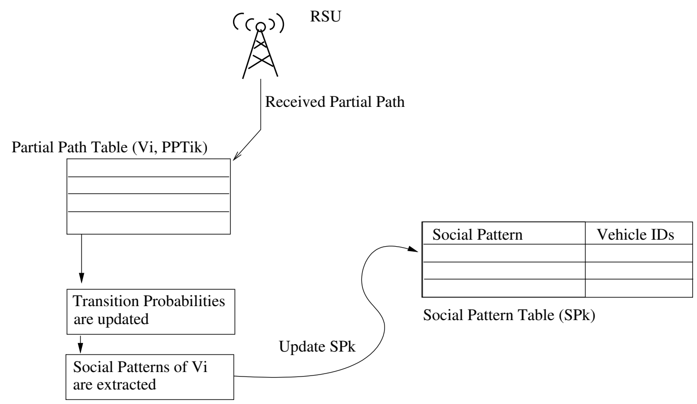


Figure 4.5: Procedure for social patterns extraction.

4.5 Social Clustering

In order to create clusters the basic mechanism of Virtual Forces Vehicular Clustering ($VFVC$) [82, 78, 83] is used.

When all nodes first enter the network, they are in non-clustered state. Nodes having higher number of positive neighbors in terms of relative force $Frel_{ij}$, maintaining closer distances to their neighbors, are qualified to be elected as cluster-heads. In the initial method [83] a lane detection algorithm is used to determine the lane the vehicle moves on. According to the lane being a turning or a non turning one the method favor vehicles that follow a non turning lane to become clusterheads. This method produces stable clusters when focusing on what happens on a central road, where cars enter and leave it all the time. In a more realistic scenario, a large area of a city, the long lifetime of clusters should not be limited on main roads but all clusters created in the city must be as stable as possible.

In order to create stable clusters we use the social behavior of vehicles based on historical data collected from RSU's that scattered along the borders of the area of interest. In order to incorporates the social behavior of the vehicles when move in Urban environments we incorporate in every beacon message one additional byte of information about the Social Pattern - flow (SN) the vehicle has. When for specific applications we are interested to create stable clusters along a central road we introduce a new metric called Route stability Number (RN) as described in section 4.5.2.

4.5.1 Sociological pattern of v_i

The first step in creating a cluster for every vehicle is to identify its neighbors. Neighborhood identification is the process whereby a vehicle/node identifies its current neighbors within its transmission range. For a particular vehicle, any other vehicle that is within its radio transmission range is called a neighbor. The neighbor set is always changing since all nodes are moving. Every moving node keeps track of all current neighbors (their id's) the current and the past distance. In order to perform clustering using social criteria, SPC maintains two different sets of neighbors. Set N_i is the set of all neighbors in range of vehicle V_i and set NS_i is the set of all neighbors that share a common social pattern.

The clustering procedure consists of two stages.

First stage of clustering

In the first stage each vehicle try to create cluster with nodes that have the same SN according to these rules:

- At any time each vehicle i recalculates total F and according to total non-clustered members with same SN within range, try to form a cluster and become the clusterhead.

- If the node has the biggest positive force applied to it and there exist at least one free node in its neighborhood NS_i, it declares itself to be a clusterhead.

- In the opposite situation, where there exist a free node j with biggest total F in range the vehicle becomes a cluster member of j.

This algorithm leads to the formation of clusters which are at most two hops in diameter and have same SN.

Second stage of clustering

After the initial clustering phase, some clusters will have been formed. However, there will also be nodes which could not join any clusters during the initial clustering phase mainly because they are surrounded by vehicles having different social patterns. In this occasion clustering is performed again using total Force applied to each vehicle. At this stage the set N_i of all neighbors is used and clusters of vehicles having different Social Patterns are created.

Figure 4.6 represents the different states of a vehicle (Undefined, Free, Member, Cluster Head), and the transitions among these states when the vehicle enters, moves around or leaves the subnetwork for the sociological pattern clustering (SPC) method. When the vehicle first enters subnetwork or leaves it, its state is undefined UN since every region is studied in isolation.

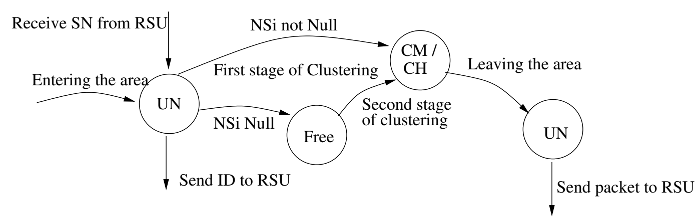


Figure 4.6: States of a vehicle.

4.5.2 Route stability of vehicle v_i

When dealing with main roads of a city, the creation of clusters is mostly binded to safety issues. A big family of transport applications uses the dissemination of traffic data or security data in a limited area [84] e.g. city block, main road etc. In order to create stable clusters on main streets, vehicles that tend to stay more time on the street are better candidates to be clusterheads (figure 4.7). If a vehicle that is going to leave the street soon, is elected as a clusterhead then major re-clustering is going to take place when it turns to another road segment, since it leaves all of its members orphans. In case a member node leaves the street in order to follow another edge of the network, only this vehicle tries to find a nearby cluster to enter.

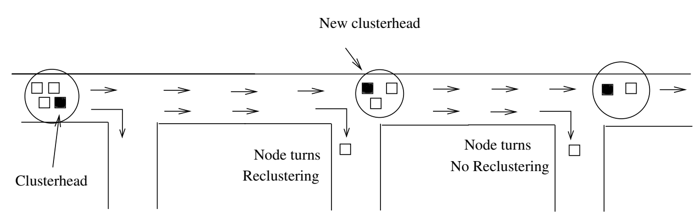


Figure 4.7: The correct choice of the clusterhead on main streets plays a significant role.

The Route Stability Clustering method uses long term probabilities of vehicles in order to choose clusterheads. Vehicles exchange beacon messages, that contains information about node Identifier (V_{id}), node location, speed vector in terms of relative motion across the axes of x and y (dx, dy), Route Stability Number RN, state and time stamp.

Route stability Number (Reliability) of Vehicle V_i that moves on road segment S_i the time period TP_k is calculated be Equation 4.5:

$$RN_{V_i} = \sum_j p_j, \quad j \in PP_i \tag{4.5}$$

where PP_i is the set of the road segments that belong to the same street and p_j are the long term probabilities of vehicle V_i that have been received by the RSU the specific time period.

From equation 4.5 it is evident that this RN metric represents the accumulative probability of each vehicle to stay on the road segments that constitute the main street. Route stability number RN is then incorporated in Equation 3.1 as parameter qi, in order to favor vehicles, that are more possible to stay on the street for the longer time, to become clusterheads (see section 4.6.2. In a multi-lane street vehicles that have bigger route stability number RN are better candidates to become clustrheads since they are going to stay longer on the street, based on the historical data of the vehicle.

4.5.3 Overhead due to clustering

In order to perform clustering, vehicles exchange simple CAM messages. Each beacon message consists of node Identifier (Vid), node location, speed vector, total force F, state, (RN or SN metric according to the method used) and time stamp. CAM messages are send every second in order to maintain up to date information about the neighborhood. Relative mobility, which is used in order to perform cluster formation, is calculated at every vehicle in isolation, using current and possible future position of every neighbor based on previous received beacons. Moreover, the clustering-specific messages are exchanged via the control channel (IEEE 802.11p) and this does not affect the dissemination of data.

When vehicles approach an exit of the subnetwork, entering the control range of a RSU they send a dedicated packet to the RSU that contains its path table. Since every vehicle leaves the subnetwork only once the overhead due to this communication is very limited (see. Figure 4.1).

4.6 Simulation and performance evaluation

This section evaluates the performance of SPC and RSC. The traffic simulation is conducted with SUMO [63] and the trace files are injected to our custom simulator in order to perform clustering. In the simulation, we use the road network of city of Erlangen. Using the hierarchical communities method we are able to divide the city in isolated regions and study the mobility of vehicles in subnetwork 2 (see Figure 4.1). The only communications paths available are via the ad-hoc network and there is no other communication infrastructure. The power of the antenna is $P_{tx} = 18$dBm and the communication frequency f is 5.9 Ghz.

The reliable communication range of the vehicles is calculated according to Table 4.3. The reliable communication range is calculated for every pair of nodes at every instance based on the diffraction caused by obstructing vehicles [78]. In our simulations, we use a

Data Rate (Mb/sec)	*Minimum Sensitivity(dBm)*
12	-77
18	-70
24	-69
27	-67

Table 4.3: Minimum sensitivity in receiver antenna according to data rate.

minimum sensitivity (P_{th}) of -69dBm to -85db which gives a transmission range of 130 to 300 meters.

According to [85] an acceptable communication range for VSC applications, which use the same broadcast messages to our clustering methods, is about 300 m. This range that can be achieved by low transmission power, like the one we use in our simulations, is enough for correct dissemination of a message in a neighborhood while it improves spatial reuse in heavy traffic. In rural environments, in scenarios with low data rate (3MBPs) authors in [85] showed that Packet Delivery Ratio (PDR) of 60 % can be achieved for such medium distances.

All the simulation parameters with their default values are represented in Table 4.4.

All nodes are equipped with GPS receivers and On-Board Units (OBU). Location information of all vehicles/nodes, needed for the clustering algorithm is collected with the help of GPS receivers. By default, 80 to 160 vehicles move in the network, and their movement pattern is determined by Krauss following model. The vehicles have maximum velocities from 40 to 50km/h, large speed deviation (60 to 140 % of legal speed limits) with 2 to 4 different flows – social profiles.

While one would like to have deterministic social profiles for every driver - vehicle , this is not possible due to the nature of driving. Even if a driver follows a standard route every day, it is still possible for the driver to deviate from it once in a while. Circumstances like a doctor appointment, road construction or alternative route due to congestion may cause driver to change the route he is predicted to follow according to his social profile. All of this points to the fact that the prediction of driver intent must be probabilistic. For these reasons vehicles are injected to the map at random sequence and follow their path according to their social profile with default probability of 67 % and range from 67 % to 97 % (see Figure 4.13).

In order to incorporate different characteristics in the method we have assigned values to parameters q (see equation 3.14) according to table 4.5. These parameters represent a special role that a vehicle may have in the network due to its mobility behavior or physical characteristics. Parameter q_R is valid only for the RSC method and it represents the Route

Independent parameter	*Range of values*	*Default value*
$Velocity\ (m\backslash sec)$	20, 50	42
$Number\ of\ Vehicles$	80,120,160	120
$Probability\ of\ following\ the\ social\ pattern\ (\%)$	67,97	67
$Number\ of\ Sociological\ patterns$	2,3,4	2
$Communication\ Range\ (m)$	130 - 300	130
$Number\ of\ RSUs$	6	6
$Subnetwork\ of\ theCity$	1-3	2

Table 4.4: Simulation parameters.

stability number of the vehicle.

parameter	*Simulated*	*Parameter value*
$Predefined\ routes\ (Q_p)$	No	1 (default)
$Vehicle's\ Height\ (Q_T)$	Yes	2(Tall), 1(Short)
$Vehicle's\ Route\ stability\ (Q_R)$	Yes	2(High), 1 (Medium), 0.5(Low)
$Driver\ behavior\ (Q_b)$	No	1 (default)

Table 4.5: Parameters of Clustering methods

To show the performance of our proposed Social clustering (SPC, RSC) methods, we compare them with the Lowest-id ($Low-id$), Dynamic Doppler Value Clustering ($DDVC$) and Mobility Prediction-Based Clustering ($MPBC$) proposed in [40], [86] and [87] respectively. The Lowest-ID algorithm forms of clusters which are at most two hops in diameter.

The basic concepts of Lowest id are the following. Each node is given a distinct ID and it periodically broadcasts the list of its neighbors (including itself). A node which only hears nodes with ID higher than itself is a "clusterhead" (CH). The Lowest-ID node that a node hears is its clusterhead, unless the Lowest-ID specifically gives up its role as a clusterhead (deferring to a yet lower ID node). A node which can hear two or more clusterheads is a "gateway". Otherwise, a node is a free node.

In $DDVC$ a cost metric derived from the Doppler shift property, the Doppler Value, is used in order to create clusters. The Doppler Value (DV) is related to the relative velocity. We simulate DDVC with parameter n_{min} having value 1.

The basic information in $MPBC$ is the relative speeds estimation for each node. In clustering stage, nodes broadcast Hello packets periodically to build their neighbors lists. Each node estimates its average relative speeds with respect to its neighbors based on these Hello packets exchanges. Nodes with lowest relative mobility are selected as CHs. During cluster maintenance stage a prediction-based method is used to solve the problems caused by relative node movements.

4.6.1 Sociological Pattern Clustering

As we mentioned in subsection 4.2.2, after splitting the city in subnetworks, these regions can be investigated in isolation. Using the map from Figure 4.1 we simulate the performance of the methods in subnetwork 2. In order to evaluate the stability of the algorithm, we measure the stability of the cluster configuration against vehicle mobility. In a high dynamic VANET, nodes keep joining and leaving clusters along their travel route. Good clustering algorithms should be designed to minimize the number of cluster changes of the vehicle by minimizing reclustering. This transitions among clusters are measured in order to evaluate the performance of the algorithm. The basic transition events the vehicle encounters during its lifetime:

- A vehicle leaves its cluster and forms a new one (becomes a clusterhead).
- A vehicle leaves its cluster (due to communication range) and joins a nearby cluster or becomes free.
- A cluster-head merges with a nearby more stable cluster.

The average cluster lifetime is another important metric that shows the performance of the clustering algorithm. The cluster lifetime is directly related to the lifetime of its cluster-head. The cluster-head lifetime is defined as the time period from the moment when a vehicle becomes a cluster-head to the time when it is merged with a nearby cluster.

SPC versus VFVC

Initial Spring Clustering and the enhanced $VFVC$ method behave well when investigating highways where vehicles don't change direction so often as in a real urban environment (the former), or when we are focused on main streets when we care about the stability of the cluster on the street and not at the whole area (the latter). In addition $VFVC$ gives good outcome when the road lanes clearly clarify the possible direction of the car that is traveling on the road. In a more realistic scenario when small road segments of a city consist of one lane, $VFVC$ degrades to the initial Spring Clustering. As shown in Figure 4.8 the performance of $VFVC$ compared to SPC in a city region when most of the roads consist of one lane is much lower but still better than the performance of $DDVC$ since except from relative speed that $DDVC$ uses to perform clustering, $VFVC$ uses virtual forces among nodes that are affected by relative mobility, current and future distance in both x and y axes. $MPBC$ performs better than the $VFVC$ method since it is based on the estimated mobility information of nodes. In urban environments, where the mobility of nodes compared to a highway is more dynamic, SPC has clear impact on cluster formation and stability (Figure 4.8).

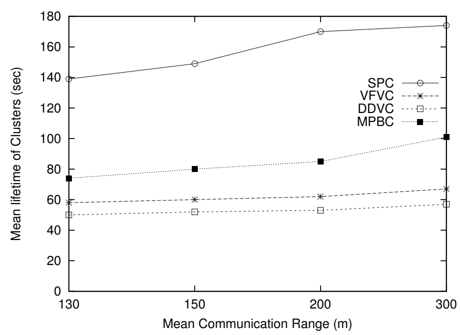


Figure 4.8: Lifetime of SPC versus $VFVC$ for a typical urban scenario [2 flows, 70 % probability of following the social pattern] for different communication ranges.

SPC versus Low-Id, DDVC and MPBC

We compare the performance of SPC, $Low - Id$, $DDVC$ and $MPBC$ methods when different transmission ranges (Figure 4.10) and different speeds (Figure 4.11) are used. We also investigate the performance of SPC against different number of Social Patterns that the vehicles have (Figure 4.12) and against different probabilities in following the correct pattern (Figure 4.13).

Number of clusters over time. The number of clusters created by a clustering algorithm is a significant parameter of the clustering procedure; too many, and thus small, clusters implies that the benefits reaped due to clustering will diminish, since the broadcast storm is not really cured, and too much communication has to take place to forward messages (too many clusterheads and too many gateways participate in the forwarding process). On the other hand, the existence of only a few, and thus quite large, clusters is also not desirable since the channel is shared among too many members of the same cluster and hence the communication latency increases. We present an experiment – with the default values of the parameters of Table 7.2, and the results are illustrated in Figure 4.9. It depicts the total number of clusters created by the competing methods over the simulation time of the experiment. We see that SPC creates a moderate number of clusters, less than that created by $Low - Id$, but more than those created by $DDVC$ and $MPBC$ for most of the time. Analogous observations were made for other values of the parameters. Therefore, SPC

can achieve the best of two worlds; relatively small transmission latency, and relatively few rebroadcast messages.

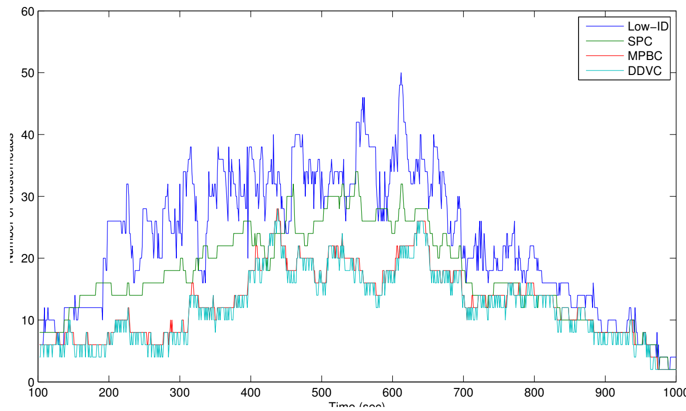


Figure 4.9: Number of heads produced by all methods during the simulation.

In order to investigate how stable the clusters created by each method are, we measure cluster lifetime and mean transitions that each vehicle encounters during the simulation. We tune different parameter each time and we can see, from the sections that follow, that the average transitions produced by our SPC technique is smaller compared to that produced by the $Low - Id$ and gives comparative numbers to $DDVC$ and $MPBC$.

Cluster stability versus communication range. Figure 4.10 shows that the average transitions of the vehicle decreases and mean cluster lifetime increases as the transmission range increases when SPC is used. This is because increasing the transmission range, increases the probability that a vehicle stay connected with its cluster-head. Communication range doesn't have any impact on $Low - Id$'s performance and though it slightly improves $DDVC$ it has a major impact on SPC stability. Since SPC creates clusters of nodes sharing common social profiles, as communication range increases the probability that such nodes stay interconnected for a longer time also increases. In $Low - Id$ only vehicle's ID is used in order to elect clusterheads and that way although increased communication range may have a positive impact on nodes connectivity it also affect them in a negative way since nodes more often find a neighbor with lowest id and perform reclustering. In $DDVC$ increase of communication range does not have so big positive impact since in an Urban environment vehicles always change directions, accelerate and deaccelerate in order to follow different road segments thus causing the method to create new clusters often. $MPBC$ achieve longer average CH lifetime compared to $Low - id$ and $DDVC$ since the method is designed for randomly and independently moving nodes, but its performance is still worse than the proposed method which incorporates drivers social profile.

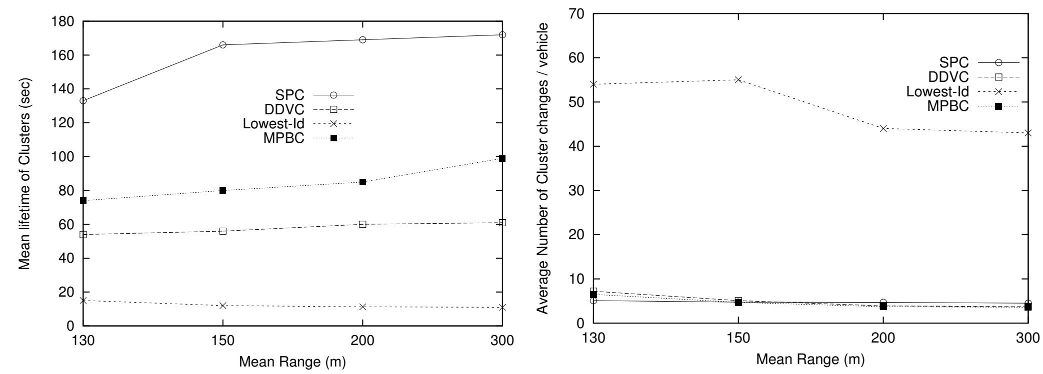


Figure 4.10: Lifetime and mean cluster changes versus range.

Cluster stability versus speed. In Figure 4.11 we observe that the impact of different vehicle speeds in a urban environment is not so clear. This is due to the fact that in urban areas the max velocity cannot be easily reached by vehicles since they always have to stop at intersections or change speed due to turns and congestion. For the maximum speeds investigated SPC has much better performance compared to $MPBC$, $DDVC$ and $Low-Id$.

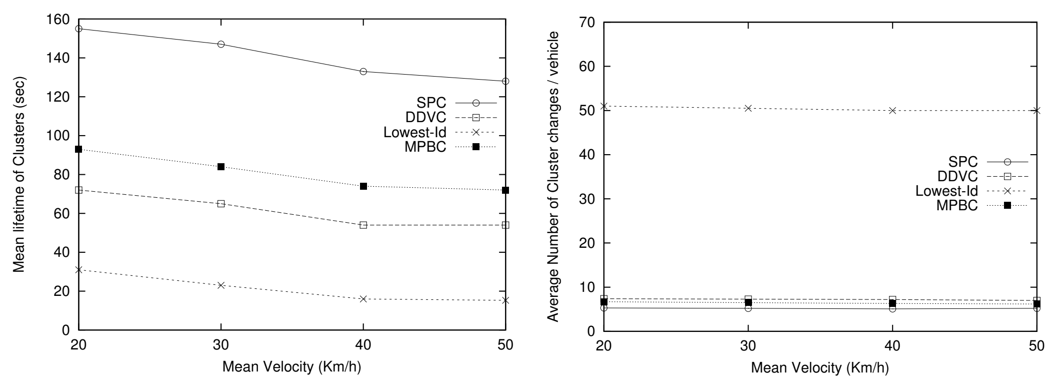


Figure 4.11: Lifetime and mean cluster changes versus speed.

Cluster stability versus social patterns. As social patterns increase, meaning that vehicles follow less common routes, the performance of SPC decreases (see Figure 4.12). From this figure, it could be seen that SPC follows the theoretical model closely, yet the actual cluster lifetime is a always better than the other methods give. $DDVC$ and $MPBC$ also degrade as the mobility of vehicles become more chaotic.

Cluster stability versus pattern following probability. The mean lifetime that our method produces, even when the probabilities that a car follows its Social Pattern drops to

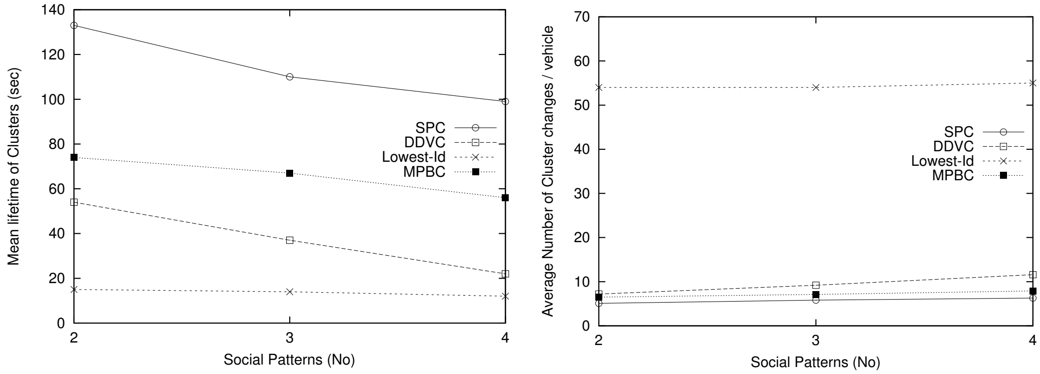


Figure 4.12: Lifetime and mean cluster changes versus number of social patterns.

67 % (Figure 4.13), is always better than those that the other methods give. All methods when the probabilities raise, show better performance in terms of mean cluster lifetime, since the mobility of vehicles becomes less chaotic. As vehicles tend to use the same routes, clusters can more easily maintain their current structure and all clustering methods perform better. *SPC* having information about the social pattern of vehicles still achieves the best outcome, increasing rather decreasing the performance gap to the competing methods.

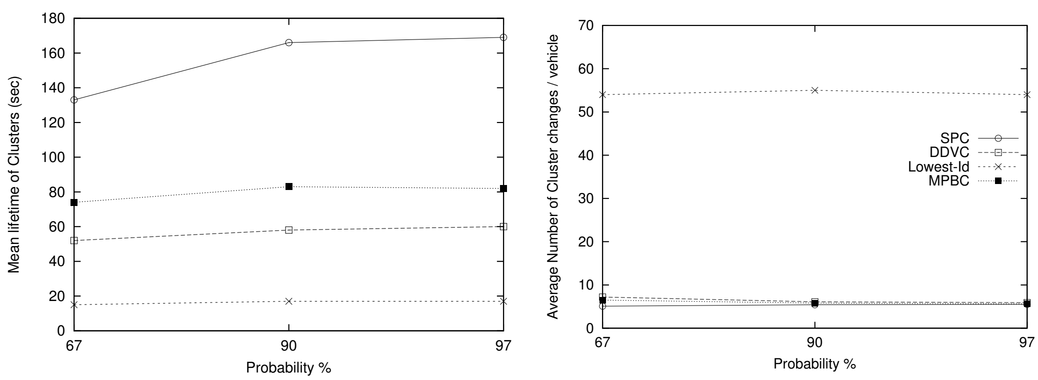


Figure 4.13: Lifetime and mean cluster changes of *SPC* versus probability of following a social pattern.

4.6.2 Route Stability Clustering.

In order to evaluate the performance of *RSC* we are interested in a main street of the area of Erlangen. The street is shown in Figure 4.14. The area consists of many intersections. On the map three main flows of vehicles are shown. These flows split the traffic of the main road of interest.

Figure 4.14: Main road and the flows that split traffic.

We focused only on one traffic direction. Vehicles follow three different route distributions according to Table 4.6. These distributions are used in order to represent the social patterns of the vehicles and are based on their historical data.

We follow vehicles until they leave the section of the road turning left or right. In that way we are focusing on what happens on a central road, where cars enter and leave it all the time if we favor cars that follow the non turning lane to become clusterhead. Using Equation 4.5 and the data from Table 4.6 we calculate the route stability RN of each vehicle regarding the street of interest. Route stability is then incorporated in Equation 3.1 as parameter q_i, in order to favor vehicles that are more possible to stay on the street for the longer time become clusterheads.

Route	*Probability to follow the route*	*Stability* R_{V_i}	*Parameter* q_i
1	90%	Low	0.5
2	90%	High	2
3	90%	Medium	1

Table 4.6: Route distributions according to different social patterns of vehicle i.

For the scenario of RSC we used the values of Table 7.2 in terms of velocity and communication range. We compare the performance of RSC, SPC ,$Low-Id$, $DDVC$ and

MPBC methods and the results are presented in Figure 4.15.

The results of simulations conducted show that *RSC* algorithm outperforms the other investigated methods, in terms of average cluster lifetime (higher) which is translated in increased cluster stability, lower percentage of orphan nodes and larger cluster sizes. The other parameters that determine the stability of a clustering method like cluster-head changes, total number of clusters and null nodes also give better values for *RSC* compared to the other methods.

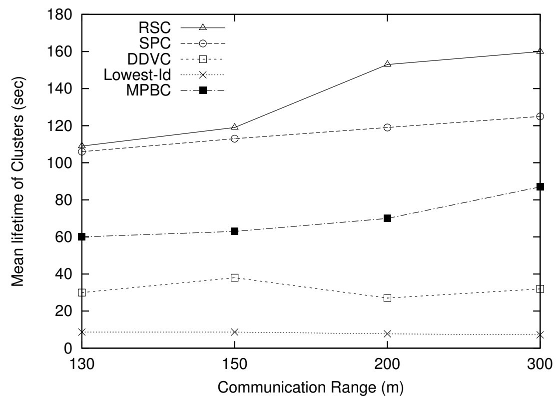


Figure 4.15: Lifetime of *RSC*.

4.6.3 Short term trajectory prediction mechanism

RSC and *SPC* methods are based on long term trajectory prediction of vehicles, based on historical data. Short term prediction, where the next road segment that the vehicle will follow is also useful for smart routing and clustering protocols. A new short term prediction mechanism for Vanets is developed, namely Rich Dictionary Markov - *RDM*.

RDM is based on two prediction mechanisms used simultaneously and complementary to each other. The first is similar to that in [2] exploiting the blending of models of various orders to calculate the probability of each state. These probabilities are used in conjunction with the second mechanism which is based on trie descent to make predictions. The first mechanism uses a blending strategy known as exclusion to calculate the probability of each state occurring in the input sequence and predict the one with the highest probability as the most likely next state. The second mechanism is guided by the phrase ”w” (Table 1) in order to traverse the trie and pinpoint to the current state of the predictor. Then in order to predict the next state it exploits the information (history) given from the higher order model, if there is any, and predicts according to specific rules. Figure 4.16 illustrates the different tries that LZ78, ALZ and RDM are building, by superimposing them into a single trie for the input sequence ”AAABABBBBBAABCCDDCBAAAA”. We see that RDM resultant dictionary is richer and the associated trie larger both in span and depth.

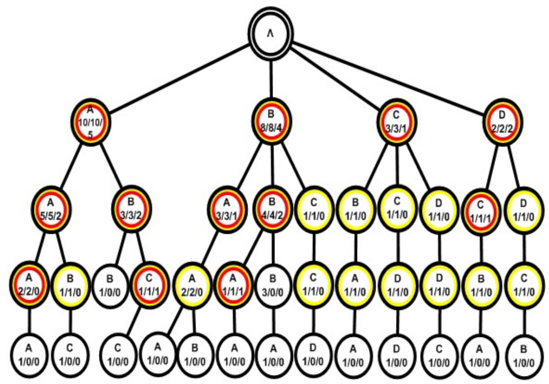


Figure 4.16: The superimposed tries of the input sequence ”AAABABBBBBAABCCDD-CBAAAA” are distinguished by the different node colour, Black (RDM) / Yellow (ALZ) / Red (LZ78), and in the same order, the numbers below each respective symbol represent the cardinality of it.

We examined the performance of the competing algorithms(RDM, $LZ78$ [88], Active LeZi-ALZ [89]), both in terms of prediction accuracy and efficiency (processing time and memory consumption) and the results are depicted in Figures 4.17. The dataset used for the evaluation, was downloaded from Project Gutenberg. It is an English translation of the novel ”The Three Musketeers” by Alexandre Dumas; it consists of more than 50 symbols. Figure 4.17 shows that RDM (ignoring the cold start) achieves prediction accuracy always larger than 30%. The mean accuracy is about 38%. In contrast, its competitors never go beyond 17% at the best case, and on the average their performance is around 11% for LZ78 and 13.4% for ALZ.

4.7 Conclusions

Vehicular networks can bring great benefit on driving safety, traffic regulation, infotainment, and in many other practical applications. These applications require effective and efficient packet exchange between vehicles, which is a very challenging problem. VANETs, especially in urban environments, a node may have up to 100 neighbors (the radio range of the IEEE 802.11p may reach up to 1 km and the density of vehicles may reach more than 100 vehicles per kilometer) which situation will cause severe wireless network congestion, causing packet collisions and thus losses, which leads to bandwidth and CPU resources waste. Vehicle clustering is an established technique to alleviate this broadcast storm problem.

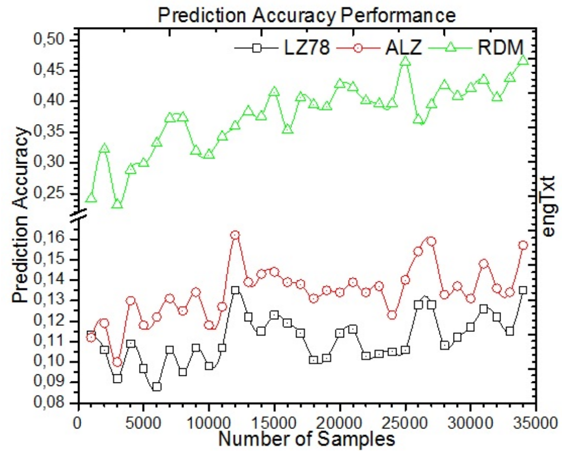


Figure 4.17: Prediction accuracy of the competing algorithms for the engTxt dataset.

Though drivers tend to follow the same or similar routes, social behavior of vehicles moving in a city are ignored from previous clustering methods. To the best of our knowledge, this work is the first one that uses macroscopic information from vehicles' history in order to create trajectory-based schemes for clustering of vehicles in VANETs. This information is combined with the microscopic information that vehicles exchange through periodic V2V messages, such as velocities, current and future positions and vehicle's physical characteristics (vehicle's height) making the proposed methods robust to the dynamic mobility that vehicles exhibit in an urban environment.

The proposed methods, namely *Sociological Pattern Clustering* (SPC), and the *Route Stability Clustering* (RSC) use historical data of each vehicle modeling them as semi-Markov processes, in order to extract the social patterns and create stable clusters. SPC assigns in every vehicle a social number SN, which represents the social pattern that this vehicle is probable to follow for the specific time period, and groups vehicles that have similar behavior. RSC, which focuses on creating stable groups on a highway-type street, calculates the long term probabilities of each vehicle and assigns to them a stability value. All the pattern extraction calculations are performed on a central server. The proposed social clustering methods are compared to $Low - Id$ [40], *Dynamic Doppler Value Clustering* [86] and $MPBC$ [87] clustering methods. The former is a typical topology-agnostic clustering method, and the latter is an high-performance mobility-based technique that uses relative speeds of nodes in order to create clusters. The obtained simulation results demonstrate the effectiveness of SPC and RSC against their competitors relative to the cluster stability and cluster size.

We have also presented a short term prediction mechanism that is based on two prediction mechanisms used simultaneously and complementary to each other in order to predict the next road segment that a vehicle will move. This short term prediction mechanism can be used along with the the sociological patterns of vehicles in order to increase prediction accuracy and cluster stability.

The proposed clustering methods can be used as a primitive in the design of networking routing protocols. The created clusters divide the initial graph to smaller networks. Then the routing algorithm can forward packets to the destination cluster, thus effectively reducing the long paths. Clusters can play the role of attractors, that attract the packets destined for them and disallow the packets to leave the clusters improving the delay performance of the network.

Chapter 5
Delay Efficient Backpressure routing protocols*

Packet scheduling/routing in wireless ad hoc networks is a fundamental problem for ad hoc networking. Backpressure routing is a solid and throughput optimal policy for such networks, but suffers from increased delays. In this chapter, we present two holistic approaches to improve upon the delay problems of backpressure-type algorithms. We develop two scheduling policies, namely Voting backpressure and Layered backpressure routing, which are throughput optimal. We experimentally compare the proposed policies against state-of-the-art delay-aware backpressure policies, which provide optimal throughput, for different payloads and network topologies. The experimental evaluation of the proposed delay reduction methods attested their superiority in terms of QoS, robustness, low computational complexity and simplicity.

5.1 Introduction

Wireless ad hoc (multi-hop) networks lack fixed infrastructure (e.g., base stations, mobile switching centers); the communication between any two nodes that are out of one another's transmission range is achieved through intermediate nodes; these intermediate nodes relay messages to set up a channel. Contemporary application areas of the ad hoc networks include modern battlefields, disaster relief, precision agriculture, e-health, ocean monitoring with underwater wireless sensor networks. Packet transmission scheduling in this type of networks is a fundamental issue since it is directly related to the achievement of a prescribed Quality of Service (QoS) and a minimum use of system resources. QoS is usually measured in terms of the average packet delay, transmission rate and maximum delay, and the main system resource to be saved is the nodes' energy so as to prolong network lifetime. In addition to delay and energy optimization, any packet scheduling/routing algorithm for ad hoc networks must be resilient to topology changes and strive for throughput optimality.

The development of a throughput-optimal routing algorithm for packet radio networks which is also robust to topology changes was first presented in [90]; it is based on the Lyapunov drift theory, and it is known as the *backpressure* ($\mathcal{BP}$) packet scheduling algorithm. Subsequently, the original concept spawns several lines of research in the topic.

*The ideas presented in this chapter appear in the following publications [C.02, S.03]

The performance of backpressure deteriorates in conditions of low, and even of moderate, load in the network, since the packets "circulate" in the network, i.e., the backpressure algorithm stabilizes the system using all possible paths throughout the network. The net effect of this mechanism is to increase delay and also to increase the energy consumption of the nodes that play the role of relays. Delay and energy are interconnected; the minimization of the average time that the packets stay in the system, implies a reduction in the average number of hops that the packets travel until they reach their destination, which in turn implies a reduction in the total energy consumption. Delay and energy consumption problems are particularly significant for ad hoc sensor networks. Routing-loop formation is another drawback of backpressure routing. In many real time applications like voice and video, high end-to-end packet delay is unacceptable. Often in such applications, a packet received with high delay is no better than packet loss. We could prevent high end-to-end delay in backpressure routing by not forwarding the packets on longer paths. But we still want to maintain the sufficient routes for any source-destination pair to provide adequate load balancing in case of high traffic load. Generally, these two objectives conflict with each other because few short routes exist in the network.

To circumvent the delay problems of backpressure, the *mean resource routing* algorithm [91] forces the links to stay inactive in order to lead the network to work in a burst mode, since for periods where the load of the network is low or moderate, link activation is prevented by a parameter M, leading to a delay increase. On the other hand, the relatively high computational cost incurred at each node by backpressure (maintenance of a queue for each possible destination, and update of these queues at each new arrival) inspired approaches based on *node grouping* in order to reduce the number of these queues [92] and thus the computational overhead, and as a side-product, reduce the delay. To alleviate the delay problems of backpressure, scheduling based on the *combination of backpressure and shortest-paths* have been proposed [93], which though demands an excessive number of queues and repeated calculation of all-node-pairs distances in case of topology changes. In summary, all these approaches either impose unpractical and non scalable assumptions, or are not very efficient. Finally, some recent ideas [94, 95] could be incorporated in an *orthogonal way* to improve analogously the delay performance of all policies. Here, we take a holistic approach in designing an efficient delay-aware backpressure policy, that is also practical – it has low computational overhead and it is robust to topology changes.

5.1.1 Motivation

In the mean resource routing ($\mathcal{MR}$–$\mathcal{BP}$) methods [96, 91], backpressure is used with a parameter M that forces the links to stay inactive as long as their differential backlog doesn't exceed the value of M, i.e., links with backpressure less than M cannot be scheduled. This means that packets stay in queues longer, which can lead to higher delays. Indeed, we tested this intuitive result in [97] by evaluating the delay performance of these methods and confirmed this behavior. Therefore, since the $\mathcal{MR}-\mathcal{BP}$ methods are not delay competitive, we do not consider them as viable competitors any more.

The authors of [92] identified the scalability problems of the backpressure policy in *wireline networks* with millions of routers (e.g., Internet) due to the maintenance of several queues per node (one queue per destination, as it is mandated by the original $\mathcal{BP}$), and proposed the creation of clusters[†] of nodes so as to reduce the number of queues per node, which as a "side-product" has the additional benefit of delay reduction. Their policy, namely *cluster-based backpressure* ($\mathcal{CB}-\mathcal{BP}$), requires maintaining one queue per *gateway* at each relay node which leads to an excessive number of gateways, which in turn alleviates any performance gains (i.e., increases delays) when the number of clusters becomes, say, more than ten. Even worse, all contemporary clustering algorithms [98, 99] (such as the Distributed Clustering Algorithm, Max-min d-hop clustering algorithm) for wireless ad hoc networks produce quite a large number of clusters and thus several dozens of gateways even for relatively small networks. In these cases, the delay of $\mathcal{CB}-\mathcal{BP}$ approaches asymptotically the delays of the original $\mathcal{BP}$ (cf. Figure 5.6c). Moreover, the strong dependence of the policy on the identity of the gateways makes it problematic in cases of gateways breakdowns. Finally, considering the technicalities of the $\mathcal{CB}-\mathcal{BP}$ method, it is evident that even when a packet has already reached the destination cluster (i.e., the cluster where the destination nodes resides), the method is agnostic on this information and it may happen to send it again out of the cluster seeking alternative paths to the destination. This behavior was detected and improved in our experiments (cf. subsection 5.7.1).

The Shortest-paths backpressure ($\mathcal{SP}-\mathcal{BP}$) method [93] assumes the pre - computation of all pairwise-node distances and then application of the backpressure notion on the shortest path(s) between source and destination. Apparently, this method achieves the lower bound of the delay, it does so at the expense of a very complex initialization phase (all-node-pairs distances must be computed). Moreover, it must maintain a quadratic number of queues at each node ($n \times (n-1)$), whereas the original backpressure maintains only a linear number of queues ($n-1$) at each node. Also, during the running phase of the algorithm, the processing of such a huge number of queues is time-consuming, which in turn would increase delays. Apart from these computational-type problems, frequent topology changes would lead the method to break down, since many shortest paths would not exist anymore. Finally, in topologies where there is only one shortest-path per node pair (which is the most common case), $\mathcal{SP}-\mathcal{BP}$ would rapidly consume the energy of that path, shortening the longevity of the network. This problem becomes worse in topologies where a lot of shortest

[†]The authors did not propose any specific clustering algorithm.

paths traverse the same set of nodes (i.e., nodes with high betweenness centrality [13]); in such topologies, these nodes deplete their energy so fast, that the network gets partitioned very rapidly.

5.1.2 Contributions

The present research work presents two scheduling policies, namely the *Voting-based Backpressure ($\mathcal{V}o\mathcal{BP}$)* and the *Layered Backpressure ($\mathcal{L}ay\mathcal{BP}$)*. For the former method, taking a purely localized approach, we require the packets to carry their immediate travel history, so that the relays are prevented from sending the packets back from where they came. This is the opposite behavior of the ACO method [100], but they are similar in the sense that they both use hints – pheromone for ACO and history for $\mathcal{V}o\mathcal{BP}$. For the $\mathcal{L}ay\mathcal{BP}$ method, taking a less localized approach, we create "layers" of nodes and use the identities of these layers to forward the packets toward the destination's layer, and "discouraging" the packet from leaving the destination layer; these layers act as attractors for the packets. The two proposed methods exploit different characteristics of the network (topology vs traffic) and may be combined in order to further improve the performance of the system.

The research work presented in this chapter makes the following contributions:

- It experimentally evaluates well established delay-aware backpressure policies, namely $\mathcal{BP}$, $\mathcal{MR}-\mathcal{BP}$, $\mathcal{CB}-\mathcal{BP}$, $\mathcal{SP}-\mathcal{BP}$ for several network topologies.
- It develops the *Voting-based Backpressure* policy, which "wipes-outs" the ping-pong effect in backpressure-based packet scheduling. The method exploits current trafiic in the network in order to lead packets away from already visited nodes, vanishing rooting-loop.
- It develops the *Layered BackPressure ($\mathcal{L}ay\mathcal{BP}$)*, which divides the network into "layers" according to the connectivity of nodes. This method maintains the same number of queues with the original $\mathcal{BP}$, and one order of magnitude less number of queues compared to $\mathcal{SP}-\mathcal{BP}$. It does not require the existence of *gateways* and aggregated queues as does $\mathcal{CB}-\mathcal{BP}$.In complex networks gateways may be so many that $\mathcal{CB}-\mathcal{BP}$ performs like original $\mathcal{BP}$. In addition, it is not a deterministic method like $\mathcal{SP}-\mathcal{BP}$ where packets are forced to travel the shortest-path among nodes. Instead the packets are "suggested" to follow the shortest path from the source to the destination layer.
- It develops the Enhanced Layered Backpressure policy $\mathcal{EL}-\mathcal{BP}$ that improves the behavior of $\mathcal{L}ay\mathcal{BP}$ in case of moving nodes. The new protocol have the same characteristics with the $\mathcal{L}ay\mathcal{BP}$ and is robust to topology changes in terms of nodes that move from one layer to another.

- It develops a new random-walk based node clustering algorithm, that exploits the connectivity among nodes.
- It evaluates experimentally the performance of the proposed packet scheduling policies, with all the previous delay-aware throughput optimal backpressure methods.

The rest of this chapter is organized as follows: In Section 5.2 we survey the most important works relevant; Section 5.3 describes the network model. In Section 5.4 the original backpressure scheme is introduced; Section 5.5 presents the $\mathcal{V}o\mathcal{BP}$ scheme; Section 5.6 describes the $\mathcal{L}ay\mathcal{BP}$ algorithm; Section 5.6.4 describes the $\mathcal{EL}-\mathcal{BP}$ algorithm; Section 5.7 presents the simulation environment and results and finally Section 5.8 concludes the chapter.

5.2 Relevant work

A lot of packet scheduling protocols have been proposed in the relevant literature. Algorithms for packet scheduling/routing can be classified as data-centric, hierarchical, location-based and those based on network flow or quality of service awareness. The interested reader should refer to [101] for detailed categorization. In data-centric protocols [102], aggregation of data during the relaying of data is utilized. Hierarchical routing protocols [103] assign different roles to nodes. The nodes are divided into clusters and cluster-heads are assigned with the role of data aggregation and reduction in order to save energy. Location-based [104] protocols on the other hand use the position information of the nodes in order to forward the information to the desired region rather than the whole network. In some approaches, route setup is modeled and solved as a network flow problem [90]. System stability and throughput maximization based on [90] have been exclusively studied [105, 106]. Related work on delay-efficient backpressure algorithms is developed in [91, 107, 93, 108, 109].

5.3 Network model

We consider a network $G = (V, L)$, where V is the set of nodes (vertices) and L is the set of links (edges). We consider the following generic properties:

- Nodes are static or mobile, the communication links are bidirectional, and nodes communicate in a multi-hop fashion.
- Topology changes may take place, due to nodes getting down or link failure.
- Network nodes are homogeneous and do not have GPS-like hardware. Links have equal to one capacity.

- Concurrent transmissions cause mutual interference since the transmission medium is shared. Matchings are the set of links that can be scheduled simultaneously. A max-weight scheduling policy is used.
- A node cannot transmit and receive at the same time.
- Time is slotted with a time slot t.
- For each node i, there is an arrival process E_i such that $E_i(t)$ is the number of exogenous arrivals up to time t. We assume that packets arrive exponentially with mean arrival rate λ. The source and destination of each packet is randomly selected among all the nodes. Special cases are also investigated by using the Zipf's distribution, where packets 'prefer' some nodes as destinations representing a more realistic scenario.

5.4 The original Backpressure algorithm

Backpressure [90] is a joint scheduling and routing policy which favors traffic with high backlog differentials. The backpressure algorithm performs the following actions for routing and scheduling decisions at every time slot t.

- *Resource allocation*

  For each link (n, m) assign a temporary weight according to the differential backlog of every commodity (destination) in the network:

  $$wt_{nmd}(t) = max(Q_n^d - Q_m^d, 0).$$

  Then, define the maximum difference of queue backlogs according to:

  $$w_{nm}(t) = \max_{d \epsilon D} wt_{nm}(t).$$

  Let $d^*_{mn}[t]$ be the commodity with maximum backpressure for link (n,m) at time slot t.

- *Scheduling*

  The network controller chooses the control action that solves the following optimization problem:

  $$\mu^*(t) = \underset{\mu \epsilon \Gamma}{argmax} \sum_{(n,m)\epsilon L} \mu_{nm} w_{nm}(t),$$

  Let Γ denote the set of all schedules subject to the one hop interference model.

  In our model, where the capacity of every link μ_{nm} equals to one, the chosen schedule maximizes the sum of weights. Ties are broken arbitrarily.

- *Routing*
  At time slot t, each link (n, m) that belongs to the selected scheduling policy forwards one packet of the commodity $d^*_{mn}[t]$ from node n to node m. The routes are determined on the basis of differential queue backlog providing adaptivity of the method to congestion.

Backpressure algorithm is throughput-optimal and discourages transmitting to congested nodes, utilizing all possible paths between source and destination. This property, leads to unnecessary end-to-end delay when the traffic load is light. Moreover, using longer paths in cases of light or moderate traffic wastes network resources (node energy).

5.5 Voting-based backpressure

The driving idea behind the $\mathcal{V}o\mathcal{BP}$ scheduling policy is the reduction of the path length traveled by a packet by reducing any cycles observed in this path. Apparently, this can be done by having the packet "carrying" its trajectory. Such an approach though, would comprise the properties of the backpressure algorithm (i.e., utilize all paths), and also it would be impractical, since these trajectories grow remarkably large for long paths, having as a consequence a considerable increase in the packet size and increased processing time by the routers. Therefore the storage of such "rich" information in the packets is not a viable option, unless we confine the amount of information. This is exactly the direction followed by the $\mathcal{V}o\mathcal{BP}$ algorithm.

$\mathcal{V}o\mathcal{BP}$ uses minimum information stored on the packets in order to select the scheduling policy at each time slot. Every packet holds the last node it has previously visited. This information is used in the scheduling phase of the protocol in order to prevent packets from revisiting the same nodes and traveling in circles in the network. Each node n maintains a separate queue of packets for each destination $Q^d_n[t]$ (as the original backpressure algorithm). Each queue d has a counter for each neighbor node C_{nmd}. This counter is updated every time a packet arrives to or leaves from node n having as final destination node d. Since the packet holds information about the last visited node a certain procedure is followed at each transition.

When packet i having as final destination node d arrives at node n from node m these actions take place.

- At node n C_{nmd} is incremented by one.
- At node m C_{nmd} is decremented by one.
- Packet i updates last visited node from m to n.

In order to find the worst neighbor all the packets that belong to each queue participate in a voting procedure updating the appropriate counters. The node that collects the more votes according to the information of the packets, is given a weight in the scheduling algorithm preventing the controller from choosing such a routing policy that moves the packets backwards in the network. Parameter A is used in order to assign weights to nodes and help the routing mechanism forward packets to delay efficient paths. The tuning of this parameter is evaluated in section 5.7.1. The Voting-based backpressure algorithm executes at time slot t as follows:

- Each queue votes for the Bad neighbor B_{nmd} of node n according to: $B_{nmd} = \max C_{nmd}$.

- Each link (m, n) is assigned temporary weights according to the differential backlog $wt_{nmd}(t) = max(Q_n^d - Q_m^d, 0)$ and a parameter A_{nmd} according to :

$$A_{nmd} = \begin{cases} 1/A, & \text{if } m = B_{nmd} \\ 1 & \text{otherwise.} \end{cases}$$

- Each link is assigned a final weight according to w_{nm} and parameter A_{nmd}:

$$w_{nm}(t) = \max_{d \epsilon D}(wt_{nmd}(t) * A_{nmd}).$$

- The network controller chooses the control action that solves the following optimization:

$$\mu^*(t) = \underset{\mu \epsilon \Gamma}{argmax} \sum_{(n,m) \epsilon L} \mu_{nm} w_{nm}(t)$$

  subject to the interference model mentioned above where adjacent links are not allowed to be active simultaneously.

In each time slot only nodes that transmit or receive packets update the information in the appropriate queues decreasing the complexity of the method. Except from the worst node, also the good node G_{nmd} ($G_{nmd} = \min C_{nmd}$) or nodes to send packets can be retrieved from the above procedure giving an extra bonus to this node for the corresponding queue. Good nodes should be the ones with the least or even zero votes, meaning that no packet has visited them in the previous hop. Parameter A_{nmd} would be in that way:

$$A_{nmd} = \begin{cases} A, & \text{if } m = G_{nmd} \\ 1/A, & \text{if } m = B_{nmd} \\ 1 & \text{otherwise.} \end{cases}$$

As shown in the experiments (cf. subsection 5.7.1), this policy can significantly reduce delays in low-loaded wireless networks, and moreover, it can be used in combination with other backpressure-type algorithms to enhance their performance. The method shows extremely good performance in networks with very low connectivity between nodes where only a few paths exist for the packets to follow in order to reach their destination. The packets by holding the information about the previously visited nodes help the fast propagation of the information.

Even though a detailed evaluation of the method is presented in a later section, to support the above claims we tested the method in a ring network consisting of twelve nodes (see Figure 5.1). The performance of $\mathcal{V}o\mathcal{BP}$ is similar to that of the shortest path backpressure. This performance is achieved without the need of any extra queues or other information, like all the distances among the nodes, except from the last visited node that every packet holds.

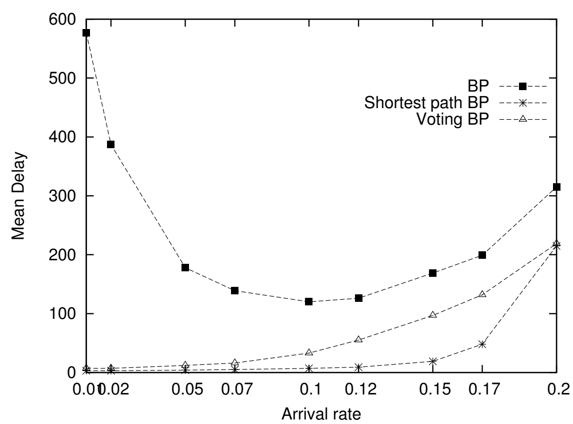


Figure 5.1: Performance of $\mathcal{V}o\mathcal{BP}$ in a 12-node ring network

5.6 Layered backpressure

This section describes a delay-efficient backpressure algorithm based on the creation of *layers*[‡] in the network. The main idea is to split the network into layers according to the connectivity among them, which also (usually) implies geographic proximity, as well. These layers divide the initial graph to k smaller networks. Then the algorithm forwards packets according to the destination layer ID, thus effectively reducing the long paths. In some sense, these layers play the role of attractors, that attract the packets destined for them

[‡]We use the term *layer* to describe node partitioning, since the target of this research is underwater sensor networks for surveillance applications, where sensors are placed in layers beneath the sea surface.

and then “disallow” the packets to leave the layer. Apparently, if we have only one layer then the policy reduces to the original backpressure algorithm.

For the implementation of the $\mathcal{L}ay\mathcal{BP}$ scheduling policy, every node n keeps a separate queue for each destination/flow going through it and also holds the layer $Layer(n)$ that the destination belongs to. The backpressure scheduling is executed using only the IDs of these layers. This is a significant difference from the work of [92], because $\mathcal{L}ay\mathcal{BP}$ does not require the knowledge of gateways' queue lengths. The “correct” partitioning of nodes into layers and a proximity-based naming of layers is of paramount importance for the performance of every routing method which based on clustering. Since in the proposed $\mathcal{L}ay\mathcal{BP}$ method no gateways are required the method is more robust to bad partitioning. Also the method, as it is shown in section 5.7.1, performs better as the number of clusters grow, in contrast to $\mathcal{CB}{-}\mathcal{BP}$. Apparently, the concept of ”correctness” is algorithmic, and we only need to set a partitioning criterion; we proposed to determine the number of layers based on the network topology, i.e., connectivity. The next two subsections describe the layer-creation and layer-naming algorithm, whereas subsection 5.6.3 presents the Layered Backpressure packet scheduling algorithm.

5.6.1 Random-walk-based layering

In this section we present a Random Walk Based Clustering (RWC) technique for creating the necessary layers for the routing protocol. RWC runs in the initiation phase of the network. The method is centralized and the number of layers Nc to be created are discovered by an exhaustive cost-optimal algorithm. Techniques for estimating this number can be found at [13]. The layers created by this method have approximately the same cardinality, which is denoted by the parameter Npc. Every node has a parameter indicating if the layer it belongs to is “final” or not. In the beginning of the algorithm all nodes are assigned a random layer ID, from 1 to Nc. The final layer of each node is determined by the RWC algorithm after a number of iterations. For all the nodes that do not have their layers fixed by the algorithm yet, we say that they belong to a “temporary” layer.

The algorithm runs in blocks where in each block a node i is selected as a source. The algorithm takes successive random steps from i for a predefined number of steps h for K iterations. Every node holds a counter indicating how many times it is visited from the random walk algorithm in the current block. The neighborhood of the node i is created according to Definition 3.

Definition 3 (Neighborhood of node i)**.** *A node j belongs to the neighborhood Gi of the node i, if the number of times the node is visited is at least equal to the times the RWC traversed node i.*

After the creation of the neighborhood Gi the proposed method operates as follows:

1. Find the layer $Poslevel$ having the maximum number of members of the current neighborhood. This is the possible layer of all the nodes that belong to the Gi.
2. If this layer is different from the layer of source node i then find a node outside the neighborhood that has this temporary layer and exchange values (layers) with node i.
3. If the number of members of the $Poslevel$ is less than Npc and greater than one then try to find a node outside the layer with this temporary layer and exchange layers with a node belonging to the Gni but having temporary layer different from the $posLevel$.
4. If the number of nodes that belong to the possible level after the previous steps is Npc with deviation of one then the layer of these nodes is fixed to the $Poslevel$.

The procedure moves in small steps inserting at each iteration at most two nodes in the possible layer leading to a relatively fair clustering of the nodes after its termination. This procedure is repeated for each node belonging to a temporary layer until all the layers are fixed. The nodes that, after this procedure still belong to temporary layers, fix their values according to the layers of their neighbors.

5.6.2 Naming of Layers

The layers created by the RWC algorithm have labels that do not represent the actual connectivity (proximity) among them. The Layered Backpressure protocol, that is based on the RWC algorithm, needs to forward packets according to the layer's labels. In order for $\mathcal{L}ay\mathcal{BP}$ to work efficiently with layers, we should assign labels to layers according to their geographic proximity. After the termination of the RWC, another algorithm is used to perform this task. We use a breadth-first-visiting (BFS) algorithm to carry out this task; though more efficient algorithms can be employed, like fractal curves, e.g., Hilbert curve, which provide a linear ordering of two-dimensional data according to their proximity, this investigation is beyond the scope of the present work.

5.6.3 The Layered BackPressure protocol

After the completion of the grouping and the assignment of IDs to the layers, the actual packet scheduling is performed as follows. Each node n maintains a separate queue of packets for each destination. The length of such queue is denoted as $Q_n^d(t)$. For every queue $Q_n^d(t)$ the node computes the parameter $Dlevel_n^d$ which represents the absolute difference between current and destination node's layer number: $Dlevel_n^d = |Layer(n) - Layer(d)|$. At each time slot t, the network controller observes the queue backlog matrix $Q(t) = (Q_n^d(t))$ and performs the following actions for routing:

Layered BackPressure at time slot t:

- Each link (n,m) is assigned a temporary weight according to the differential backlog $wt_{nmd}(t) = (Q_n^d - Q_m^d)$ and parameter A_{nmd} according to:

$$A_{nmd} = \begin{cases} 2, & \text{if } Dlevel_n^d > Dlevel_m^d \\ 1/2, & \text{if } Dlevel_n^d < Dlevel_m^d \\ 1 & \text{otherwise.} \end{cases}$$

- Each link is assigned a final weight according to w_{nm} and parameter A_{mnd}:

$$w_{nm}(t) = \max_{d \epsilon D}(wt_{nmd}(t) * A_{nmd}).$$

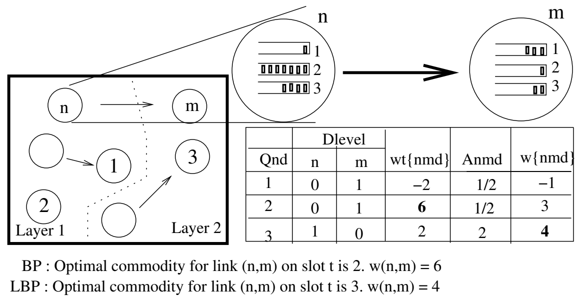
- The network conrtoller chooses the control action that solves the following optimization:

$$\mu^*(t) = \underset{\mu \epsilon \Gamma}{argmax} \sum_{(n,m) \epsilon L} \mu_{nm} w_{nm}(t)$$

subject to the interference model mentioned above where adjacent links are not allowed to be active simultaneously.

A simple example is illustrated in Figure 5.2. The network in Figure 5.2 consists of 7 nodes where all nodes are senders and only nodes with id's $1, 2, 3$ are destination nodes. In Figure 5.2 we observe the status of the queue backlogs at a specific time slot t. The network is divided in two layers. Layer one includes nodes in the left side of the dot line. Running the initial backpressure algorithm the commodity that achieves maximum differential backlog for link (n, m) is 2 while for the proposed layer backpressure is 3. If we assume that link (n, m) is scheduled at that time slot then for the initial backpressure algorithm a packet destined for node 2 will be forwarded to node m pushing it further away and forcing it to follow longer path in order to reach its destination (node 2). With the layer backpressure policy at the same time moment the link (n, m) if scheduled forwards a packet of commodity 3 from node n to node m which is closer to the destination of the packet. Furthermore node m belongs to the same cluster as node n, which according to the proposed layer backpressure policy will prevent the packet from returning to node n.

Theorem 1 (Throughput optimality). *The $\mathcal{L}ay\mathcal{BP}$ and $\mathcal{V}o\mathcal{BP}$ algorithms are throughput optimal.*



	Dlevel				
Qnd	n	m	wt{nmd}	Anmd	w{nmd}
1	0	1	−2	1/2	−1
2	0	1	**6**	1/2	3
3	1	0	2	2	**4**

Figure 5.2: An example of the $\mathcal{LayBP}$ execution.

Proof. The original backpressure algorithm is throughput optimal. In order to prove that the $\mathcal{VoBP}$ and $\mathcal{LayBP}$ are also throughput optimal, the Lyapunov stability criterion [110] is used. The idea behind the Lyapunov drift technique is to define a non-negative function, called the Lyapunov function, which represents the aggregate congestion of all queues (Q_n^d) of the network. The drift of the function at two successive time slots is then taken, and in order for the policy to be throughput optimal, this drift must be negative, when the sum of queue backlogs is sufficiently large. For both strategies, we use:

$$L(Q) = \sum_{nd} \theta_n^d (Q_n^d)^2$$

as the Lyapunov function where weights θ_n^d are used to offer priority service.

Recall that each link is assigned a final weight according to w_{nm} and parameter A_{mnd}:

$$w_{nm}(t) = \max_{d \epsilon D}(wt_{nmd}(t) * A_{nmd}).$$

This equation can be rewritten in the following form:

$$w_{nm}(t) = \max_{d \epsilon D}(A_{nmd} * Q_n^d - A_{nmd} * Q_m^d).$$

which is equivalent to :

$$w_{nm}(t) = \max_{d \epsilon D}(\theta_n^d * Q_n^d - \theta_m^d * Q_m^d),$$

where weights θ_i^d are used to offer priority service.

Queue dynamics at each time slot satisfy :

$Q_n^d(t+1) \leq max[Q_n^d(t) - \sum_b \mu_{nb}^d(t), 0] + A_n^d(t) - \sum_a \mu_{an}^d(t),$

where $\mu_{nm}^d(t)$ are routing control variables, representing the amount of commodity d data delivered over link (n, m) during slot t and $A_n^d(t)$ represent the process of exogenous commodity d data arriving to source node n.

$$(Q_n^d(t+1))^2 \leq ([Q_n^d(t))^2 + (\sum_b \mu_{nb}^d(t))^2 + (A_n^d(t) + \sum_a \mu_{an}^d(t))^2 -$$

$2[Q_n^d(t)(\sum_b \mu_{nb}^d(t) - A_n^d(t) - \mu_{an}^d(t))$

Multiplying both sides with θ_n^d, summing over all valid entries (n, d) and using the fact that the sum of squares of non-negative variables is less than or equal to the square of the sum we take :

$$\sum_{nd} \theta_n^d (Q_n^d(t+1))^2 \leq \sum_{nd} \theta_n^d (Q_n^d(t))^2 + \sum_{nd} \theta_n^d (\sum_b \mu_{nb}^d(t))^2 +$$

$$\sum_{nd} \theta_n^d (A_n^d(t) + \sum_a \mu_{an}^d(t))^2 - 2 \sum_{nd} \theta_n^d Q_n^d(t) (\sum_b \mu_{nb}^d(t) - A_n^d(t) - \mu_{an}^d(t))$$

The equation can be rewritten:

$\Delta L(Q) \leq 2BN - 2\sum_{nd} \theta_n^d \epsilon Q_n^d(t)$,where $B \triangleq \frac{1}{2N} \sum_{n\varepsilon N} \theta_{max} [(\mu_{max,n}^{out})^2 + (A_n^{max} + \mu_{max,n}^{in})^2]$

Using the above we can rewrite drift inequality :

$\Delta L(Q) \leq 2B'N\theta_{max} - 2\sum_{nd} \theta_n^d \epsilon Q_n^d(t)$, where $B' \triangleq \frac{1}{2N^2} \sum_{n\varepsilon N} [(\mu_{max,n}^{out})^2 + (A_n^{max} + \mu_{max,n}^{in})^2]$

This drift inequality is in the exact form for application of the Lyapunov drift lemma, proving the stability of the algorithm.

The weighted sum of all queues is :

$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t} E\{\sum_{n,d} \theta_n^d Q_n^d(\tau)\} \leq \frac{NB'\theta_{max}}{\epsilon_{max}}$ □

In $\mathcal{V}o\mathcal{BP}$ method where the weights are dynamically varying over time, such that $\theta_n^d \in [\theta_{min}, \theta_{max}]$, network stability is still ensured [111].

5.6.4 The Enhanced Layered Backpressure policy

Routing protocols must be dynamic in order to cope with mobility of nodes in modern wireless networks. Widely varying mobility characteristics are expected to have a significant impact on the performance of routing protocols that are based on node grouping (like $\mathcal{CB}-\mathcal{BP}$, $\mathcal{L}ay\mathcal{BP}$) in order to route packets even if links among nodes are updated. In case of grouping-based routing protocols, high mobility of nodes which lead them to change groups, degrades the performance of the methods since this 'wrong' information is used in the routing procedure. Although $\mathcal{LayBP}$ doesn't use gateways, it still suffers from this behavior if the layer that the moving nodes belong to, are not updated. The differential backlog of each link is computed according to the difference between current and destination's node layer. It is clear that $\mathcal{LayBP}$ behavior can be affected of 'misplaced' nodes. In this case packet may be forwarded to layers different than the desired making the method inappropriate.

Looking at Figure 5.3 we see that node 7 (moving node) initially belongs to cluster 2 at moment T0. At moment T1 node 7 moved to cluster 1 but is agnostic of it. Packets destined to node 7 are still forwarded by the LayBP to cluster 2 making it difficult to reach their final destination thus making the intermediate node's queues to grow up.

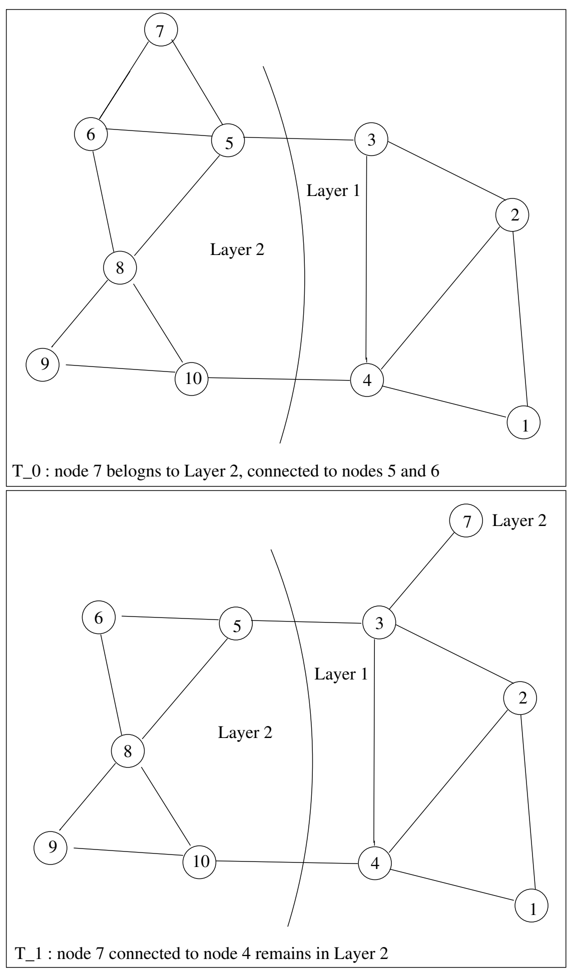


Figure 5.3: Example network with a moving node.

In order to cope with node mobility we incorporate in $\mathcal{LayBP}$ algorithm another step in all nodes recalculate their cluster according to their neighborhood. In the initiation phase every node has a counter $C0_{nl}$ for every layer ID, indicating how many neighbors belong to it and a variable $Layer_n$ indicating the layer that node n belongs to. Every time slot t the following actions are performed:

- Calculate C_{nl}, the total number of neighbors that belong to every layer detected.
- if $C_{nl} >= C0_{nl}$ for $l = Layer_n$ then moving node remains in the same layer.
- else calculate the layer with the most neighbors $M_{nl} = maxC_{nl}$. if $M_{nl} > C_{nl}$ then moving node belongs to layer M_{nl}.
- assign for every layer l, $C0_{nl} = C_{nl}$ as new initial values for next time slot.

This algorithm can be executed every k time slots according to how fast we want the system to adapt to node mobility. Moving nodes need to execute the algorithm every second in order to find the appropriate layer they belong to. Static nodes need to update their information more rarely, than the moving nodes do, since a certain number of neighbors must be replaced in order to affect them. The algorithm does not perform reclustering, but only 'helps' layers incorporate moving nodes.

5.7 Evaluation

For the evaluation of the proposed scheduling policies we developed a custom simulator, modeling topology features, packet scheduling, link activation, etc. We considered as competitors state-of-the-art delay-aware backpressure policies that have been published so far in the literature, with the exception of the $\mathcal{MR}{-}\mathcal{BP}$ policies which was shown in subsection 5.1.1 that it provides no delay performance benefits. We simulated the execution of the algorithms for various network topologies for large simulation times and recorded their average performance.

5.7.1 Static networks

Experimental setting

Methods compared. As competing methods we examined currently proposed delay-aware backpressure policies along with the original backpressure method as the baseline algorithm. We emphasize here that the shortest path backpressure [93] is the theoretically optimal w.r.t. delay method. This method performs the best in all cases and acts as the lower bound, but it is vulnerable to topology changes (cf. Figure 5.10). For the methods needing some form of clustering, i.e., $\mathcal{CB}-\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$, the clustering is performed with the proposed random-walk clustering algorithm. For the $\mathcal{CB}-\mathcal{BP}$ in order to improve the misbehavior of the method when source and destination are in the same cluster, the traffic controller can directly route them without relaying to any gateway, even if source is a gateway of $cluster(d)$ itself.

Performance measures. As performance measures we used the average end-to-end delay and the number of messages successfully delivered per unit time (network throughput).

Network topologies. As network topologies, we considered a $4 \times 4 = 16$ nodes grid network, and those illustrated in Figure 5.4. The grid network is very common in the literature [91, 93] and it comprises a very "comfortable" case for $\mathcal{SP}-\mathcal{BP}$ since there are more than one shortest paths between each pair of nodes. Also, the other two topologies (those in Figure 5.4) were borrowed from [98] with 22 and 30 nodes, respectively. For the methods that use clusters, the clustering is depicted in the figure; the clusters of the grid net are comprised by the four quadrants. In summary, we double the size of the network (from 16 to 30 nodes) and also variate the type of connectivity among clusters, i.e., clusters in a "ring" (16-node net), clusters "almost in a line" (22-node net), and clusters in a "all-to-all communication". The performance of the methods are similar regardless of the size of the network.

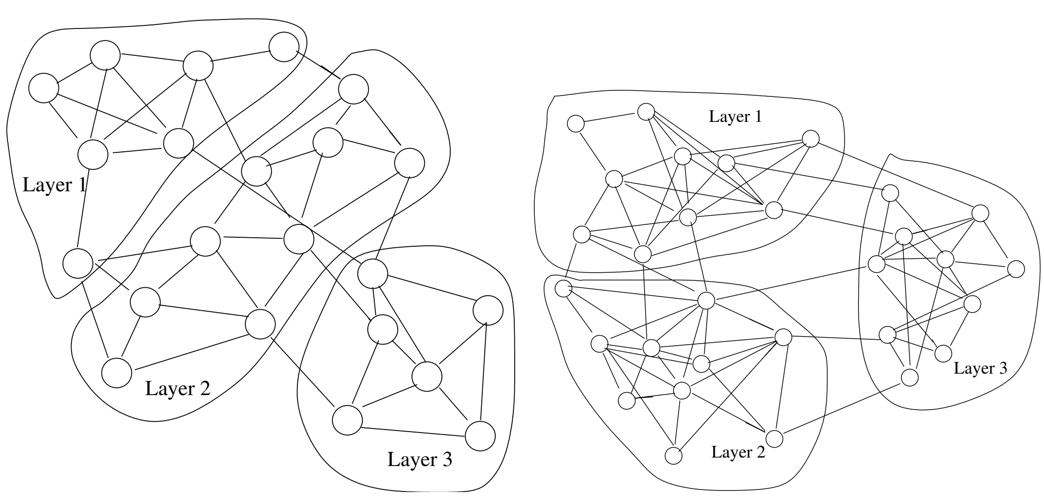


Figure 5.4: Network topologies with 22 and 30 nodes.

Packet generation. The exogenous arrival process at each node is independent Bernoulli with rate λ. The destination of every generated packet is randomly chosen. Experiments are also conducted where the destination is chosen according to Zipf's law in order to simulate situations where there is a "preference" toward some specific destination node. Since in the conducted experiments with Zipfian-selected destination, the obtained results showed no alteration in the relative performance of the algorithms, all the results we present concern the uniformly-at-random-selected destination. Different values of λ are chosen according to the number of nodes of the network in order to simulate moderate network traffic. This means that the same λ might represent low load for a small network, and high traffic for a much larger network.

Experimental results

The goal of the experimental evaluation is, apart from testing the delay and throughput performance of all the competing methods, to validate all the claims reported in subsection 5.1.1 out the misbehavior of $\mathcal{CB}-\mathcal{BP}$ and $\mathcal{SP}-\mathcal{BP}$. First of all, we examine the impact of parameter A on the performance of the proposed methods so as to tune them for the rest of the experiments.

Parameter A - Tuning of the proposed methods. Both $\mathcal{V}o\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$ use the parameter A to forward packets; $\mathcal{L}ay\mathcal{BP}$ uses it to forward packets to the destination cluster according to layers' IDs, and in $\mathcal{V}o\mathcal{BP}$, the parameter A plays the role of discouraging packets to move backwards in the network. The case of $A = 1$ corrsponds to the traditional backpressure algorithm. We evaluated the performance of the methods for various A and we present in (Figure 5.5) the ones concerning $\mathcal{L}ay\mathcal{BP}$ ($\mathcal{V}o\mathcal{BP}$ method showed similar performance). The obtained results showed no significant gains (either in terms of delay reduction or throughput increase) for values other than $A = 2$; thus, for the rest of the experiments, we use $A = 2$.

Impact of layer/cluster number. In section 5.1.1, we stated that the performance of $\mathcal{CB}-\mathcal{BP}$ deteriorates with increasing number of clusters; indeed this is the pattern of its behavior, as shown in Figure 5.6 (tested on a grid topology growing with the number of clusters), where beyond 5 clusters, the performance of $\mathcal{CB}-\mathcal{BP}$ is almost 25 times worse than that of $\mathcal{L}ay\mathcal{BP}$. The same behavior is observed in all the network topologies we simulated. That behavior is predicted by the description in [92], where the increased number of clusters implies increased number of gateways and thus more delays. $\mathcal{CB}-\mathcal{BP}$ requires a very small constant number of clusters where most ad hoc clustering protocols produce a number of clusters which is linear or logarithmic in the number of network nodes. On the other hand

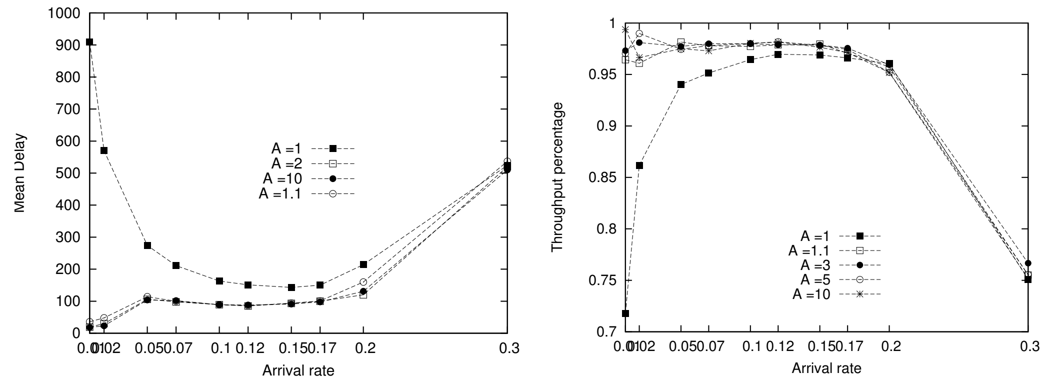


Figure 5.5: Tuning of parameter A - delay prformance of $\mathcal{L}ay\mathcal{BP}$

the proposed $\mathcal{L}ay\mathcal{BP}$ method performs better as the network is divided in more clusters (see Figure 5.6). This positive characteristic makes the proposed routing method more efficient when combined with most of the automated algorithmic network clustering algorithms.

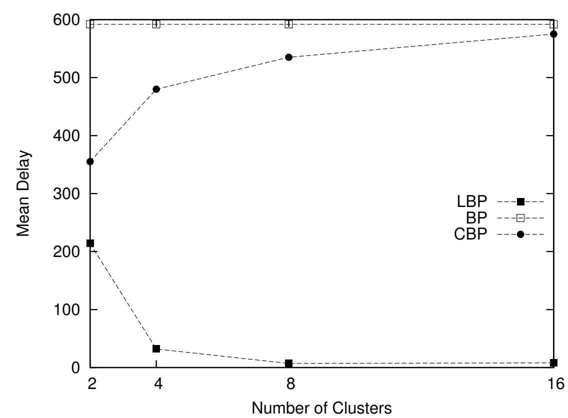


Figure 5.6: Impact of the number of clusters on the grid network topology.

Delay and throughput performance. The plots in Figure 5.7 depict the delay performance of the competing methods. The first generic observation is that the performance of $\mathcal{L}ay\mathcal{BP}$ follows – as a trend – the performance of $\mathcal{SP}-\mathcal{BP}$, whereas the performance of $\mathcal{CB}-\mathcal{BP}$ follows – as a trend – that of the original $\mathcal{BP}$. Additionally, in all cases $\mathcal{L}ay\mathcal{BP}$ is the best policy outperformed only by $\mathcal{SP}-\mathcal{BP}$ (the theoretically optimal policy).

$\mathcal{V}o\mathcal{BP}$ method performs better than all the methods (except of course from $\mathcal{SP}-\mathcal{BP}$), when the network traffic is extremely low and in the topologies where there do not exist many alternative paths for the traveling packets. This has to do with the average node degree of the network. For instance, the delay of $\mathcal{V}o\mathcal{BP}$ is significantly higher (see Figure 5.7c) for the 30-node network (presented in Figure 5.4b) which is quite dense.

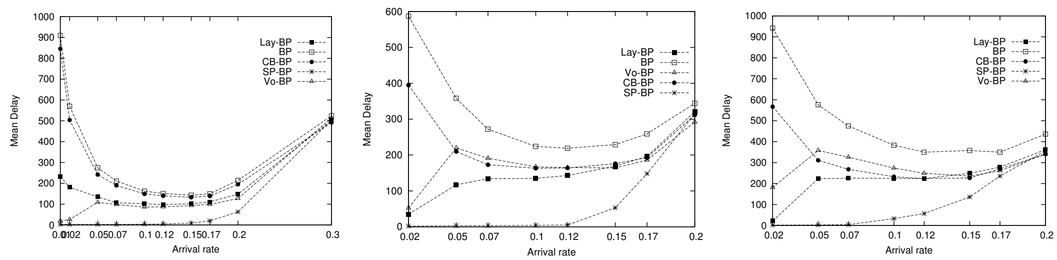


Figure 5.7: Delay performance for the three network topologies (16, 22 and 30 nodes).

The plots in Figure 5.8 depict the throughput performance of the competing methods. The obvious observation concerns the optimality of all methods, including optimality of $\mathcal{V}o\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$ according to Theorem 1.

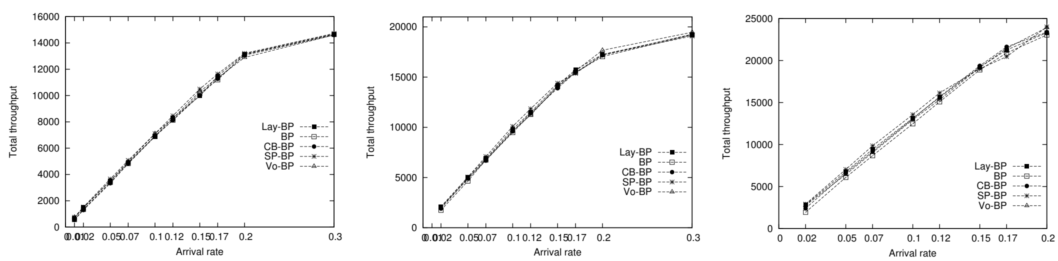


Figure 5.8: Throughput performance for the three network topologies (16, 22 and 30 nodes).

Maximum end to end delay. The quality of service (QoS) is defined by the following technical parameters: transmission rate, loss rate,average throughput, maximum end-to-end delay, and maximum delay jitter. Maximum end to end delay is an important metric of QoS when critical message transmission is concerned. In Figure 5.9 maximum end to end delay is shown for the 30 node topology. We observe that both $\mathcal{BP}$ and $\mathcal{CB}-\mathcal{BP}$ undergo high maximum end to end delay for low and moderate traffic. Both proposed methods manage to keep maximum end to end delay low while they even outperform $\mathcal{SP}-\mathcal{BP}$ for moderate network traffic. This is due to the fact that $\mathcal{SP}-\mathcal{BP}$ guide all packets through short routes leading to congestion over the links that compose these paths and making packets suffer from very long delays.

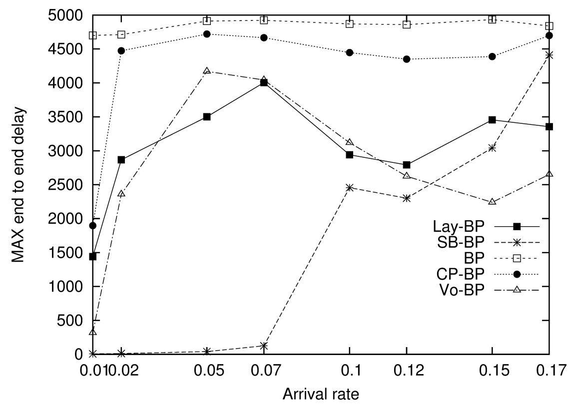


Figure 5.9: Maximum end to end delay for the network topology with 30 nodes

Impact of topology change (link breakdown). To validate the claims about the impact of topology on the shortest path backpressure method, we performed an experiment on a small sample network of 10 nodes. The network consists of two clusters; the first includes six nodes and second cluster includes four nodes. The clusters are connected to each other with two links only. We inactivate one of the two links that connect the two clusters without recalculating the shortest paths among all node pairs. The results presented in Figure 5.10 show that $\mathcal{SP}-\mathcal{BP}$ practically breaks down – as expected – whereas $\mathcal{L}ay\mathcal{BP}$ maintains its strength. We do expect that dramatic changes will impact also $\mathcal{L}ay\mathcal{BP}$, but $\mathcal{L}ay\mathcal{BP}$ is more robust to topology changes since it does not require exact knowledge of shortest paths among nodes, but rather shortest paths among clusters.

Since both $\mathcal{V}o\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$ proposed methods favor packets to follow delay efficient paths, but in neither case in a deterministic way, the routing algorithm may get "stuck" only temporarily due to link breakdown. Both methods manage to overcome effectively link breakdown, a common situation in ad hoc networks, for both low and moderate network traffic.

Load Deviation In sensor networks an important performance metric of a routing method is load deviation which is strictly related to energy consumption. Limited energy resources, demand methods to balance the load among all nodes while keeping the total load relatively low. In figure 5.11 load deviation of the typical 16 grid network is presented. Is is shown that both $\mathcal{L}ay\mathcal{BP}$ and $\mathcal{V}o\mathcal{BP}$ outperform $\mathcal{CB}-\mathcal{BP}$ and $\mathcal{BP}$. On the other hand, though $\mathcal{SP}-\mathcal{BP}$ approach has the lower load deviation, this doesn't clearly represent energy consumption, since complex calculations that the method demands to be performed by all nodes every second are not calculated.

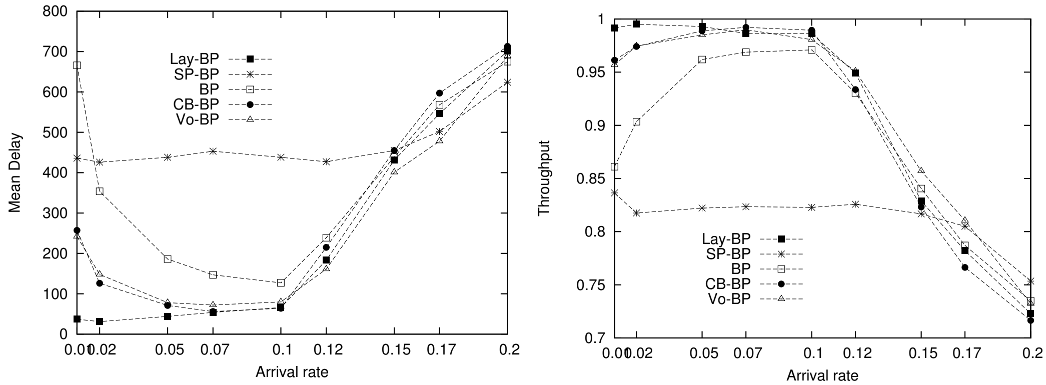


Figure 5.10: Impact of topology changes on the delay and throughput performance of $\mathcal{SP}{-}\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$.

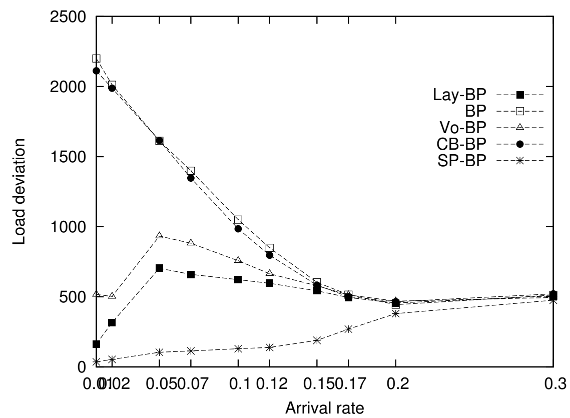


Figure 5.11: Load deviation for the grid network topology

5.7.2 Dynamic networks

Experimental setting

We evaluated the performance of the cluster-based algorithms in networks with mobile nodes in order to measure the adaptivity of the $\mathcal{EL}{-}\mathcal{BP}$ method. It is important here to mention that $\mathcal{EL}{-}\mathcal{BP}$ copes with topology changes in terms of mobility nodes, which is different than the case evaluated earlier, where a link breakdown is used in order to show the lack of adaptivity of $\mathcal{SP}{-}\mathcal{BP}$ method to topology changes. For the evaluations we conducted, we assume that moving nodes update their link information, according to some handshaking mechanism, but are unaware of layer changes.

Network topologies. As network topology, we considered that illustrated in Figure 5.12. The topology is a network of 13 nodes where node 13 moves from layer 1 (time slot 0) to layer 2 (time slot 1000) and finally to layer 3 (time slot 2000). $\mathcal{BP}$, $\mathcal{CB}-\mathcal{BP}$ and $\mathcal{L}ay\mathcal{BP}$ are unaware of layer change while all the active links of the moving node are updated. In $\mathcal{EL}-\mathcal{BP}$ method, update algorithm runs at every time slot and moving nodes join the appropriate layer according to neighbor's information.

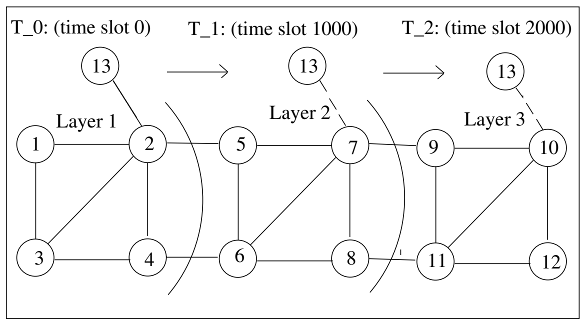


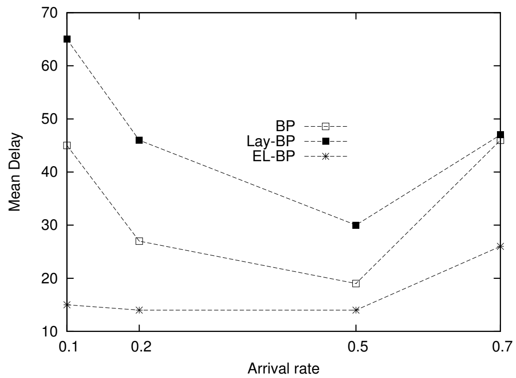
Figure 5.12: Dynamic network with 3 layers.

Packet generation. In order to have a clear view of the performance of the cluster-based methods we generated at each experiment only one flow from node 1 to the moving node. The exogenous arrival processes are independent Bernoulli processes with rate λ.

Experimental results

Delay and throughput performance. It is clear that $\mathcal{CB}-\mathcal{BP}$ brakes down in situations where nodes change cluster. Based on the fact that packets only after entering the correct cluster can be directed to the correspondent queue, this makes the performance of $\mathcal{CB}-\mathcal{BP}$ severely bad. Indeed, in our simulations, the performance of $\mathcal{CB}-\mathcal{BP}$ for such cases was 20 times worse than that of the rest of the algorithms, and thus we do not show it the plots. The plots in Figure 5.13 show the delay and the throughput performance of the $\mathcal{L}ay\mathcal{BP}$, $\mathcal{EL}-\mathcal{BP}$ and the original backpressure.

By studying the delay performance of the compared methods,we see that $\mathcal{L}ay\mathcal{BP}$ behaves worse than the original $\mathcal{BP}$ in cases of moving nodes. The $\mathcal{EL}-\mathcal{BP}$ gives the best outcome since the moving node updates the layer information according to the neighborhood. It is clear that in situations where nodes move along many layers (second dynamic topology) the behavior of $\mathcal{L}ay\mathcal{BP}$ degrades, while $\mathcal{EL}-\mathcal{BP}$ retains its behavior. All methods give similar outcomes as far as the number of messages successfully delivered per unit time is concerned (Network Throughput).



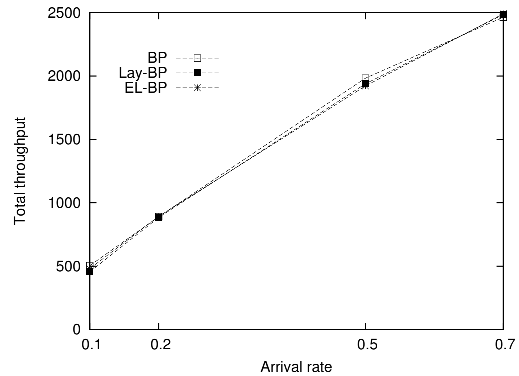


Figure 5.13: Impact of moving nodes on the delay and on the throughput performance of $\mathcal{BP}$, $\mathcal{LayBP}$ and $\mathcal{EL}{-}\mathcal{BP}$ for the dynamic network with the 3 layers.

5.8 Chapter conclusions

We considered the problem of packet scheduling in wireless networks using backpressure-type algorithms. The purpose of the work is to address the delay-efficiency problems of the backpressure-type algorithms. In this context, we described two algorithms; the first method, namely $\mathcal{V}o\mathcal{BP}$, alleviates the ping-pong effect in packet scheduling, and the second algorithm investigates the issue of creating layers of nodes and performing backpressure scheduling using the IDs of the layers so as to "push" the packets toward the destination layer. We conclude that Voting backpressure performs ideally for low connectivity topologies while Layered backpressure is a high performance policy compromising a little delay for robustness, low computational complexity and simplicity. Push-based data dissemination mechanisms on the other hand use fixed RSU or moving vehicles to periodically deliver data messages to other vehicles. Information about road conditions or traffic can be disseminated by RSUs and low tuning and access time are crucial parameters of the performance of a push-based broadcast system.

Chapter 6
Smart Broadcasting in Intelligent Transportation Systems*

Wireless data broadcast received a lot of attention from industries and academia in recent years. In any form of a push-based broadcast, access latency and tuning time are vital issues, and in order to address the tradeoff among these competing goals, the broadcasting of indices along with the data is the most viable solution. Currently, two broad indexing families exist: those that exploit some form of a tree structure, and those that are based on some 'distributed access on the air' mechanism. The latter family is the most popular and viable, because it allows for following 'air-pointers' without the need to first find a tree root. The champion method of the distributed air index is the *Exponential index* which however is not appropriate when the access pattern is skewed, i.e., some data items are more popular than the others. To address this shortcoming, we design a *Distributed Skip Air Index* (*DiSAIn*), which exploits access statistics in order to improve average tuning time, while it preserves the access latency equal to that of the original Exponential index. To attest the superiority of the proposed indexing method, we perform a detailed simulation evaluation of the two competing methods.

6.1 Introduction

A wealth of wireless networks is nowadays present such as cellular, WiMAX, vehicle-to-infrastructure, and so on. In all these types of communication systems, broadcasting is the fundamental operation that exploits the shared medium, i.e., the wireless channel, to transmit information to clients. The communication protocols in such architectures can be implemented either as pure pull-based, pure push-based [112] or on-demand broadcasting [113, 114]. In pure pull-based broadcast, the clients request information via an uplink channel, and subsequently the server allocates a channel for the requesting client, and transmits the information. In pure push-based broadcast, the server sends information over a common broadcast channel to all listening 'consumers' (clients). In on-demand broadcast, the clients pose requests to the server, and the server broadcasts the responses via a shared broadcast channel – thus a single response can satisfy multiple requests. Apparently, (pure-push and on-demand) broadcasting is a preferred choice for modern wireless networks, since

*The ideas presented in this chapter appear in the following publication [C.03] it overcomes the scalability issues associated with the large number of consumers and the large volume of transmitted data.

Consider for instance a data dissemination [115] application in an Infrastructure-to-Vehicle (I2V) case. In this push-based broadcast system, the RSUs broadcast information concerning issues relevant to the moving vehicles, e.g., traffic congestion reports, updated routing instructions for the vehicles of a fleet, and so on. Each RSU constructs a broadcast program with the needed info and broadcasts it periodically. All vehicles can tune in to the broadcast channel in order to retrieve the information (data items) without sending explicit requests for them, thus avoid 'choking' the uplink channel. Figure 6.1 illustrates such a scenario. A vehicle wants to retrieve data item B from a broadcast channel. The importance of knowing when item B will be broadcasted is crucial, because the vehicle can decide to accelerate to get it from the next RSU, or slow down to get it from the current RSU. The presence of *air indexes* would give an answer about the time of broadcast. Regulation of the vehicle's velocity and knowledge of the distance between successive RSUs could provide a suggestion to the driver about its driving policy.

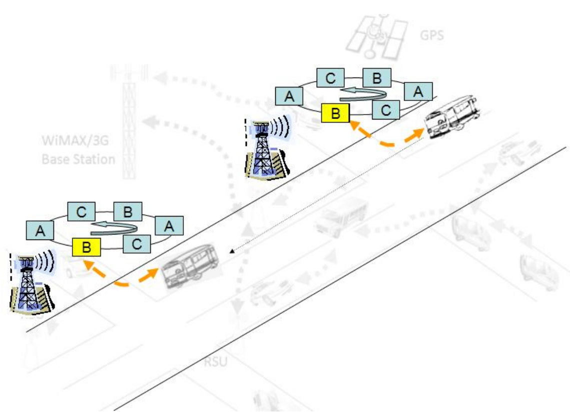


Figure 6.1: I2V type of communication.

Such information is important in an overall CO_2 emission control policy. The more accurate the information about the place of the desired data in the overall broadcast cycle is, the less alteration (acceleration or slow down) in the velocity of the vehicle is required in order to be in range of an RSU for the whole time period the desired data is broadcasted. Optimizing driving behavior patterns, such as idling, speeding, fast stops and hard braking, have a direct impact on fuel economy and CO_2 emissions which is a great challenge[†] that the transportation section faces today. Besides, such information is also useful for hybrid geocast routing protocols [116], for use by dead drops [117] and so on.

[†]http://www.reduction-project.eu/

In most situations the majority of the vehicles that move in a specified area tend to retrieve some particular data instead of showing equal preference to all data; therefore the skewness in data access is very high [118]. An index that help the listener find faster this information would definitely improve the driving behavior patterns.

To overcome that problem of 'blindly' searching among all transmitted data, the servers usually interleave along with the data, and some *index* packets, to help the consumers 'move' into the broadcasted information. The index packets contain the necessary information to guide the user through the transmitted data, until s/he reaches the information. These index packets play the role of the indices that we encounter into the traditional disk-based database systems. The difference between the two is that the broadcast channel is a *one-dimensional medium*, and the interleaving of index packets increases the *access time*, in an attempt to decrease the *tuning time*. These performance measures are typically used to measure the efficiency of a wireless data broadcast system; access time is the time elapsed from the moment a request is issued by a client to the moment the requested data is returned, and tuning time is the duration of time a client stays tuned in to collect requested data items. All indexing schemes attempt to achieve the best tradeoff between tuning time and access time.

6.1.1 Motivation

A lot of indexing schemes have been proposed in the literature (cf. Section 6.2 for a detailed survey); they can be categorized as *hierarchical tree-based* and *distributed* solutions. Examples of the first family include an adaptation of the idea of B^+-tree indexing in wireless environments [119], and application of signature trees as indexing methods [120]. B^+-trees and hashing are appropriate for uniform access patterns. To deal with skew access patterns, other tree-based methods include [121, 122]. Adaptation of such indexing schemes (e.g., B^+-tree) to work in multiple broadcast channels are described in [123, 124].

Distributed solutions such as the adaptation of the traditional hash-based indexing technique in wireless environments was earlier described in [125], and later was generalized in [126, 127], which can be seen as a distributed air-based implementation of a skip list. Hybrids between the two families are described in [128, 129]. In summary, any method that is (completely or partially) based on a broadcasted tree is inferior, compared to the distributed solutions, because the users can not start their search immediately after they tune into the broadcast channel.

In [126], Xu et al. proposed the exponential index [127], *ExpIn*. In *ExpIn*, each bucket contains a data part and an index table. The index table consists of a number of rows which varies according to the database size. Each entry indexes a segment of buckets with sizes that grow exponentially (i.e., $2^0, 2^1, 2^2, \cdots$). The first entry contains the next bucket ($2^0 = 1$) and for each $i > 1$ the i_{th} entry points to the segment of buckets that are 2^{i-1} to 2^i away (see Figure 6.2). Instead of the base 2, the algorithms can use any integer number, but this achieves an acceptable tradeoff between access time and tuning time.

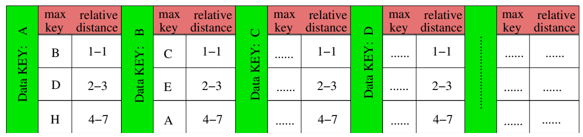


Figure 6.2: Exponential index structure for a sample database consisting of the elements 'A'–'H'.

In [126, 127] it was shown that the *exponential index* (*ExpIn*) is superior to the hashing schemes of [125] (and to the tree-based schemes), and currently this indexing scheme is the state-of-the-art for broadcast indexing. However, the exponential index is able to achieve good tuning time only when the broadcasted data have equal probability of access. In real life applications though, some items are more frequently accessed by clients than some others (the less popular ones); i.e. a *skewed access pattern* [130] is prevalent in the real life. This feature makes *ExpIn's performance degrades proportionally to the skewness of data*, and it comprises a motivation of the present work.

6.1.2 contributions

We develop an indexing scheme for a single broadcast channel, suitable for skewed access patterns that is distributed in its nature, i.e., no matters when a user tunes into the broadcast channel, s/he can immediately start the searching for the desired item. The research work presented in this chapter makes the following contributions:

- It develops a broadcast index, namely the *Distributed Skip Air Index (DiSAIn)*, which exploits the different access probabilities of data items, in order to improve the mean tuning time, while retaining the same access time as that of the exponential indexing scheme.

- It evaluates experimentally the performance of the proposed broadcast index, against the Exponential Index, for several database sizes and access probabilities distributions.

The rest of this chapter is organized as follows: Section 6.2 briefly survey the most relevant work; Section 6.3 describes the network model, i.e., any assumptions made in the present work; Section 6.4 presents the Distributed Skip Air Index; Section 6.5 presents the simulation environment, the experiments and obtained results. Finally, , and Section 5.8 concludes the chapter.

6.2 Relevant work

Disk-based indexing (e.g., B^+-trees, Hashing, Skip Lists) for traditional as well as for advanced applications is a thoroughly investigated area during the past years. An adaptation of the idea of B^+-tree indexing in wireless environments was first described in [119], where instead of the disk addresses the leaves of the B^+-tree store the arrival time of each datum in the broadcast channel. Similarly, an adaptation of the traditional hash-based indexing technique in wireless environments was earlier described in [125], and later was generalized in [127]. Hybrids between the two approaches are described in [128, 129] and application of signature trees as indexing methods in reported in [120], which of course can support only equality queries. Adaptation of such indexing schemes (e.g., B^+-tree) to work in multiple broadcast channels are described in [123, 124]. They do not propose new schemes but simply different allocation methods for the nodes of the indexing tree. In all these works it is assumed that: a) there is a global ordering among the transmitted data, and b) the access pattern is uniform, that is, the access probability is the same for all data, which is quite unrealistic.

Deviating from the uniform access probability assumption, several works considered the effect of access skew on the design of indexing schemes. A new scheme is proposed in [122], which is a k-ary version of the basic binary Alphabetic Tree [131] over the data, whereas [132, 124] adapted the indexing method of [119] to deal with non-uniformity in access. Various methods were based on the construction of a binary or k-ary Alphabetic Tree to develop indexing schemes for multiple broadcast channels [133, 134, 135]. These methods do not provide new types of tree-structured indices, but rather a new allocation method for the tree-structured method of Alphabetic trees to the multiple channels. All these works [133, 134, 122, 132, 124] assume that: a) there is a global ordering among the transmitted data. There are only a couple of works [136, 121], which deviated from both the uniformity and global ordering assumptions. Finally, remotely related to the present work are the broadcast indexing techniques, e.g., for multidimensional data [137].

6.3 Network model

We will abstract our application area with the generic model which is depicted in Figure 6.3. We will use the term 'client' to any device, e.g., vehicle, smart phone, that is interested to get information from the broadcast channel.

We will consider a single channel wireless broadcast environment with the following generic properties:

- The broadcast schedule is push-based.

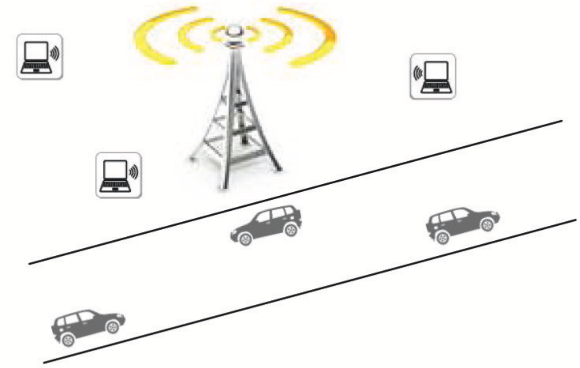


Figure 6.3: Generic architecture of broadcast-based wireless networks.

- Data access patterns are skewed.
- All data items are of equal size.
- Time on the broadcast channel is divided into slots where each slot can accommodate one data item.
- The server broadcasts data items in the range of 1 to database size.
- The server retains access statistics.
- Clients require only one item at one time, i.e. a single-item query.
- The client after acquiring the desired data, remains idle for an exponentially distributed time period before next data request.
- Initiation phase takes place every N broadcast cycles in order to stay updated to variations to data access probability distribution.

6.4 The Distributed Skip Air Index (*DiSAIn*)

The objective of the *DiSAIn* index is to address the problem of skewed access probabilities observed in real life applications and the appropriate data dissemination in VANETs. To achieve this, it adds more 'pointers' to the original *ExpIn* index, but it retains some of the properties of the *ExpIn* index: each bucket of the broadcast cycle contains a data bucket and an index table. The index table consists of i entries. Each entry indexes a range of buckets that are 2^{i-1} to $2^i - 1$ buckets away and holds the maxkey value of these buckets. But for the last bucket, the *DiSAIn* maintains another index, the skewed index SI, that points to the most popular element MP of the segment. The construction of the index table takes place in an initiation phase where the distance of MP from the $maxkey$ value of the last segment is calculated for each data element. In case where all data elements of the last segment have equal access probabilities, then the *DiSAIn* indexes one of these elements at random.

Figure 6.4 illustrates a sample *DiSAIn* index; it is supposed that the elements 'A' to 'H' are to be indexed, and that the element 'F' is the most popular one. In general, one fourth of the total number of pointers point to this 'hot' element. If the client tunes into the broadcast channel just before item 'A', then it retrieves the index table that corresponds to the bucket of 'A'. In case where the client issues a query for item 'F', a pointer points to it directly, minimizing the tuning time while keeping the access latency constant.

The effectiveness and efficiency of the *DiSAIn* index is not only based on the direct indexing of the most popular item(s), which creates huge earnings, since in real environments most of the clients retrieve hot items. Even in cases where clients are not interested in the popular ones, this additional pointer shortens the tuning time, since it provides evidence about the relative position of other items. For instance, looking at the index that corresponds to element 'A', the client can deduce that between elements 'F' (five places away) and element 'H' (seven places away) there is one element in-between them, therefore this is the element 'G'.

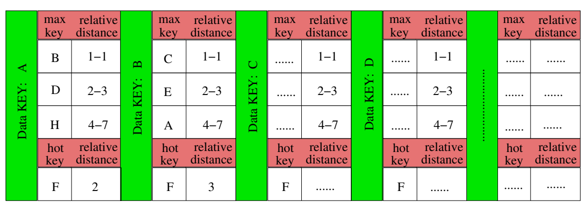


Figure 6.4: A small example of the *DiSAIn* indexing strategy.

In the previous discussion, we described the *DiSAIn* index as a plain addition of one pointer to the *ExpIn* index. Actually, the real flesh of the index, is the addition of more pointers towards the rest of the popular items. Clearly, there is a tradeoff and an optimal value of added pointers. We refer to this version of the index as the *Enhanced DiSAIn*. We do not examine further this generalized version but leave it for possible future research.

6.5 Evaluation

For the evaluation of the proposed indexing scheme against the state-of-the-art broadcast indexing method, i.e., *ExpIn*, we developed a simulation model where several clients (50) access the data served by a server through a broadcast channel. The dataset size varies from 256 to 2048 items. We assume that the server has prior knowledge of access profiles or it employs statistical methods to estimate access statistics similar to [138].

6.5.1 Simulation setup

For the evaluation of the method we developed a system where a node (client) requests data according to a probability distribution. Flat broadcast is employed to broadcast the data items. The data are served by a server through a broadcast channel. After the client acquires the desired data it waits for an exponential distributed time interval with mean time T before it generates another request. Thus, it is a blocking client. For every system configuration, we run 50 experiments of 5000 queries with different popular items, and we compute the average access latency and average tuning time for every experiment. We implemented a Zipfian model for item popularity distribution with parameter θ:

$$p_x = \frac{(1/x)^\theta}{\sum_{x=1}^{n}(1/x)^\theta} \qquad 1 \leq x \leq n \tag{6.1}$$

where θ controls the *skewness* of the access distribution for n items. For $\theta = 0$ the Zipfian reduces to the uniform distribution, whereas larger values of θ derive increasingly skewer distributions. Using this formula, we can derive relative popularities for items. As performance measures, we used the tuning time and access latency. It has to be noticed here, that time is measured according to the internal simulator clock and does not correspond to actual seconds; it only the relative timing performance of the contestants that counts and not the absolute time difference (for instance, in Figure 6.6 we see that the actual difference in tuning time on the leftmost plot is $5.25 - 4.35 = 0.8$, but the relative percentage is $\frac{0.9}{5.2}\% = 17\%$).

6.5.2 Energy consumption model

We assume two states of energy consumption: doze mode and active mode. E_s describes the amount of energy consumption in an energy state s per unit time. The average energy consumption can be measured by the amount of unit energy in a given time. It simulates the amount of time that a client remains tuned into the broadcast channel. The result of a query processing contains unnecessary active mode time units, because the client has to follow more pointers in order to retrieve the desired data. Note that the frequent alternation between the active and the doze by turning “on” and “off” electronic circuitry may incur additional energy consumption. However, as circuit designers become more concerned about reducing energy consumption, switching energy will become less dominant and in our experiments we don’t take it into account. The energy consumption in doze mode is about

1000 times less than that in active mode [139]. In order to evaluate energy consumption we set:

$$E_{active} = 1000 \text{ and } E_{doze} = 1 \text{ energy unit.}$$

6.5.3 Experimental results

Impact of the number of pointers.

The main idea of the Distributed Skip Air Index as described in section 6.4 is to add an extra pointer in the most popular item of the last bucket. This simple feature leads to a significant improvement in mean tuning time in cases where the distribution of data queries is somewhat skewed. We sketch here the impact of the addition of more pointers, and specifically the addition of one more pointing to the second most popular item of the last bucket. The results in Figure 6.5 attest that indeed there are gains. The same conclusion is easily reached in cases where a second pointer is added which points to the most popular item of the $se - 1$ bucket, and so on. In the rest of the chapter, we do not evaluate this version of the index further, but deal with the simpler *DiSAIn* index.

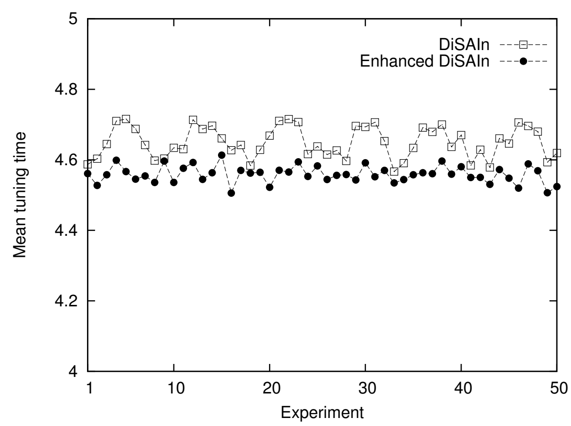


Figure 6.5: Mean tuning time for *DiSAIn* ($n = 256, \theta = 1.0$).

Impact of skewness in the access pattern.

In Figure 6.6 we observe the impact of skewness on the tuning time of the competing algorithms. Consistently with the theory, when the access pattern is uniform (leftmost plot), the *ExpIn* index performs as good as the *DiSAIn*. But, as the θ increases the performance gap between the two methods widens, and the *DiSAIn* index is the improving its behavior in contrast to *ExpIn*. These observations are consistent across various numbers of items.

Performance w.r.t. the mean access time.

We evaluated the performance of the methods with respect to the access time (see Fig-

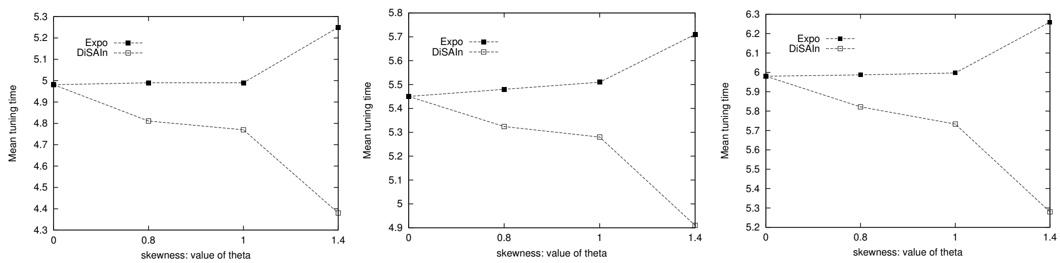


Figure 6.6: Impact of skewness on mean tuning time (database size 256, 512, 1024).

ure 6.7). We observed that *DiSAIn* has no significant negative effect in mean access time. The addition of extra pointer(s) in the indexing scheme increases the size of each item leading to a total increase in mean access time. In case of one extra index to the most popular item of the last bucket, the increase in the size of each bucket is estimated according to:

$$s' = \frac{s_{ne}}{s_o + s_e}, \tag{6.2}$$

where s_o is the size of a data item, s_e the size of the original exponential index and s_{ne} is the size of the extra index. In a typical system of 1024 data items with $s_o = 128$ bytes and $s_e = 4$ bytes the extra index is approximately 0.4 bytes leading to a total increase of each bucket and of average access latency of 0.3%, which is really negligible. Similar results were obtained for various values of n and θ.

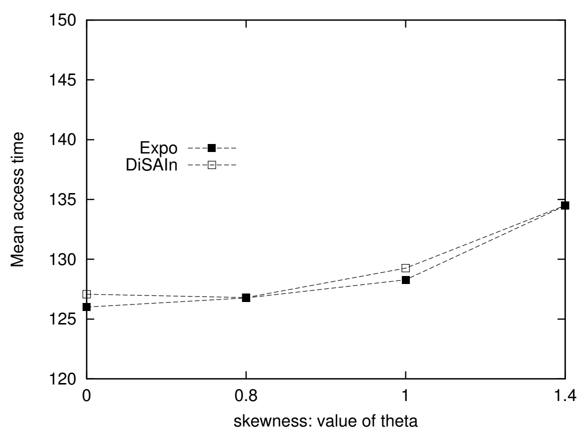


Figure 6.7: Mean access latency ($n = 256, \theta = 0.5$).

Performance w.r.t. the database size.

The database size plays a significant role in the performance of the exponential index. As the number of items increase, the effect that skewed data have in system performance increases also. From Figure 6.8, we observe that when the number of broadcasted items gets really large, then the performance gap in tuning time between the two competing indexing schemes broadens. This is expected since the *ExpIn* index needs to follow more pointers in order to retrieve the desired data.

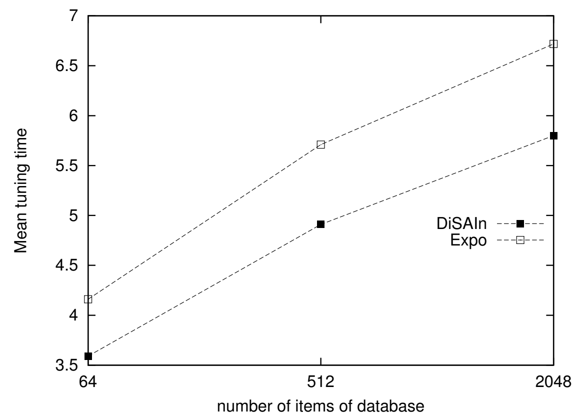


Figure 6.8: Impact of number of items on mean tuning time:($\theta = 1.4$).

Performance w.r.t. energy consumption.

We evaluated the performance of the methods with respect to the energy consumption. We computed access latency and mean tuning time (Table 6.1).

$$E_{active} = 1000 \text{ and } E_{doze} = 1 \text{ energy unit.}$$

θ	*Access latency*		*Tuning time*	
	Expo	*DiSAIn*	*Expo*	*DiSAIn*
0.0	255.35	255.35	5.48	5.48
0.8	255.14	256	5.5	5.37
1.0	256	257	5.5	5.23
1.4	271	268	5.78	4.9

Table 6.1: Access - tuning time ($n = 512, \theta = 0.0, 0.8, 1.0, 1.4$).

From Table 6.1 we can compute active and doze time units and energy consumption (Table 6.2). Active time units equal mean tuning time while doze time units are computed using the equation :

$$|N_{doze}| = |accesstime| - |tuningtime| \tag{6.3}$$

theta	*Doze*		*Active*		*Energy*	
	Expo	*DiSAIn*	*Expo*	*DiSAIn*	*Expo*	*DiSAIn*
0.0	249.87	249.87	5.48	5.48	5729.87	5729.87
0.8	249.64	250.63	5.5	5.37	5749.64	5620.63
1.0	250.5	251.77	5.5	5.23	5750.5	5481.77
1.4	265.22	263.1	5.78	4.9	6045.22	5163.1

Table 6.2: Active-doze time unit, energy consumption ($n = 512$).

It is obvious that *DiSAIn* is more efficient in terms of energy consumption for all experiments conducted. As the skewness of data increase *ExpIn* becomes energy inefficient while on the other hand *DiSAIn* consumes less and less energy making it the ideal choice for such environments.

6.6 Chapter conclusions

We investigated the issue of indexing broadcast information under skewed access patterns. Even though there are a lot of broadcast indexing schemes in the relevant literature, our approach fills the gap by proposing a distributed index which has no root, and therefore the clients can start searching for information without waiting for a 'designated root'. The main idea of the proposed Distributed Skip Air Index is to add some pointers to the most popular items so that the clients can locate them quickly. We implemented a simulation environment to investigate the performance of the proposed scheme and presented detailed experiments that assess the superiority of the method, clearly outperforming the state-of-the-art distributed broadcast index.

Pull-based data dissemination mechanism is one other mode of communication in VANETs. With this mechanism any vehicle is enabled to query information about specific location or target. Vehicles provided with updated information regarding road traffic conditions can take proper directions in order to avoid being trapped in heavy traffic jams. This information can be retrieved from RSUs scattered at intersections, which can route vehicles to proper directions according to their final destination and the current road conditions.

Chapter 7
Pull-based data dissemination mechanism for congestion avoidance*

In the overall effort of reducing CO2 emissions especially in large cities vehicular communications can play an important role. Intelligent transportation systems, which aim to use information and communication technology are considered to be a major factor in this effort. Eco-routing is already used to suggest most environmental-friendly routes in order to reduce overall mileage and CO2 emissions based on historical data. We propose a real time system based on DSRC communication capabilities in order to reroute vehicles to the most ecological route, avoiding congested roads and minimizing the overall travel time and C02 emissions.

7.1 Introduction

Traffic congestion is a problem modern cities have to face; it costs time, fuel and, thus, money. Extending the road network is not suited to fight congestion due to spatial, financial and environmental constraints. Recent progress in the area of information and communication technology, however, promises to make today's transportation systems not only more efficient, but also safer, more reliable and more convenient. Vehicular Ad Hoc Networks (VANETs) are considered a central part of these Intelligent Transportation Systems (ITS). VANETs enable many actors in traffic (e.g., vehicles, traffic lights or road side units) to exchange information and to coordinate their driving behavior. As no or little underlying infrastructure is required and message exchange is carried out with low latency times, VANETs are an excellent tool to reduce congestion in the context of ITS.

By using inter vehicle communication vehicles can collect traffic jam information by calculating the approximate arrival time of vehicles at any location proposed in [140]. However, these types of services are already available in 3^{rd} generation mobile systems but these services are very costly and are not available in the infrastructure less environment or infrastructure damaged environment. The cost of vehicular ad hoc networks is very high but the facilities obtained on traffic safety, commercial applications given in [141, 142, 143] can show the requirement of VANETs.

Eco-routing is a mechanism used to suggest environmental-friendly routes in order

*The ideas presented in this chapter appear in the following publications [C.07,S.04] to reduce overall mileage and CO2 emissions based on historical data. It is well known, however, that finding the optimal route is a complex and often daunting task, as re-routing decisions, whether due to accidents or traffic jams, may result in further jams and increased driving distance. A small fraction of additional traffic may lead to further road congestion, and therefore to additional micro jams and even much longer, secondary jams [144]. In order to alleviate this effect, optimization problems (through the use of inter-vehicle communications) should be formulated and solved. These optimization problems could include a set of building blocks, or an area of a city [145].

7.2 Fuel efficiency and CO2 reduction approaches

Applications of ICT(Information and Communication Technologies), such as cruise control, platooning, and traffic signal management, along with driver's behavior promotion and vehicle and infrastructure improvements can help to promote fuel consumption and CO2 emission reduction. Approaches that are based on inter vehicle communications can be divided in four major categories.

Cruise Control

A study of traffic flow improvement by utilizing vehicular communications for CACC has been done in [146]. The proposed CACC (Cooperative Adaptive Cruise Control) utilized communication technology as well as position systems to enhance the overall performance of the application by reducing an impact of a traffic shock wave on the flow of traffic since it enables anticipatory braking actions. This traffic shock wave is a major cause of traffic congestion. With the use of inter vehicle communications, upstream vehicles do not have to brake as severely when a down stream vehicle brakes. In [147], CACC was proposed with direct communication with the preceding vehicle only. Performance of CACC was also investigated in [148].

Platooning

Vehicle platooning is also one of the promising applications in order to provide fuel consumption and CO2 emission reduction. In [149], the authors focused on the platooning in automated highway systems (AHSs). Other platooning management methods have been proposed in [150].

Traffic Signal Management

TLVC (Traffic-light to-vehicle communication) is a novel idea investigated in a number of recent research papers. The authors in [151] proposed new eco-friendly routing protocols. In [152] a solution that uses intelligent traffic lights, mobile devices and wireless communication to reduce car emissions is presented.

IVC communications for Eco-routing

In [153] an event-driven architecture (EDA) is studied, as a mechanism for detecting traffic jams. The EDA can detect several types of traffic jams. CoTEC [154], is a novel co-operative and distributed V2V mechanism to efficiently detect traffic congestion. Innovative eco-routing methods based on V2V communications are proposed in [155].

We propose a novel architecture for CO2 reduction, based on DSRC technology. The method combines all modes of communication - infrastructure to vehicle(I2V), vehicle to infrastructure (V2I) and infrastructure to infrastructure(I2I) - in order to perform eco-routing of vehicles that travel in an urban environment.

7.3 Contributions

The present work presents a new congestion avoidance routing protocol for VANETs. Several parameters are investigated in our scenarios in a Urban environment. The research work presented in this chapter makes the following contributions:

- A new IVC eco-routing method for Urban vehicular environments, the $ErouVe$, is described.
- I2V, I2I and V2I communications are combined in order to take appropriate routing decisions.
- $ErouVe$ exploits information gathered by RSU's from passing vehicles in order to solve a unique optimization problem for every vehicle.
- A performance evaluation of the proposed method against a baseline method ($Shortest-Path$) is conducted, which attest the superiority of the new structure.

7.4 system description

We consider a network $G = (N, L)$, where N is the set of nodes (intersections - RSUs) and L is the set of links (road segments). Road segments that are adjacent to RSU n belong to the set $S(n)$. $M(l)$ is the subset of vehicles that have traversed road segment l the last s seconds. Each RSU calculates average values for each road segment l that is adjacent to its intersection and sends this information to adjacent RSUs. RSUs in order to exchange such information - mean travel duration, mean CO2 emissions - communicate with each other through beacon messages that are exchanged every s seconds.

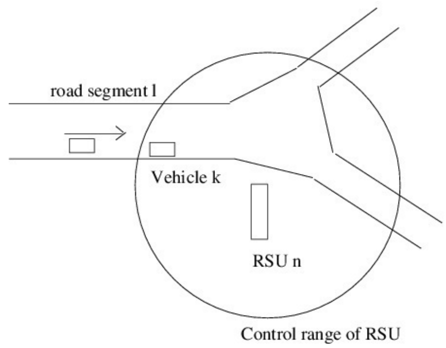


Figure 7.1: Decentralized CO2 reduction system based on DSRC communications

7.4.1 Initialization phase

In the initialization phase each RSU n computes centrally the distance D_{nm} from every other RSU m using Dijkstra's algorithm, based on GPS data. No time cost or CO2 cost is initially calculated for the road segments. According to this data, each RSU is aware of its adjacent RSUs, and the road segments through which they are connected. Table I represents the structure of information kept by each RSU for I2I communication.

A/A	*Adjacent RSU*	*Road segment*
1	RSU_k	S_n
2	RSU_l	S_o
3	RSU_m	S_p

Table 7.1: Connection table of RSU n.

Each vehicle that enters the simulated area follows the shortest path to its destination ignoring time or ecological parameters.

7.4.2 Communication phase

For vehicle k (V2I):

As vehicle k, traveling on road segment l, enters in the control range (c.f. section 7.6) of the intersection, RSU near the boundary of the control range impels this vehicle to:

- calculate total time traveled TT_{lk} and C02 emitted C_{lk} on the road segment l. Parameter C_{lk} is divided to the distance that vehicle k traveled along road segment l.
- send TT_{lk} and C_{lk} to RSU

- send destination d to RSU
- ask RSU to choose the next road segment to follow, sending an R_q message.

For RSU at intersection n (V2I):

Each RSU receives from approaching vehicles parameters TT_{lk}, C_{lk} and calculates average values for each road segment l that is adjacent to the intersection n, taking into account only values from the last s seconds in order to have updated information.

$$TT_l = \sum_{k \in M(l)} TT_{lk} \quad and \quad C_l = \sum_{k \in M(l)} C_{lk} \tag{7.1}$$

For RSU at intersection n (I2I):

Each RSU sends the accumulated values of mean travel time and mean CO2 / distance to adjacent RSU's every s seconds through the use of beacon messages.

For RSU at intersection n (I2V):

Each RSU, after solving the optimization problem(c.f. next section), sends routing instructions to vehicle via R_a message (route answer).

The main communication phases of the system are demonstrated in Figure 7.2.

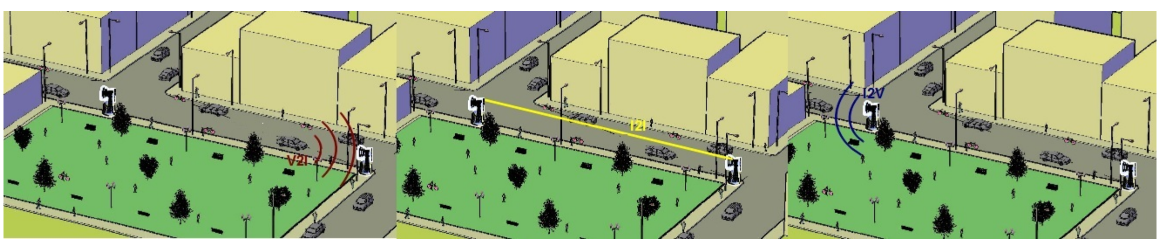


Figure 7.2: Application example of $ErouVe$ System. A. Vehicle approaching intersection sends a beacon message to RSU B. Infrastructure-to-infrastructure communication for information exchange C. Vehicle receives routing instructions.

7.5 $ErouVe$ algorithm

After introducing the basic communication, we present the Ecological Routing of Vehicles ($ErouVe$) algorithm applied in a traffic system. The proposed system, in order to eco-route vehicles, solves a decentralized optimization problem.

After receiving the route request message - R_q message - from a vehicle k, RSU solves an optimization problem in order to guide the vehicle to a more green route through the road network.

Next road segment selection algorithm

For each segment l' adjacent to the current road, that the vehicle k is currently on, assign a local weight $w_{l'k}(t)$ according to $TT_{l'}$, $C_{l'}$ and whether or not it helps the vehicle k reach its destination

$$w_{l'k}(t) = (w_1 * TT_{l'}) + (w_2 * C_{l'}) + (w_3 * A_{l'}) \tag{7.2}$$

, where parameter A indicates if the road segment l' is closer to the destination d of vehicle k or not according to equations 7.3, 7.4.

$$A_{l'} = 1/(D_{ld(k)} - D_{l'd(k)}), \quad D_{ld(k)} > D_{l'd(k)} \tag{7.3}$$

$$A_{l'} = D_{ld(k)} - D_{l'd(k)}, \quad D_{ld(k)} \leq D_{l'd(k)} \tag{7.4}$$

Then, define the road segment l'' with the minimum weight:

$$w_{l''k} = \min_{l' \epsilon S(n)} w_{l'k}(t)$$

s.t.

$$A_{l''} < D_{th}$$

Parameters w_1,w_2,w_3 are used in order to focus to one of the different optimization parts, e.g. time, distance or CO2 emissions. In the default system settings, all optimization parts have the same significance. If w_1 and w_3 take very low values(e.g 0.001 or 0) then only CO2 (ml/m) is used as a routing parameter, which could lead cars to follow green but maybe too long routes. This would have a direct effect in the system's performance since total CO2 emissions would not decrease and moreover cars would have to travel more time in order to reach their destinations. The opposite happens if only time is used in order to select the next road to travel. Cars in this scenario would follow the fastest road (in terms of mean velocity) neglecting how far from their destination this road would lead them and how much CO2 they would emit when traveling on it. All three parameters are crucial in order to keep cars on logical paths and minimize CO2 emissions and need to be combined when calculating weights of roads.

After solving the optimization problem, RSU sends routing instructions to each vehicle with a *Ra* message. Routing instructions are given to vehicles at each intersection in order to achieve traffic load balance, total CO2 and travel time reduction. The algorithm runs in an autonomous way at each intersection.

7.6 Simulation

We use the simulator Veins [144]. Veins is an open source framework for running vehicular network simulations. It is based on two well-established simulators: OMNeT++, an event-based network simulator, and SUMO, a road traffic simulator. [57]. In order to calculate CO2 emissions we use the EMIT model integrated in Veins. EMIT is a simple statistical model for instantaneous emissions and fuel consumption for light-duty composite vehicles based on speed and acceleration. For our experimentation we use the 'Category 9' -e.g Dodge Spirit 1994 - vehicle [156] implemented in VEINS.

7.6.1 The Veins Framework

Veins is a simulation environment that is based on OMNeT++ for event-driven network simulation and SUMO for road traffic microsimulation [57].

Realistic communication patterns of VANET nodes are modeled with the help of OMNeT++, a simulation environment free for academic use, and its INET Framework extension, a set of simulation modules released under the GPL. OMNeT++ runs discrete, event-based simulations of communicating nodes on a wide variety of platforms and is getting increasingly popular in the field of network simulation.

Scenarios in OMNeT++ are represented by a hierarchy of reusable modules written in C++. Modules relationships and communication links are stored as Network Description (NED) files and can be modeled graphically. Simulations are either run interactively in a graphical environment or are executed as command-line applications.

The INET Framework provides a set of OMNeT++ modules that represent various layers of the Internet protocol suite, e.g. the TCP, UDP, IPv4 and ARP protocols. It also provides modules that allow the modeling of spatial relations of mobile nodes and of IEEE 802.11 transmissions.

Traffic simulation is performed by the microscopic road traffic simulation package SUMO. Developed by German research organizations DLR and ZAIK, this simulator is in widespread use in the research community, which makes it easy to compare results from different network simulations. Availability of its C++ source code under the terms of a GPL license made it possible to integrate into the simulation core all needed extensions.

SUMO performs simulations both running with and without a GUI and can import city maps from a variety of file formats including the freely available OpenStreetMap data. It thus allows high-performance simulations of huge networks with roads consisting of multiple lanes, as well as of intra-junction traffic on these roads, either using simple right-of-way rules or traffic lights. Vehicle types are freely configurable with each vehicle following statically assigned routes, dynamically generated routes, or driving according to a configured timetable. Both simulators have been extended by modules that allow the road traffic simulation to communicate with its counterpart via a TCP/IP connection. The current version of Veins integrates with the newly-created TraCI interface to SUMO. Using this interface, a connected network simulator is able to send a series of commands to individual vehicles, influencing their speed and routes. During the simulation, at regular intervals, the manager module triggers the execution of one timestep of the road traffic simulation, receives the resulting mobility trace and triggers position updates for all modules it had instantiated. Special mobility modules contained in vehicles modules process and act upon these updates.

7.6.2 Emissions and Fuel Consumption Model

Emission models can broadly be categorized into macroscopic, mesoscopic, and microscopic models. Macroscopic models estimate pollutant emissions and fuel consumption mainly based on the average travel speed of the traffic flow. These macroscopic models entail enormous simplifications on the accuracy of physical processes involved in pollutant emissions, which leads to reduced accuracy in calculations. Moreover, these models cannot capture individual speed fluctuations and cannot take into account individual operation conditions, which are of crucial interest when analyzing a proposal that considers network and vehicle dynamics. Mesoscopic models use more disaggregate trip variables, such as the average speed, the number of stops, and stopped delay, to estimate a vehicles emission rates on a link-by-link basis. Some regression models that were developed were found to predict fuel consumption and emission rates of Hydrocarbons (HC), Carbon Monoxide (CO), and NOx to within 88%90% of instantaneous microscopic emission estimates.

Microscopic emission models overcome some of the limitations of large-scale macroscopic models mainly by considering individual vehicles dynamics and their interactions. Emissions and fuel consumption are estimated based on instantaneous individual vehicle variables that can frequently be obtained (e.g., second by second) from a microscopic traffic simulator or another alternative source [e.g., the Global Positioning System (GPS) data logger]. Commonly, these parameters are divided into the following two categories: 1) vehicle parameters and 2) traffic/road parameters. Vehicle parameters include, among others, vehicle mass, fuel type, engine displacement, and vehicle class. On the other hand, network parameters (traffic and road conditions) account for instantaneous vehicle kinematics (e.g., speed or acceleration), aggregated variables (e.g., the time spent in the acceleration mode), or road characteristics (e.g., road grade). Because microscopic emission and fuel consumption models have higher temporal precision and better capture the effects of vehicle dynamics/interactions, they are better suited to evaluate the environmental gains derived from an ITS measure, such as the VTL system.

Several microscopic models have been proposed by the scientific community. These models can be classified into emission maps (speed/acceleration lookup tables), purely statistical models, and load-based models. EMIT is a simple dynamic emission model that was derived from statistical and load-based emission models. It takes as input instantaneous velocity v_t (m/s) and acceleration a_t (m/s^2) and predicts the fuel consumption rate f_t (g/s) at time point t.

7.6.3 Simulation parameters

The competitor is the $Shortest-Path$ routing algorithm, where each vehicle follows the path with the least total distance to the destination. In our simulation, we consider various road traffic and network data parameters. For the evaluation of our method we used synthetic road networks.

The simulation environment (Figure 7.3) is a one direction road about 2km long with two available paths. The upper part is longer (upper part: 275 meters, lower part: 190 meters), while the lower part is shorter. Both parts have the same capacity in cars (2 lanes). These parts merge at junction 2, where the upper part can occupy two lanes of the next 3 lane road segment and the lower part can occupy only one lane. This setup is used in order to demonstrate a typical city scenario, when one lane can be temporarily blocked due to a stopped car or an accident. Similar situations, where traffic is accumulated in the last part of a road segment, arise when a road intersects with another with higher priority or when a traffic light exists. Any of the above cases combined with medium traffic, make a road segment that seems to be the best choice (the lower road segment in our simulated scenario) incapable to accommodate all the vehicles and major traffic congestion instances occur. Updated information of the road conditions, which are collected by the incoming cars can effectively - in most cases - alleviate traffic congestion incidents.

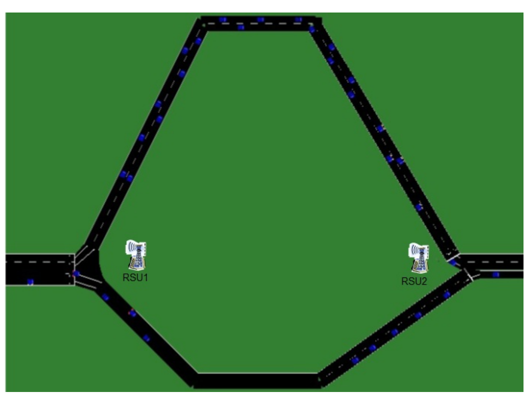


Figure 7.3: Simulation environment

All nodes are equipped with GPS receivers and On Board Units (OBU). Location information of all vehicles/nodes, needed for the routing algorithm is collected with the help of GPS receivers. The communication paths are available via the ad-hoc network and the RSUs, which are scattered in every junction in the simulation environment. The communication range of both vehicles and RSUs is set to 300 meters. Every RSU has three important ranges, when communicating with any approaching vehicle:

- *The communication range*: This is the communication range that can be achieved between RSUs and vehicles according to the setup of the system. For our simulations the communication range is set to 300 meters.
- *The handshake range*: this is the range after which the approaching vehicle enters the area of the RSU. At this point the vehicle stores the position of the RSU, using the information of the packets the RSU sends. In our simulation the handshake range is set to 100 meters.
- *The control Range*: The control range is the distance where the vehicle receives the message of the RSU with the rerouting instructions. This range is the closest to the RSU but is set to a medium value in order to give time to the vehicles to perform rerouting, if necessary. In our simulation the control range is set to 50 meters.

7.6.4 Evaluation criteria

To demonstrate the benefits of the $ErouVe$ system, a number of variables need to be analyzed. In this section, we want to analyze the environmental impact of $ErouVe$, comparing the individual and aggregated CO2 emissions of vehicles that travel towards the same destination with and without the $ErouVe$ system. Since vehicles have to take one of the two available paths, when approaching the first junction, the main difference in their travel would be the upper or the lower part of the road between junctions (called **area of interest** from now on), while the rest of their trip would be the same as far as distance is concerned. For that reason the individual performance of each vehicle in these two road segments (upper and lower) is evaluated in this section. Mean aggregated values of the system's performance parameters, e.g. time and CO2, are also compared for both the area of interest and for the complete route of the vehicles.

Parameters w_i are set to their default values 1 for this simple simulation scenario. Performance evaluation of the system according to the values of these parameters is an open issue for investigation. Optimal values for these parameters for any topology, is a matter of discussion. Parameter s has a direct impact on the performance of $ErouVe$, since it affects the accuracy of the aggregated values. Values from 30 seconds to 3 minutes were used in order to capture this impact. Finally two scenarios were investigated with 45 and 90 vehicles representing a medium and a high dense situation in an urban environment. The communication range is set to 300 meters.

Independent parameter	*Range of values*	*Default value*
$Number\ of\ Vehicles$	45, 90	45
$Parameter\ s(sec)$	30, 200	30

Table 7.2: Simulation parameters.

For this analysis, the definition of each variable whether for individual vehicle analysis or aggregated calculations is given as follows:

- Route (area of interest) CO2 emissions per vehicle k (individual; in grams): cumulative sum of an individual vehicle CO2 emissions.
- Route (area of interest) Travel time per vehicle (individual; in seconds).
- Average CO2 emissions per vehicle / distance traveled (aggregated; in grams/m).
- Average velocity per vehicle (aggregated; in meters/second).

Individual Vehicle Results

CO2 emissions per vehicle

In Figure 7.4, CO2 emissions (ml) of each vehicle are demonstrated. $ErouVe$ outperforms $Shortest - Path$ almost for every vehicle, though vehicles have to take longer paths. This is due to the fact that at the exit of the lower part of the map, vehicles can occupy only one lane at the next road segment, leading to road congestion and thus increasing CO2 emissions and travel time of each vehicle in the specific road segment. $ErouVe$ alleviates such traffic congestions by rerouting vehicles to the upper road segment -and thus splitting traffic- when traffic grows at the lower shortest segment. Decrease of more than 50 % in individual CO2 emissions of vehicles during traveling the area of interest are observed.

Travel time per vehicle

Similar to CO2 emissions, travel time is also better when cars are rerouted by the least congested, yet longer, road (see Figure 7.5). The deviation in time is smoothened when vehicles are smartly rerouted and and thus the travel time of most vehicles is reduced. With the use of the $ErouVe$ system, travel times vary from 10 to 20 seconds (for the area

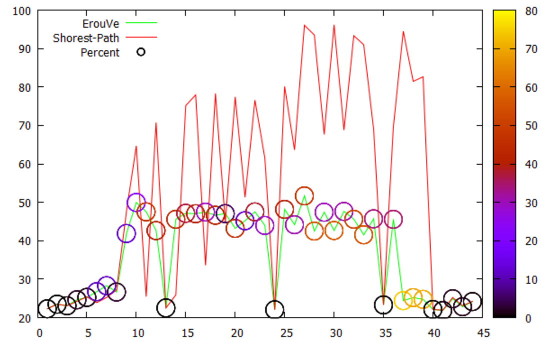


Figure 7.4: CO2 emissions per vehicle

of interest), while when vehicles follow the $Shortest - Path$, in terms of distance, the time that a vehicle needs to cross this segment of the road can be up to 70 seconds.

Although $ErouVe$ manages to keep travel time of all vehicles relatively low, some vehicles that use the $ErouVe$ system take longer times to pass the area of interest. This is due to the fact that although traffic jam exists in the lower part of the road, some vehicles manage to pass through it in a rather short time due to the lane they happen to occupy.

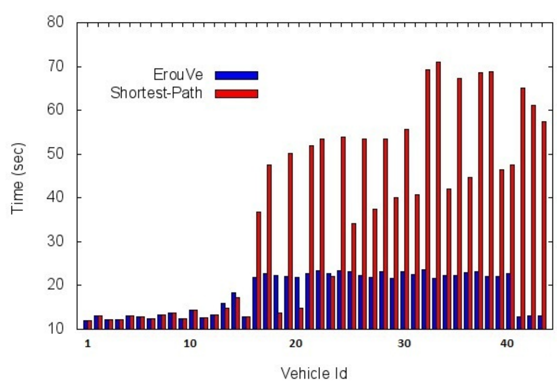


Figure 7.5: Travel time per vehicle

Parameter s

Updated information of traffic conditions play a significant role in the performance of $ErouVe$. In Figure 7.6 we see that when parameter s is relatively long (3 minutes), accumulated values for CO2 (CC_l) and time (TT_l) do not represent realistic data of the road conditions, rendering $ErouVe$ inefficient. The performance of our method is then degraded and the benefits are lost. A relatively small value that keeps data up to date is an important factor for the optimal performance of the method.

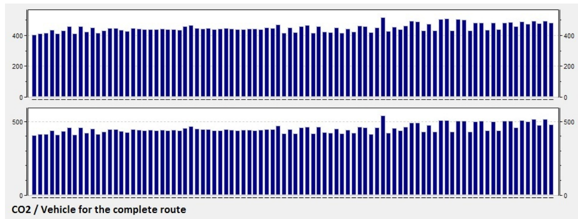


Figure 7.6: Parameter s affects CO2 emissions : Upper diagram s=30 sec, Lower diagram s =170 sec.

Aggregated Results

The density of the cars play a significant role in congestion management methods. Tables 7.3 and 7.4 show aggregated performance values for different number of vehicles. We see that for different vehicle densities, $ErouVe$ outperforms $Shortest-Path$ for every investigated parameter of the system's performance e.g. CO2, CO2/m, time. When all vehicles follow the same shorter path the capacity of the road cannot satisfy the instantaneous flow. This is a typical bottleneck effect where the section of a route with a small carrying capacity produces major congestion problems. With the use of $ErouVe$ bottlenecks can be bypassed at a very short time and vehicles can be rerouted to more efficient paths in terms of CO2 and time.

Method	*CO2 (ml)*	*CO2 / Distance (ml/m)*	*Velocity*
	scenario 1 : 45 vehicles		
$Shortest-Path$	53.1	0.366	4.03
$ErouVe$	36.4	0.185	10.07
	scenario 2 : 90 vehicles		
$Shortest-Path$	49.6	0.342	4.48
$ErouVe$	39.2	0.228	8.79

Table 7.3: Mean performance values for the area of interest

As seen in Table 7.4 when comparing $Shortest-Path$ and $ErouVe$ with the default system parameters, $ErouVe$ achieves a decrement in CO2 emissions (ml/m) between 4% and 6%, while the additional distance that each vehicle has to travel is at most 45 meters(2.20 % longer that the shortest one).

Comparison of Tables 7.3 and 7.4 shows that the improvements, when measured on

Method	*CO2 (ml)*	*CO2 / D (ml/m)*	*Distance(m)*
	scenario 1 : 45 vehicles		
$Shortest-Path$	450	0.221	2043
$ErouVe$	436	0.208	2088
$\%Difference$	3.11	5.86	2.20
	scenario 2 : 90 vehicles		
$Shortest-Path$	481	0.235	2047
$ErouVe$	467	0.225	2076
$\%Difference$	2.90	4.21	1.41

Table 7.4: Average performance results of the complete routes

the entire vehicle paths are significant but not so impressive as when the area of interest is investigated in isolation. This happens for two reasons. First the alternative paths that constitute the area of interest are only 190 and 275 meters long, which are rather small segments compared to the whole trip that each vehicle has to travel ($\sim$ 2km). Another reason may be the fact that rerouting of vehicles happens in an individual way leading to major lane changing of vehicles when approaching first junction. A cooperative lane changing system or a local mechanism for rerouting combining V2V and I2V communications would eliminate this effect leading to further improvement of the system's performance.

Traffic management

Splitting traffic efficiently has a major impact on car emissions and on the time that a vehicle needs in order to reach its target. In Figure 7.7 we present how $ErouVe$ splits vehicle's flow in the two available paths for different number of simulated vehicles.

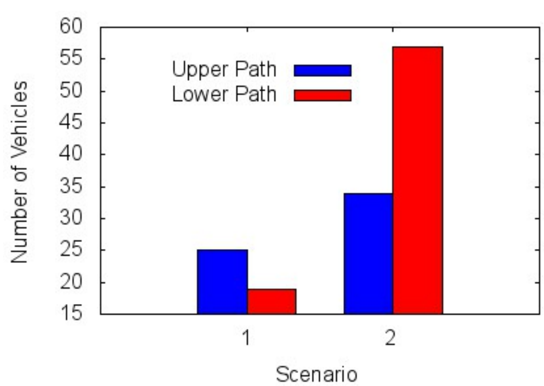


Figure 7.7: $ErouVe$ manages traffic efficiently

In Figure 7.8 CO2 emissions of vehicles, according to the path they follow and the method they use for routing, is represented. When all vehicles follow the $Shortest - Path$ the deviation of values is larger. Using the proposed routing mechanism $ErouVe$, vehicles that take the longer path emit more CO2 but this increase is limited due to the fact that the method ensures that both paths are not over occupied.

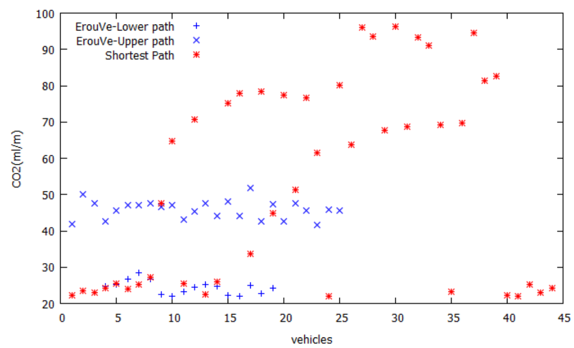


Figure 7.8: $ErouVe$ manages CO2 efficiently

7.7 Chapter conclusions

We presented a novel pull-based congestion avoidance routing method based on DSRC communications. The method exploits data transmitted from vehicles, in order to guide them in selecting the greener road segment to follow at each intersection. Based on the communication between RSUs and vehicles, the local routing problem is expanded from a single road segment to the whole area of investigation that covers all the surrounding road segments, helping reducing CO2 emissions and total travel time, while keeping additional traveled distance low.

Chapter 8
Conclusions and Future work

8.1 Summary of the Contributions

This thesis addresses the issues of data dissemination in ad hoc networks and especially in VANETs. Push-based broadcast indexes, clustering techniques, centrality metrics and delay efficient routing protocols are proposed and evaluated against state of the art methods. Issues such as delay efficiency, long cluster lifetime, proper ranking and skewed data indexing have been addressed for different network topologies. Most of the proposed methods are distributed or can be easily transformed to distributed, in order to apply to mobile add hoc networks.

8.1.1 Centrality metrics

Network protocols for mobile ad hoc wireless networks are based on localized algorithms, which means that they are allowed for performance and scalability purposes to use only local information, e.g., two or three hop connectivity information. Such localized centrality metrics present a potential for control of communication, safety issues, routing protocols, information dissemination and so on. We presented centrality metrics that characterize the network topology using only limited, local connectivity information one or two hop information. Nodes using beacons, that broadcast periodically, identify their neighborhood and compute their centrality based on the number of neighbors a node belongs to and the ranking this node has in each one. The methods are distributed and can be applied to any ad hoc network, static or mobile.

8.1.2 Node Clustering

The alleviation of the broadcast storm problem in VANETs is alleviated with the use of different schemes e.g. Probability, Counter, Distance, Location-based, Cluster-Based Schemes, Push-Based data dissemination mechanism etc. Focusing on the cluster based scheme we proposed various clustering mechanisms that take in mind different motion, structural and social characteristics of the nodes in order to elect clusterheads. The methods are developed for both highways and urban environments. The validation of the methods through simulations for different network characteristics revealed that the usage of complex metrics for the election of clusterheads give significant benefits in terms of cluster stability and lifetime.

8.1.3 Social vehicle Clustering

Vehicle clustering is a crucial network management task for vehicular networks in order to address the broadcast storm problem, and also to cope with the rapidly changing network topology. Developing algorithms that create *stable clusters* is a very challenging procedure because of the highly dynamic moving patterns of vehicles and of the dense topology. Previous approaches on vehicle clustering are based on either topology-agnostic features, such as vehicle IDs, or are based on hard to set parameters, or finally exploit very limited knowledge of the vehicles' trajectories.

We developed a pair of algorithms, namely *Sociological Pattern Clustering (SPC)*, and the *Route Stability Clustering (RSC)* – the latter being a specialization of the former – that exploit, for the first time in the relevant literature, the "social behavior" of vehicles, i.e., their tendency to share the same/similar routes. Both methods exploit the historic trajectories of vehicles gathered by road-side units located in the area of interest, and use the recently introduced clustering primitive of *virtual forces* as defined in [83]. The mobility, i.e., mobile patters of each vehicle is modeled as a semi-Markov process.

In order to assess the performance of the proposed clustering algorithms, we performed a detailed experimentation by simulation comparing its behavior to that of two high-performance state-of-the-art algorithms, which are representatives of two families of vehicle clustering algorithms, namely the *Low-Id* [40] and *DDVC* [86] protocols. The comparison involved the investigation of the impact of a range of parameters on the performance of the protocols, such as vehicle speed and transmission range, existence and strength of social patterns for both urban and highway-like environments. All the received results attested the superiority of the proposed algorithms for creating stable and "meaningful" clusters.

We have also presented a short term prediction mechanism that is based on two prediction mechanisms used simultaneously and complementary to each other in order to predict the next road segment that a vehicle will move. Tailored for the resource-rich VANET environments, the proposed *RDM* Markov predictor with its rich dictionary and its effective prediction mechanisms achieves highly accurate on-line trajectory predictions.

8.1.4 Delay efficient routing

Delay efficiency is a critical parameter for routing protocols. Throughput optimality ensures that packets are delivered to their final destination, while small delay is directly related to the achievement of a prescribed Quality of Service. When critical messages have to be delivered to the destination end to end delay plays an important role. Based on a well established throughput optimal routing method we developed several delay efficient protocols for static and mobile ad-hoc networks. The experimental evaluation of the proposed delay reduction methods attested their superiority in terms of QoS, robustness, low computational complexity and simplicity.

8.1.5 Broadcast Index

Push-based data dissemination mechanisms use fixed RSU or moving vehicles to periodically deliver data messages to other vehicles. These messages are managed by data centers which collect data from applications to deliver it to the vehicles. A computer with a wireless interface or an info-station can play the role of data center. This type of mechanism is useful for applications which need to advertise information to a set of vehicles. In most situations the majority of the vehicles that move in a specified area tend to retrieve some particular data instead of showing equal preference to all data; therefore the skewness in data access is very high. Performance of most push-based mechanisms degrade proportionally to the skewness of data. We developed a broadcast which exploits the different access probabilities of data items, in order to improve the mean tuning time, while retaining the same access time low.

8.1.6 Congestion avoidance

Traffic congestion is a problem modern cities have to face; it costs time, fuel and, thus, money. Extending the road network is not suited to fight congestion due to spatial, financial and environmental constraints. VANETs are an excellent tool to reduce congestion in the context of ITS. We proposed a new congestion avoidance scheme based on IVC communications. Vehicles give information about the roads they traverse to RSUs and the systems assign weights to the road segments according to this data. The validation of the proposed routing scheme through simulations revealed that the usage of alternative paths other than the shortest path (in terms of hops) give significant performance benefits and reduce significantly the mean time cost for the individual driver and total environmental cost for the city.

8.2 Future Work

The core work presented in this thesis can be extended in many ways. In distinguishing central nodes in a ad hoc network the proposed methods can be extended to take in mind several aspects of actors. Social characteristics of nodes can influence the significance a node may have in a region. Nodes that destined to areas of great importance, when trying to disseminate critical information like an accident, a traffic jam must also be favored to when centrality metrics are computed.

The proposed centrality metrics can be used as primitives in the design of networking protocols, like cooperative caching for ad hoc wireless networks, and routing in vehicular networks where other attributes like energy of nodes or link quality could be incorporated in the centrality metric in order to better represent significance of actors in real time.

In the area of clustering of nodes in a vehicular network we believe that crucial questions remain and should be considered as future work. Though the proposed methods of this thesis incorporate many different parameters of vehicles, such as direction, future position, velocity, height and even social profiles of the drivers, application driven methods should be developed in the near future in order to cope with different situations. Road safety and traffic congestion avoidance for example are two different circumstances that may arise in a VANET, with different requirements and limitations in terms of delay, dissemination coverage etc. Clustering parameters must be tuned for every different application that runs over the VANET and this must be done in a distributed and automated way. Beacons that vehicles exchange in order to define their neighborhood and elect clusterheads lead to awareness degradation due to interference in high load situations. Selective beaconing or longer intervals may cope with this problem. Short term prediction mechanisms finally, can be used along with the the sociological patterns of vehicles in order to increase prediction accuracy and cluster stability.

In this thesis we proposed two delay efficient routing methods. For the VoBp approach, we require the packets to carry their immediate travel history, so that the relays are prevented from sending the packets back from where they came. For the LayBP method, taking a less localized approach, we create layers of nodes and use the identities of these layers to forward the packets toward the destinations layer, and discouraging the packet from leaving the destination layer. This method is also developed and evaluated for low mobile networks. One interesting extension of the proposed work would be the development of a distributed cluster based delay efficient routing algorithm for VANETs that is based on the Backpressure methodology. The proposed method can route packets according to the clustering of nodes [83] or road segments [73].

Data dissemination from RSUs was also studied in this thesis. RSUs can play an important role in a smart city, covering certain areas of interest and giving accurate information fast. In this thesis we presented an application that uses information collected by RSUs in order to cope with road congestion situations. In the future, information from RSUs that are n hops away from the current junction can be exploited, in order to give longer term routing instructions to vehicles. Optimal values for the various parameters of the system is another open issue. Finally routing information itself can be embedded in the optimization problem of each vehicle giving the method a glance in the near future.

In Intelligent transportation systems, different applications for traffic avoidance have been developed the past years. Our proposed method exploits IVC communications in order to eco-route vehicles. Other methods that are based on TMCs (Traffic Message Channels), smart phone applications and GPSs try to cope with the same problem. The combination of LTE and DSRC communication capabilities of vehicles and their drivers can enhance the performance of these methods. Road users others than cars, e.g. pedestrians, cyclists etc can play an active role in data dissemination with the use of LTE. Protocols for these combined communications must be developed in order to gather local and global information from different sources and have more reliable outcomes.

Bibliography

[1] H. Hartenstein and K. P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications magazine*, 46(6):164–171, 2008.

[2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials*, 13(4):584–616, 2011.

[3] M. L. Sichitiu and M. Kihl. Inter-vehicle communication systems: A survey. *IEEE Communications Surveys & Tutorials*, 10(2):88–105, 2008.

[4] Y. Toor, P. Mühlethaler, A. Laouiti, and A. De La Fortelle. Vehicle ad hoc networks: Applications and and related technical issues. *IEEE Communications Surveys & Tutorials*, 10(3):74–88, 2008.

[5] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk. A survey of inter-vehicle communication protocols and their applications. *IEEE Communications Surveys & Tutorials*, 11(2):3–20, 2009.

[6] J. B. Kenney. Dedicated Short-Range Communications (DSRC) standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[7] Y. L. Morgan. Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics. *IEEE Communications Surveys & Tutorials*, 12(4):504–518, 2010.

[8] Rola Naja. Wireless vehicular networks for car collision avoidance. In *Springer 2013*, 2013.

[9] Rola Naja. *Wireless vehicular networks for car collision avoidance.* Springer, 2013.

[10] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[11] S. Abdallah. Generalizing unweighted network measures to capture the focus in interactions. *Social Network Analysis and Mining*, 2011. DOI: 10.1007/s13278-011-0018-8.

[12] W. Hwang, T. Kim, M. Ramanathan, and A. Zhang. Bridging centrality: Graph mining from element level to group level. In *Proceedings of ACM SIGKDD*, pages 336–344, 2008.

[13] D. Katsaros, N. Dimokas, and L. Tassiulas. Social network analysis concepts in the design of wireless ad hoc networks protocols. *IEEE Network magazine*, 24(6):23–29, 2010.

[14] M. Saravanan, G. Prasad, S. Karishma, and D. Suganthi. Analyzing and labeling telecom communities using structural properties. *Social Network Analysis and Mining*, 2011. DOI: 10.1007/s13278-011-0020-1.

[15] F. Gilbert, P. Simonetto, F. Zaidi, F. Jourdan, and R. Bourqui. Communities and hierarchical structures in dynamic social networks: Analysis and visualization. *Social Network Analysis and Mining*, 1(1):83–95, 2011.

[16] P.F. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.

[17] A. Bavelas. A mathematical model for group structure. *Human Organizations,*, 7:16–30, 1948.

[18] S. Brin, L. Page, R. Motwani, and T. Winograd. PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Computer Science Department, Stanford University, 1999.

[19] L.C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the ACM KDD Conf.*, pages 420–429, 2007.

[21] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social-based forwarding in Delay Tolerant Networks. In *Proceedings of the ACM MOBIHOC Conf.*, pages 241–250, 2007.

[22] N. Dimokas, D. Katsaros, and Y. Manolopoulos. Cooperative caching in wireless multimedia sensor networks. *ACM Mobile Networks and Applications*, 13(3-4):337–356, 2008.

[23] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos. High performance, low overhead cooperative caching in wireless sensor networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 233–243, 2009.

[24] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, 2000.

[25] Wei Gao, Guohong Cao, T. La Porta, and Jiawei Han. On exploiting transient social contact patterns for data forwarding in delay-tolerant networks. *Mobile Computing, IEEE Transactions on*, 2013.

[26] N. Dimokas, D. Katsaros, L. Tassiulas, and Y. Manolopoulos. High performance, low complexity cooperative caching for wireless sensor networks. *ACM Wireless Networks*, 17(3):717–737, 2011.

[27] M. Zhang and R. S. Wolff. Routing protocols for vehicular ad hoc networks in rural areas. *IEEE Communications magazine*, 46(11):126–131, 2008.

[28] N. Sastry and P. Pan Hui. Path formation in human contact networks. In *Handbook of Optimization in Complex Networks*. Springer, 2011. To appear.

[29] A. Korn, A. Schubert, and A. Telcs. Lobby index in networks, 2008. Available: http://arxiv.org/abs/0809.0514.

[30] S. Nanda and D. Kotz. Localized bridging centrality for distributed network analysis. In *Proceedings of IEEE ICCCN*, 2008.

[31] L. C. Freeman, S. P. Borgatti, and D. R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13:141–154, 1991.

[32] M. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27:39–54, 2005.

[33] N. Perra and S. Fortunato. Spectral centrality measures in complex networks. *Physical Review E*, 78(036107):036107–1–03107–10, 2008.

[34] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[35] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

[36] P. H. Pathak and R. Dutta. Using centrality-based power control for hot-spot mitigation in wireless networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[37] L. Maglaras and D. Katsaros. Distributed skip air index for smart broadcasting in intelligent transportation systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 624–629, 2012.

[38] M. Torrent-Moreno, M. Killat, and H. Hartenstein. The challenges of robust inter-vehicle communications. In *Proceedings of the IEEE Vehicular Technology Conference – Fall (VTC)*, volume 1, pages 319–323, 2005.

[39] P. Basu, N. Khan, and T.D.C. Little. A mobility based metric for clustering in mobile ad hoc networks. In *International Workshop on Wireless Networks and Mobile Computing (WNMC)*, pages 413–418, 2001.

[40] M. Gerla and J.T.-C Tsai. Multicluster, mobile, multimedia radio network. *ACM Wireless Networks*, 1, 1995.

[41] E. Sakhaee and A. Jamalipour. Stable clustering and communications in pseudo-linear highly mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(6):3769–3777, 2008.

[42] C. Shea, B. Hassanabadi, and S. Valaee. Mobility-based clustering in VANETs using affinity propagation. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2009.

[43] Y. Zhang and J.M. Ng. A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 3161–3165, 2008.

[44] S. Kuklinski and G. Wolny. Density based clustering algorithm for vehicular ad hoc networks. *International Journal of Internet Protocols and Technology*, 4(3):149–157, 2009.

[45] J. Blum, A. Eskandarian, and L. Hoffman. Mobility management in IVC networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 150–155, 2003.

[46] X. Zhang, H. Su, and H.-H Chen. Cluster-based multi-channel communications protocols in vehicle ad hoc networks. *IEEE Wireless Communications magazine*, 13(5):44–51, 2006.

[47] N. Dimokas, D. Katsaros, and Y. Manolopoulos. Energy-efficient distributed clustering in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 70(4):371–383, 2010.

[48] W. Chen and S. Cai. Ad hoc peer-to-peer network architecture for vehicle safety communications. *IEEE Communications magazine*, 43(4):100–107, 2005.

[49] I. Sen and D. W. Matolak. Vehicle channel models for the 5-GHz band. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):235–245, 2008.

[50] A. Paier, J. Karedal, N. Czink, C. Dumard, T. Zemen, F. Tufvesson, A. F. Molisch, and C. F. Mecklenbräuker. Characterization of vehicle-to-vehicle radio channels from measurements at 5.2GHz. *Wireless Personal Communications*, 50(1):19–32, 2009.

[51] J. S. Otto, F. E. Bustamante, and R. A. Berry. Down the block and around the corner: The impact of radio propagation on inter-vehicle wireless communication. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 605–614, 2009.

[52] M. Boban, T. T. V. Vinhoza, M. Ferreira, J. Barros, and O. K. Tonguz. Impact of vehicles as obstacles in vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(1):15–28, 2011.

[53] S. Kaul, K. Ramachandran, P. Shankar, S. Oh, M. Gruteser, I. Seskar, and T. Nadeem. Effect of antenna placement and diversity on vehicular network communications. In *Proceedings of the IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 112–121, 2007.

[54] J. Maurer, T. Fugen, T. Schafer, and W. Wiesbeck. A new inter-vehicle communications (IVC) channel model. In *Proceedings of the IEEE Vehicular Technology Conference – Fall*, volume 1, pages 9–13, 2004.

[55] J.and Boeglen H. Ledy, Poussard A. M., Hilt B., and Vauzelle R. A semi-deterministic channel model for VANETs simulations. *International Journal of Vehicular Technology*, 2011.

[56] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner. Propagation by diffraction. *International Telecommunication Union Radiocommunication Sector, Geneva, Recommendation P.526*, 2007.

[57] J. Epstein and Peterson D. W. An experimental study of wave propagation at 850MC. *Proceedings of the IRE*, 41(5):595–611, 1953.

[58] . Navteqs navstreets street data reference manual. October 2009.

[59] A. J. Dean and S. N. Brennan. Terrain-based road vehicle localization on multi-lane highways. *In Proceedings of the American Control Conference, Piscataway, NJ, USA, 2009. IEEE Press*, pages 6707–712, 2009.

[60] C.D. McGillem and T.S. Rappaport. A beacon navigation method for autonomous vehicles. *IEEE Transactions on Vehicular Technology*, 1989.

[61] M. Bertozzi and A. Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, pages 62–81, 1998.

[62] M. Jabbour, P. Bonnifait, and V. Cherfaoui. Enhanced local maps in a gis for a precise localisation in urban areas. *In Proceedings of the 9th IEEE Conference on Intelligent Transportation Systems (ITSC 2006)*, 2006.

[63] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner. Sumo (simulation of urban mobility); an open-source traffic simulation. *4th Middle East Symposium on Simulation and Modelling (MESM)*, pages 183–187, 2002.

[64] L.A. Maglaras and D. Katsaros. Enhanced spring clustering in vanets with obstruction considerations. In *77th IEEE Vehicular Technology Conference - Spring (VTC-Spring), Dresden, Germany*, 2013.

[65] L. Maglaras and D. Katsaros. Distributed clustering in vehicular networks. In *Proceedings of the International Workshop on Vehicular Communications and Networking (VECON)*, 2012.

[66] M. Kyriakakos, N. Frangiadakis, L. Merakos, and S. Hadjiefthymiades. Enhanced path prediction for network resource management in wireless LANs. *IEEE Wireless Communications magazine*, 10(6):62–69, 2003.

[67] T. Liu, P. Bahl, and I. Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks. *IEEE Journal on Selected Areas in Communications*, 16(6):922–936, 1998.

[68] D. Katsaros and Y. Manolopoulos. Prediction in wireless networks by Markov chains. *IEEE Wireless Communications magazine*, 16(2):56–64, 2009.

[69] O. V. Drugan, T. Plagemann, and E. Munthe-Kaas. Detecting communities in sparse MANETs. *IEEE/ACM Transactions on Networking*, 19(5):1434–1447, 2011.

[70] E. Bulut and B. K. Szymanski. Exploiting friendship relations for efficient routing in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(12):2254–2265, 2012.

[71] G. Pallis, D. Katsaros, M. D. Dikaiakos, N. Loulloudes, and L. Tassiulas. On the structure and evolution of vehicular networks. In *Proceedings of the IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 502–511, 2009.

[72] D. Naboulsi and M. Fiore. On the instantaneous topology of a large-scale urban vehicular network: The Cologne case. In *Proceedings of the ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2013.

[73] Q. Song and X. Wang. Efficient routing on large road networks using hierarchical communities. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):132–140, 2011.

[74] D. Katsaros, N. Dimokas, and L. Tassiulas. Social network analysis concepts in the design of wireless ad hoc network protocols. *IEEE Network magazine*, 24(6):23–29, 2010.

[75] J. Y. Yu and P. H. J. Chong. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 7(1):32–48, 2005.

[76] O. Younis, M. Krunz, and S. Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20(3):20–25, 2006.

[77] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *ACM/Kluwer WINET*, 8(2–3):153–167, 2002.

[78] L. A. Maglaras and D. Katsaros. Enhanced spring clustering in VANETs with obstruction considerations. In *Proceedin of the IEEE Vehicular Technology Conference - Spring (VTC-Spring)*, 2013.

[79] M. Gerlach. VaNeSe – An approach to VANET security. In *Proceedings of the International Workshop on Vehicle-to-Vehicle Communications (V2VCOM)*, 2005.

[80] J. M. De Fuentes, A. I. Gonzalez-Tablas, and A. Ribagorda. Overview of security issues in vehicular ad-hoc networks. In *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*, 2011.

[81] R. Simmons, B. Browning, Z. Yilu, and V. Sadekar. Learning to predict driver route and destination intent. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2006.

[82] L. A. Maglaras and D. Katsaros. Distributed clustering in vehicular networks. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 593–599, 2012.

[83] L.A. Maglaras and D. Katsaros. Clustering in urban environments: Virtual forces applied to vehicles. In *Proceedings of the IEEE Workshop on Emerging Vehicular Networks: V2V/V2I and Railroad Communications*, 2013.

[84] Andre-Luc Beylot and Houda Labiod. *Vehicular Networks: Models and Algorithms.* John Wiley & Sons, 2013.

[85] Fan Bai, Daniel D. Stancil, and Hariharan Krishnan. Toward understanding characteristics of dedicated short range communications (dsrc) from a perspective of vehicular network engineers. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 329–340, 2010.

[86] E. Sakhaeee and A. Jamalipour. Stable clustering and communications in pseudolinear highly mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(6):3769–3777, 2008.

[87] Minming Ni, Zhangdui Zhong, and Dongmei Zhao. Mpbc: A mobility prediction-based clustering scheme for ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 60(9):4549–4559, 2011.

[88] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536, 1978.

[89] Karthik Gopalratnam and Diane J Cook. Online sequential prediction via incremental parsing: The active lezi algorithm. *Intelligent Systems, IEEE*, 22(1):52–58, 2007.

[90] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE TAC*, 37(12):1936–1948, 1992.

[91] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *Proc. INFOCOM Mini*, 2009.

[92] L. Ying, R. Srikant, and D. F. Towsley. Cluster-based back-pressure routing algorithm. In *Proc. INFOCOM*, pages 484–492, 2008.

[93] L. Ying, S. Shakkotai, and A. Reddy. On combining shortest-path and back-pressure routing over multihop wireless networks. In *Proc. INFOCOM*, 2009.

[94] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari. LIFO-Backpressure achieves near optimal utility delay tradeoff, 2010. Available at: http://arxiv.org/abs/1005.4984.

[95] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali. Routing without routes: The backpressure collection protocol. In *Proc. IPSN*, 2010.

[96] E. Athanasopoulou, L. Bui, T. Ji, R. Srikant, and A. Stoylar. Backpressure-based packet-by-packet adaptive routing in communication networks, 2010. Available at: http://arxiv.org/abs/1005.4984.

[97] L. Maglaras and D. Katsaros. Layered backpressure scheduling for delay reduction in ad hoc networks. In *Proc. WoWMoM*, 2011.

[98] A. D. Amis and R. Prakash. Clusterhead selection in wireless ad hoc networks. US Patent 6,829,222 B2, 2004.

[99] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE TPDS*, 17(4):292–306, 2006.

[100] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE TSMC: Part B*, 26(1):29–41, 1996.

[101] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.

[102] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. MOBICOM*, 2000.

[103] A. Manjeshwar and D.P. Agrawal. TEEN: A protocol for enhanced efficiency in wireless sensor networks. In *Proc. IPDPS*, 2001.

[104] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. MOBICOM*, 2001.

[105] M. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proc. INFOCOM*, pages 1723–1734, 2005.

[106] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[107] G. R. Gupta and N. Shroff. Delay analysis for multi-hop wireless networks. In *Proc. INFOCOM*, 2009.

[108] Longbo Huang and Michael J. Neely. Delay efficient scheduling via redundant constraints in multihop networks. *Perform. Eval.*, 68(8), August 2011.

[109] Longbo Huang and M.J. Neely. Delay reduction via lagrange multipliers in stochastic network optimization. *Automatic Control, IEEE Transactions on*, 2011.

[110] L. Georgiadis, M. Neely, and L Tassiulas. Resource allocation and cross-layer control in wireless networks, 2006.

[111] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE JSAC*, 23(1):89–103, 2005.

[112] N. H. Vaidya and S. Hammed. Scheduling data broadcast in asymmetric communication environments. *ACM Wireless Networks*, 5(3):171–182, 1999.

[113] D. Aksoy and M. J. Franklin. $R \times W$: A scheduling approach for large-scale on-demand data broadcast. *IEEE/ACM Transactions on Networking*, 7(6):846–860, 1999.

[114] Y. Zhang, J. Zhao, and G. Cao. Service scheduling of vehicle-roadside data access. *ACM Mobile Networks and Applications*, 15(1):83–96, 2010.

[115] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and Y.-Y. Hsu. A survey and challenges in routing and data dissemination in vehicular ad hoc networks. wireless communications and mobile computing. In *Wireless Communications and Mobile Computing*, pages 787–795, 2011.

[116] H. Alshaer and J. M. H. Elmirghani. Road safety based on efficient vehicular broadcast communications. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1155–1160, 2009.

[117] S. S. Chawathe. Using dead drops to improve data dissemination in very sparse equipped traffic. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 962–967, 2008.

[118] C. Liaskos, S. Petridou, G. I. Papadimitriou, P. Nicopolitidis, and A. S. Pomportsis. On the analytical performance optimization of wireless data broadcasting. *IEEE Transactions on Vehicular Technology*, 59(2):884–895, 2010.

[119] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, 1997.

[120] W.-C. Lee and D. L. Lee. Using signature techniques for information filtering in wireless and mobile environments. *Distributed and Parallel Databases*, 4(3):205–227, 1996.

[121] D. Katsaros, N. Dimokas, and Y. Manolopoulos. Generalized indexing for energy-efficient access to partially ordered broadcast data in wireless networks. In *Proceedings of the IEEE International Database Engineering and Applications Symposium (IDEAS)*, pages 89–96, 2006.

[122] N. Shivakumar and S. Venkatasubramanian. Efficient indexing for broadcast based wireless systems. *ACM/Baltzer Mobile Networks and Applications*, 1(4):433–446, 1996.

[123] C.-H. Hsu, G. Lee, and A. L. P. Chen. Index and data allocation on multiple broadcast channels considering data access frequencies. In *Proceedings of the Conference on Mobile Data Management (MDM)*, pages 87–93, 2002.

[124] W. G. Yee and S. B. Navathe. Efficient data access to multi-channel broadcast programs. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pages 153–160, 2003.

[125] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Power efficient filtering of data on air. In *Proceedings of the Conference on Extending Data Base Technology (EDBT)*, pages 245–258, 1994.

[126] J. Xu, W.-C. Lee, and X. Tang. Exponential index: A parameterized distributed indexing scheme for data on air. In *Proceedings of the 2nd ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004.

[127] J. Xu, W.-C. Lee, X. Tang, Q. Gao, and S. Li. An error-resilient and tunable distributed indexing scheme for wireless data broadcast. *IEEE Transactions on Knowledge and Data Engineering*, 18(3):392–404, 2006.

[128] A. Seifert and J.-J. Hung. FlexInd: A flexible and parameterizable air-indexing scheme for data broadcast systems. In *Proceedings of the Conference on Extending Data Base Technology (EDBT)*, volume 3820 of *Lecture Notes in Computer Science*, pages 902–920, 2006.

[129] X. Yang and A. Bouguettaya. Adaptive data access in broadcast-based wireless environments. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):326–338, 2005.

[130] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 126–134, 1999.

[131] T. Hu and A. Tucker. Optimal computer search trees and variable-length alphabetical codes. *SIAM Journal on Applied Mathematics*, 21(4):514–532, 1971.

[132] J. Xu and K.-L. Tan. An analysis of selective tuning schemes for nonuniform broadcast. *Data and Knowledge Engineering*, 22(3):319–344, 1997.

[133] S. Jung, B. Lee, and S. Pramanik. A tree-structured index allocation mathod with replication over multiple broadcast channels in wireless environments. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):311–325, 2005.

[134] S.-C. Lo and A. L. P. Chen. Optimal index and data allocation in multiple broadcast channels. In *Proceedings of the IEEE Conference on Data Engineering (ICDE)*, pages 293–302, 2000.

[135] J. Zhong, W. Wu, Y. Shi, and X. Gao. Energy-efficient tree-based indexing schemes for information retrieval in wireless data broadcast. In *Proceedings of the International Conference Database Systems for Advanced Applications (DASFAA)*, pages 335–351, 2011.

[136] M.-S. Chen, K.-L. Wu, and P.S. Yu. Optimizing index allocation for sequential data broadcasting in wireless mobile computing. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):161–173, 2003.

[137] B. Zheng, W.-C. Lee, K. C. K. Lee, D. L. Lee, and M. Shao. A distributed spatial index for error-prone wireless data broadcast. *The VLDB Journal*, 18(4):959–986, 2009.

[138] T. Sakata and J. X. Yu. Statistical estimation of access frequencies: Problems, solutions and consistencies. *ACM Wireless Networks*, 9(6):647–657, 2003.

[139] P. V. Argade. a high-performance low-power microprocessor. In *Proceedings of COMPCON*, pages 88–95, 1993.

[140] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito, and T. Higashino. A method for sharing traffic jam information using inter-vehicle communication. In *Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on*, 2006.

[141] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, 2004.

[142] Xue Yang, Jie Liu, N.F. Vaidya, and Feng Zhao. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, 2004.

[143] 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*, VANET '04, 2004.

[144] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011.

[145] Qing Song and Xiaofan Wang. Efficient routing on large road networks using hierarchical communities. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):132–140, 2011.

[146] D. de Bruin, J. Kroon, R. van Klaveren, and M. Nelisse. Design and test of a cooperative adaptive cruise control system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 392–396, 2004.

[147] R.P.A. Naus, G.J.L.and Vugts, J. Ploeg, M.J.G. van de Molengraft, and M. Steinbuch. Towards on-the-road implementation of cooperative adaptive cruise control. In *Proceedings of the 16th World Congress and exhibition on Intelligent Transport Systems and Services (ITS-16)*, 2009.

[148] C. Desjardins and B. Chaib-draa. Cooperative adaptive cruise control: A reinforcement learning approach. *Intelligent Transportation Systems, IEEE Transactions on*, 2011.

[149] Myung Jin Woo, Jae Weon Choi, and Hyung Seok Han. Carrier phase gps/millimeter-wave radar for vehicle platooning. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, 2001.

[150] Michele Segata, Falko Dressler, Renato Lo Cigno, and Mario Gerla. A simulation tool for automated platooning in mixed highway scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2012.

[151] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak. Vehicular networks for reduction of fuel consumption and co2 emission. In *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010.

[152] Ciprian Dobre, Adriana Szekeres, Florin Pop, Valentin Cristea, and Fatos Xhafa. Intelligent traffic lights to reduce vehicle emissions. In *26th EUROPEAN CONFERENCE ON MODELLING AND SIMULATION*, 2012.

[153] F. Terroso-Saenz, M. Valdes-Vela, C. Sotomayor-Martinez, R. Toledo-Moreo, and A.F. Gomez-Skarmeta. A cooperative approach to traffic congestion detection with complex event processing and vanet. *Intelligent Transportation Systems, IEEE Transactions on*, 2012.

[154] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano. Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, 2010.

[155] Rui Zhang, Zhijun Li, Cheng Feng, and Shouxu Jiang. Traffic routing guidance algorithm based on backpressure with a trade-off between user satisfaction and traffic load. In *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012.

[156] A. Cappiello, I. Chabini, E. K. Nam, A. Lue, and M. A. Zeid. A statistical model of vehicle emissions and fuel consumption. *IEEE 5th International Conference on Intelligent Transportation Systems*, 10(1):801–809, 2002.