Reduction
2011-2014

Deliverable 1.3
VANET Packet Scheduling/Routing
and Information Dissemination
August 30, 2012

Contents

Project acronym: Reduction
Project full title: Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behaviour Adaptation

Work Package: 1

Document title: VANET Packet Scheduling/Routing and Information Dissemination
Version: 1.3
Official delivery date: August 31, 2012
Actual publication date: _____
Type of document: Report
Nature: Public

Authors: Dimitrios Katsaros and Leandros Maglaras

Approved by: _____

Version	Date	Sections Affected
1.0	August 17, 2012	Initial version
1.1	August 25, 2012	Review comments processed
1.2	August 26, 2012	Updated to reflect changes in the implementation
1.3	August 27, 2012	Updated to reflect changes in the implementation

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 [25, 28, 56, 62, 71], 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 [30, 44] (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).

The aim of REDUCTION's first work package (WP1) is to address the challenges involved with the hardware and the (communication and networking) protocols needed to realize an environment where information will travel from vehicles to other vehicles or to infrastructure elements, and vice versa. The role of the University of Thessaly within this WP is to develop all the necessary communication infrastructure and wireless communication/networking protocols. In accordance with DoW, the very first goal involves the modeling of the vehicular network as a Constraint Queueing Network (CQN) and subsequently the development of packet scheduling/routing policies with guaranteed (optimal) throughput and stability properties; these policies should also be delay-aware, since many application-level tasks of REDUCTION are time-critical. We did significant steps towards this goal, as described in subsection 4.5 of this report. A second goal which was pursued in parallel was the design of communication protocols for information dissemination based on vehicle clustering, aggregation, caching, and so on. Since this goal had many 'subtargets', we strove for the investigation of the fundamental one, which is the development of clustering techniques; later on, the rest of the aims can be followed, since aggregation can be directly done by clusterheads, and cooperative caching protocols can be implemented with the aid of clustering; for instance (the late) Denko [17] exhibited the feasibility of the idea.

1.1 Vehicular communications and networking aspects in REDUCTION

To enhance the safety of drivers, to provide a comfortable driving environment and to contribute to fuel economy, messages for different purposes need to be sent to vehicles through intervehicle communications. *Unicast* routing is a fundamental operation for vehicle to construct a source-to-destination routing in a VANET. *Multicast* is defined by delivering multicast packets from a single source vehicle to all multicast members by multi-hop communication. *Geocast* routing [41] is to deliver a geocast packet to a specific geographic region. Vehicles located in this specific geographic region should receive and forward the geocast packet; otherwise, the packet is dropped. *Broadcast* protocol is utilized for a source vehicle sends broadcast message to all other vehicles in the network. Apart from 1-hop broadcast operations, the other types of communications, require some lower-level packet scheduling algorithms and specific network organization. Among the most promising cross-layer scheduling methods is the *back-pressure* protocol [61]; on the other hand, in order to address the broadcast storm problem [66] in VANETs, some form of *vehicle clustering* is required.

As already described in *Deliverable 1.1* (February 29, 2012), in order to address the scalability problems [66] arising in any ad hoc network, and in particular in a VANET,

we chose a two tier architecture for REDUCTION, which necessitates the creation of *clusters*. More arguments in favor of clustering can be found at subsection 3.1. Thus, we had to survey and evaluate the existing MANET and VANET clustering protocols, and depending on the outcome of the evaluation to develop new clustering protocol(s). Indeed, in subsection 3.3 we evaluated three graph-theoretic clustering protocols [20] that were proposed for MANETs, but seemed promising for VANETs as well, due to their low message overhead (low congestion/delay) and their fast convergence (ideal for rapidly changing topologies). One of them was a newly proposed randomized algorithm that creates maximum independent sets with optimal message complexity; randomization seems a promising technique for clustering and has recently received attention [19]. Our study showed that there is no single clear winner for all situations; we extended our efforts and developed a new clustering protocol in 3.4, namely the *SPRING* clustering protocol, that is proposed as a communication-efficient and low-delay solution. Our investigation is not yet complete, and we plan to extent the current performance analysis of the protocol [39] a) with more simulations and comparisons against the state-of-the-art VANET clustering protocol(s), and b) with implementation of it in a real small-scale VANET using the NITOS testbed of the University of Thessaly.

Information dissemination is a crucial issue for VANETs [47], but unfortunately there is no single solution to fit all needs. Despite the wealth of routing and information dissemination protocols for VANETs in the literature [34, 36, 60], none of them can offer at the same time throughput and delay optimality as needed for REDUCTION. We searched for solution to this problem in the family of the so-called cross-layer solutions, and in particular those protocols which are based on Lyapunov optimization [61]. Unfortunately, even though they are throughput-optimal, they suffer from long delays in packet delivery. Thus, we worked on a line of research that combines this packet scheduling/routing cross-layering technique, with clustering; our results [40] – cf. subsection 4.5 – extent our earlier work for static ad hoc networks [38]. Still, we work on this problem to convert it into a fully distributed protocol.

(Uni-Multi-Geo-Broad)cast routing can be implemented with the aid of four basic operations that were originally specified by the GEONET project [48]. These operations are the *geo-unicast*, *geo-broadcast*, *geo-anycast* and *topo-broadcast*. VANET clustering and packet scheduling are essential protocols for developing these operations, which in their turn are vital to REDUCTION, and they will be implemented over the hardware provided by DELPHI (the other partner in WP1), namely their DSRC communications box.

Geo-unicast.

The GeoUnicast forwarding algorithm is executed by a router to relay a packet to the next hop. The ETSI standard defines two GeoUnicast forwarding algorithms:

- Greedy Forwarding (GF) algorithm. With the Greedy Forwarding (GF) algorithm, the router uses the location information of the destination carried in the GeoUnicast packet header and selects one of the neighbors as the next hop. The algorithm applies the most forward within radius (MFR) policy, which selects the neighbor with the smallest geographical distance to the destination, thus providing the greatest progress when the GeoUnicast packet is forwarded.

- Contention-based forwarding (CBF) algorithm. With the Contention-based forwarding (CBF) algorithm, a receiver decides to be a forwarder of a GeoUnicast packet. This is in contrary to the sender-based forwarding scheme specified in GF, where the sender determines the next hop. The CBF algorithm utilizes

timer-based re-broadcasting with overhearing of duplicates in order to enable an implicit forwarding of a packet by the optimal node. With CBF, the router broadcasts the GeoUnicast packet. All neighbours, which receive the packet, process it: the router buffers the packet in its CBF packet buffer and starts a timer with a timeout that is inversely proportional to the distance between the router's local position and the destination's positions.

Geo-broadcast – Geographically-Scoped broadcast (GBC).
The GeoBroadcast forwarding algorithm is executed by a router to relay a packet to the next hop. The ETSI standard defines three forwarding algorithms; the two are experimental, so the following one is the dominant:

- Simple GeoBroadcast forwarding algorithm. The algorithm utilizes the function $F(x,y)$ specified in order to determine whether the router is located inside, at the border or outside the geographical target area carried in the GeoBroadcast packet header. If the router is inside or at the border of the area, the packet shall be re-broadcasted. If it is outside the area, the packet shall be forwarded by the GF algorithm.

$$F(x,y) = \begin{cases} = 1, & \text{for } x = 0 \text{ and } y = 0 \text{ (at the center point)}, \\ > 0, & \text{inside the geographical area}, \\ = 0, & \text{at the border of the geographical area}, \\ < 0, & \text{outside the geographical area}. \end{cases}$$

Geo-anycast – Geographically-Scoped Anycast (GAC).
It is the forwarding mechanism that transports data from a single node to any of the nodes within a geographically area. Compared to geographically-scoped broadcast, with geographically-scoped anycast a packet is not forwarded inside of the geographic area when the packet has reached the area. Therefore, the operations for GeoAnycast packet handling are similar to those for GeoBroadcast packet with minor changes to the source, forwarder and receiver node operation, to support this slightly different functionality.
Topo-broadcast – Topologically-scoped broadcast (TSB).
TSB offers re-broadcasting of a data packet from a source to all nodes that can be reached in certain number of hops (all nodes in an *n*-hop neighborhood). Single-hop broadcast is a special case of TSB that is used to send heartbeats including application data payload.

1.2 Relevance of VANET networking aspects to CO2 reduction

In the previous paragraphs we used a strictly 'technical networking language' to present the concepts and developments in the context of REDUCTION with respect to wireless communications and networking. The astute reader will have already understood the implication of those protocols to fuel economy and subsequently to CO2 reduction. For the sake of completeness though, we will provide concrete evidence of the positive impact of such kind of protocols to driver safety and to fuel economy with the aid of an example (see Figure 1) that involves an accident in a highway at a peak time.
Event takes place. We suppose that 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.
Traffic management improvement. The approaching vehicles send warning messages to the immediately following vehicles to reduce speed so as to avoid a new car crash. This is a typical, widespread safety application in today's automotive industry. If the vehicles are organized into clusters, the clusterhead of each group suppresses the redundant messages; instead of naive flooding we employ cluster-based routing. Therefore, we have fewer message collisions, we have shorter MAC access time, and thus faster delivery of the message. So far fuel reduction is not present. These messages are propagated further back to the other following vehicles that will quickly be notified about the blocked highway; thus they will have the possibility to follow an exit of the highway. Apparently, the faster the message spreads backwards, the more vehicles will get it before reaching the exit.
Congestion avoidance. By having vehicles following these exits, the highway is not severely congested.
Travel time reduction. Those vehicles that followed exits and 'escaped' from the accident site – instead of standing still and consuming fuel – are now able to go to their destination by alternative routes in shorter time.
Fuel and CO2 reduction. Shorter travel time (usually) translates into saving more fuel.

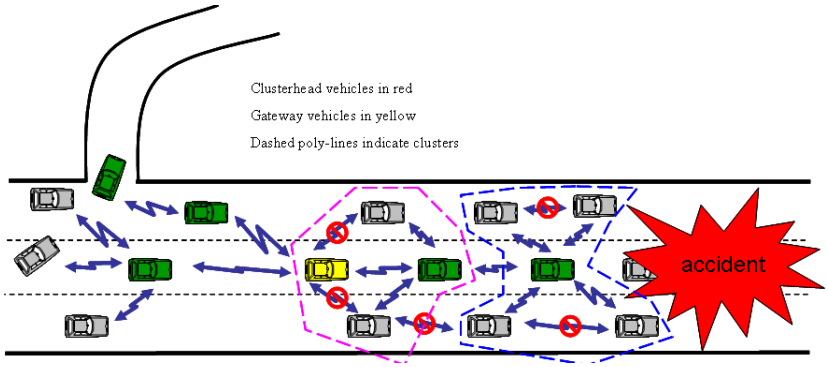


Figure 1: Illustration of an example where vehicle clustering and delay-aware packet scheduling can save fuel.

The potential of V2V communication for improving fuel efficiency has also been demonstrated in [70], showing that vehicular communications can assist to reduce average fuel consumption especially under high traffic density and long traffic light cycles. Other projects have investigated the impacts on fuel efficiency when using wireless communications between vehicles (V2V) or vehicle-to-infrastructure (V2I) employing different algorithms to smoothly slow down at a red traffic light or to reach it at the next green phase [42, 69].

2 Routing in VANETs

In VANET, the routing protocols – despite many detailed categorizations – can in general be classified into five categories: *topology-based* routing protocols, *position-based* routing protocols, *cluster-based* routing protocols, *geocast* routing protocol and *broadcast* routing protocol. These protocols are characterized on the basis of area/application

where they are most suitable, and in the next paragraphs we provide a brief survey of them.

2.1 Topology-based routing protocols

These routing protocols use links information that exists in the network to perform packet forwarding. They are further divided into Proactive, Reactive and Hybrid Protocols.

1. Proactive routing protocols

The proactive routing means that the routing information, like next forwarding hop is maintained in the background irrespective of communication requests. The advantage of proactive routing protocol is that there is no route discovery since the destination route is stored in the background, but the disadvantage of this protocol is that it provides low latency for real time application. The various types of proactive routing protocols are: FSR, DSDV, OLSR, CGSR, WRP, TBRPF.

2. Reactive/Ad hoc based routing

Reactive routing opens the route only when it is necessary for a node to communicate with each other. Reactive routing consists of route discovery phase in which the query packets are flooded into the network for the path search and this phase completes when route is found. The various types of reactive routing protocols are AODV, PGB, DSR,TORA, JARR.

3. Hybrid Protocols

The hybrid protocols are introduced to reduce the control overhead of proactive routing protocols and decrease the initial route discovery delay in reactive routing protocols

2.2 Position-based routing protocols

Position based routing consists of class of routing algorithm. They share the property of using geographic positioning information in order to select the next forwarding hops. Position based routing is broadly divided in two types: Position based greedy V2V protocols, Delay Tolerant Protocols.

1. Non DTN
2. DTN Position Based Routing Protocols
3. Hybrid Position Based Protocols

2.3 Broadcast routing

Broadcast routing is frequently used in VANET for sharing, traffic, weather and emergency, road conditions among vehicles and delivering advertisements and announcements. The various Broadcast routing protocols are BROADCOMM, UMB, V-TRADE, and DV-CAST.

2.4 Geocast routing protocols

Geocast routing is basically a location based multicast routing. Its objective is to deliver the packet from source node to all other nodes within a specified geographical region (Zone of Relevance ZOR). The various Geo cast routing protocols are IVG, DG-CASTOR and DRG

2.5 Cluster-based routing protocols

Cluster based routing is preferred in clusters. A group of nodes identifies themselves to be a part of cluster and a node is designated as cluster head will broadcast the packet to cluster. Good scalability can be provided for large networks but network delays and overhead are incurred when forming clusters in highly mobile VANET. The various Clusters based routing protocols are COIN, LORA-CBF, TIBCRPH, CBDRP.

1. CBDRP : Cluster-Based Directional Routing Protocol [57]

Cluster-based directional routing protocol (CBDRP) is focused on highway scenarios. Clusters are formed from vehicles that move to the same direction. In this method the header of a cluster selects another header according to the moving direction of vehicle to forward packets.

2. TIBCRPH: Traffic Infrastructure Based Cluster Routing Protocol with Handoff [68]

Frequent disconnected Network is a common feature of VANETS due to high mobility of vehicles. In order to ensure the service quality of nodes' communications, it makes use of the handoff idea of cellular networks and propose a new protocol which is special for VANET, the proposed scheme is called Traffic Infrastructure Based Cluster Routing Protocol with Handoff.

3. LORA-CBF: Location Routing Algorithm with Cluster-Based Flooding [52]

In LORA-CBF, there are three types of nodes: headers, gateways and members. The cluster-head maintains information about its members and gateways.Packet forwarding follows the same principles as the greedy routing. Clusterhead and gateways send out the location request (LREQ) packets when the location of the destination is not available as well as the phase of the Location Reply (LREP) messages. The proposed LORA-CBF shows highly heterogeneous performance results.

4. COIN: Clustering for Open IVC Network [12]

Cluster head selection in COIN is based on vehicular dynamics and driver intentions instead of ID or relative mobility as in conventional clustering methods. Relative mobility between a cluster head and a member node should be low, so they remain in radio contact for as long as possible.

5. HCB: Hierarchical Cluster-based routing [73]

It is a novel based Hierarchical Cluster routing protocol designed for highly mobility adhoc networks. They attempt to mitigate the impact of rapid topology change by exploiting the node heterogeneity and organizational structure of clusters HCB is two-layer communication architecture. In layer-1 mostly nodes communicate with each other via multi-hop routing. Among these nodes some also have another interface with long radio communication range called super nodes which exist on both layers. Super nodes are able to communicate with each other via the base station in layer-2.

6. CBLR: Cluster-based location routing [53]

This algorithm assumes all vehicles can gather their positions via GPS. The network is divided to multiple clusters each having a cluster head and a group of members within the transmission range. The cluster-head and members are formed as follow: A new vehicle transmits a Hello Message. If the vehicle gets a reply from the cluster-head vehicle, the new vehicle would become a member of the cluster. If not, the new vehicle becomes the clusterhead.

7. CBR: Cluster-based routing [37]

CBR (Cluster Based Routing) Protocol is a routing protocol based on position and clusters. The geographic area is divided into square grids. Each node calculates optimal neighbor cluster header to forward data to the next hop by using geographic in-

formation. This protocol does not consider velocity and direction which are important parameters in VANET.

3 Clustering in VANETs

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). 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. Since VANETs have a very limited capacity, it is desired that the number of messages can be reduced e.g., using aggregation.

The dynamic nature of wireless ad hoc networks though, requires that solutions for multihop network protocols must be distributed. Of the solutions proposed for scaling down networks with large numbers of nodes, *network clustering* is among the most investigated for mobile ad hoc networks [4, 5, 8, 9, 14, 24, 32, 58, 72, 65, 78], for sensor ad hoc networks [18, 77], and for vehicular ad hoc networks [39, 46, 67].

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.

In a clumpy approach, clustering algorithms can be divided in two major families. The first with its roots in *graph theory*, exploits the localized network structure for estimating dynamically the clusterheads (CHs), while the second provides mechanisms to ameliorate the fact that nodes belonging to the backbone are solely responsible for carrying out all communication, thus running out of energy very soon. Namely, the latter family addresses the energy consumption problem, that is essentially proposes ways to rotate the role of CH among nodes of clusters e.g., the SPAN [15], the LEACH [26] and the HEED [76] protocols. The proposed methods use the residual energy of each node in order to direct its decision about whether it will elect itself as a CH or not. However, this family's methods ignore topological features of the nodes.

The former family of protocols encompasses the most representative and successful solutions in the area of network management. This is because while it exploits the wealth of information extracted from the network topology particularities, it can also take into account application specific requirements, such as QoS. Moreover, algorithms of this family can easily be combined with a round robin CH rotation method as was described in [18], and thus exploit all the advantages the energy-efficient algorithms provide. Apparently, for the case of VANETs, energy consumption by vehicles is not an issue, since their battery can provide nearly unlimited power.

3.1 Why clustering?

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 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 multipath 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.

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 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.

To implement a robust clustering algorithm, two problems need to be solved: a clustering metric has to be selected and clustering methods need to be designed. A clustering metric should provide the quantitative definition of closeness which is fundamental to the clustering problem. Furthermore, clustering metric should reflect the characteristics of its target networks. In VANETs, the network topology is frequently changing. If a metric can resist against the unnecessary changes while maintaining connectivity, it is robust. A suitable metric can reflect features of nodes and improve performance of the clustering algorithm. Clustering methods are the approaches to assign nodes into clusters. Strictly speaking, in vehicular networks, the problem is not only how to form a cluster structure, but also how to maintain the cluster structure in a dynamic network.

3.2 Clustering techniques

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 [63] 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 MOBIC [10], which is an extension of the Lowest-ID algorithm [23]. 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 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 clustrer 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 DDVC clustering (DLDC)

[51] 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 [54]. In Distributed group mobility adaptive clustering (DGMA) [79] 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 [31]. Blum et al. [12] proposed a Clustering for Open IVC Networks (COIN) algorithm where cluster-head election is based on vehicular dynamics and driver intentions.

3.2.1 Clustering in VANETs

In cluster-based routing protocols vehicles near to each other form a cluster. Each cluster has one cluster-head, which is responsible for intra and inter-cluster management functions. Intra-cluster nodes communicate each other using direct links, whereas inter-cluster communication is performed via clusterheaders. In cluster based routing protocols the formation of clusters and the selection of the cluster-head is an important issue. In VANET due to high mobility dynamic cluster formation is a towering process. Many clustering techniques for VANETS have been developed lately. The cluster based methods are divided in five major categories. The methods that are focused in Urban environments, those that suitable for the VANET environment on highways , methods that combine V2V and V2I communication and the two-tier architectures. One other category of clustering methods may be those that were initially produced for Manets can be used in Vanets with some modifications.

Urban environments.

Affinity propagation is an algorithm for image processing, and APROVE has proved that its distributed case can be utilized for VANETs [54]. APROVE distributively elects clusterheads by using affinity propagation from a communications perspective. The method is tuned for highway scenarios.

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 [31].

In [2]The proposed algorithm is based on the assumption that each vehicle knows its exact lane on the road via a lane detection system and an in-depth digital street map that includes lane information.The clusterhead is selected based on the flow of the majority of traffic. A lane weight (LW) metric is applied for each traffic flow (LT, RT and NT). The clusterhead selected is towards the middle of the cluster and continuing in the same direction as the majority of the traffic flow.

Clustering in highways.

In [21] during the process of the clusterhead selection, the closest position to the average and the closest velocity to the average of all proximal vehicles are calculated along with connectivity level to determine the most stable clusterhead. The method performs well in highways but not in Urban envirnonments where traffic flow must be taking in mind.

Cluster-based directional routing protocol (CBDRP) [57] is focused on highway scenarios. Clusters are formed from vehicles that move to the same direction. In this method the header of a cluster selects another header according to the moving direction of vehicle to forward packets.

In LORA-CBF, there are three types of nodes: headers, gateways and members. The cluster-head maintains information about its members and gateways.Packet for-

warding follows the same principles as the greedy routing. Clusterhead and gateways send out the location request (LREQ) packets when the location of the destination is not available as well as the phase of the Location Reply (LREP) messages. The mobility of the vehicles on a motorway has been used to evaluate average route discovery time, end-to-end delay and other performance metrics.

The network in [53] is divided to multiple clusters each having a cluster head and a group of members within the transmission range. Packets are forwarded from source to destination by a procedure similar to AODV. The main difference is that only cluster heads and gateways forward packets.

Cluster head selection in COIN [12] is based on vehicular dynamics and driver intentions. Relative mobility between a cluster head and a member node should be low, so they remain in radio contact for as long as possible.

Two Tier architecture.

Hierarchical Cluster Based Routing [73] is designed for highly mobility adhoc networks. HCB is two-layer communication architecture. In layer-1 mostly nodes communicate with each other via multi-hop routing. Among these nodes some also have another interface with long radio communication range called super nodes which exist on both layers. Super nodes are able to communicate with each other via the base station in layer-2.

In [64] authors propose a two-tier architecture where vehicles are first organized into groups in VANETs.Traffic information is broadcasted and exchanged among vehicles through IVC. Some vehicles in the groups are selected to form a high-tier P2P overlay through infrastructure wireless communication. These vehicles are called superpeers and serve as a bridge between the high-tier and low-tier networks to handle message exchanges and lookups.

In [11], an integration of VANET and 3G networks using mobile gateways is introduced. A VANET-3G integrated network architecture is presented which defines the concept of mobile gateways. A mobile gateway refers to the dual-interfaced vehicle that relays data from other vehicle sources to the UMTS backhaul network.

V2I-V2V protocols.

In [59] authors propose an RSU-assisted cluster head selection and backup protocol in order to cope with situations where the communication is lost between two adjacent cluster heads.

Frequent disconnected Network is a common feature of VANETS due to high mobility of vehicles. In order to ensure the service quality of nodes' communications, [68] make use of the handoff idea of cellular networks and propose a new protocol which is special for VANET, the proposed scheme is called Traffic Infrastructure Based Cluster Routing Protocol with Handoff.

MANET clustering methods.

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 [63] 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. Many clustering methods initially created for Vanets cannot cope with these characteristics of the VANETS.

A well-known mobility-based clustering technique is MOBIC [10], which is an extension of the Lowest-ID algorithm [23]. 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 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 clustrer 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 DDVC clustering (DLDC) [51] the Doppler shift of communication signals is used in order to create clusters. In Distributed group mobility adaptive clustering (DGMA) [79] group mobility information which contains group physical center's coordinates, group size, group velocity is used for clustering.

[50] is a Cluster Based Vehicular Adhoc Network Model for Simple Highway Communication.Vehicle within a radio coverage range can communicate directly or though roadside unit. In this model a very few Fixed roadside units are assumed. All the cluster heads are synchronized in a specific time interval.

3.3 Evaluation of three low-overhead, high performance clustering schemes

Recently, *RanMIS* [1] considered the problem of computing a *maximal independent set (MIS)*, a fundamental distributed computing procedure, that seeks to elect a set of local leaders in a network. It generates the maximal set of nodes in such a manner that, no two of them are neighbors, by using an extremely harsh broadcast model that relies only on carrier sensing. Since the set is maximal, every node in the network is either in the *MIS* or a neighbor of a node in the *MIS*.

The model consists of an anonymous broadcast network in which nodes have no knowledge about the topology of the network. It has its roots to the solution of a similar problem that evolves during the development of the fly's nervous system, when sensory organ precursor (SOP) cells are chosen [1].

According to the specified assumptions, nodes receive as input, an upper bound on the number of nodes in the network n, and an upper bound D, on the number of neighbors any node can have (if no such bound is known, then D is set to n). Furthermore, it is assumed that they all wake up together at the same synchronous round (and start executing the algorithm), also that they can detect collisions, and finally that no failures occur.

The algorithm proceeds in $\log D$ phases, each consisting of $M \log n$ steps, where M is a constant. Initially, all nodes are active. Each step in each phase i consists of two exchanges.

In the first exchange, each active node broadcasts a message to its neighbors with probability p_i. Such as in the biological model, the probability p_i increases with i. In the second exchange, a node that has broadcasted a message in the first exchange joins the *MIS* if none of its neighbors had broadcasted at the first exchange. Such node broadcasts again a message to its neighbors, telling them to become inactive, and exits

the algorithm.

For the sake of completeness we give in Figure 2 the pseudocode of the *RanMIS* algorithm [1] which is synchronously executed by all nodes.

```
Algorithm: RanMIS (n, D) at node u
// n: number of participating nodes in network topology
// u: node under consideration
// D: upper bound on the number of neighbors
// B: 1 bit message
// M: constant

1 . For(i = 0 : log D)
2 .   For (j = 0 : M log n)
3 .       * exchange 1 *
4 .       v = 0;
5 .       With probability 1/2^(log D - i),
              broadcast B to neighbors;
              set v = 1;
6 .       If received message from neighbor, then
              v = 0;
7 .       * exchange 2 *
8 .       If v = 1 then
9.            Broadcast B;
              join MIS;
              exit the algorithm.
10.       Else
11.           If received message B in this exchange, then
                  mark node u inactive;
                  exit the algorithm.
12.       End.
13.   End.
14. End.
```

Figure 2: The Randomized MIS clustering algorithm.

RanMIS competitors were selected from two distinct protocol categories. The first category is oriented on providing the network with a two-layer hierarchical organization comprised by groups of nodes, i.e., clusters. One 'special' node of each cluster (the CH) participates in the so-called backbone; the backbode nodes form a *dominating set (DS)* over the flat network topology. This means that, each node that is not in the backbone, has at least one backbone node as its neighbor. Backbone nodes are joined via *gateway* nodes. Apparently, if the flat network is connected, it is deterministically guaranteed, that the backbone is connected as well. Distributed Clustering Algorithm (*DCA*) is a high-performance representative of this category.

The second category retains the layered structure of the first category, and it is oriented on facilitating routing without the need of gateway nodes. The concept behind this category is that of building a *Connected Dominating Set (CDS)* directly, without firstly selecting CHs and then joining them. It was initially introduced by Das et al. in [16] when they proposed the concept of creating a *communication spine* inside a flat network topology in order to support unicast, multicast, and fault-tolerant routing within an ad hoc network. *WuLi* is a practical and robust representative of this category.

Contrary to *RanMIS* 's randomized nature, its competitors use iterative clustering

techniques, meaning that a node waits for a specific event to occur or certain nodes to decide about their role before making a decision themselves.

The first class, which enables *WuLi* [72], exhibit a high degree of localization. In such a class as soon as a node has collected information about its surrounding topology, it is able to decide whether it will be part of the backbone or not. The only information it needs to wait for, is the identity of the node in its (*h* hop) neighborhood. In the second class, the degree of localization shown by the algorithms is limited with respect to that of the first. Such protocols implement a distributed version of the heuristics for finding an *independent set* of nodes which is maximal, and a *dominating set* which is minimal. As a representative algorithm of this class we consider the *DCA* [8]. Being maximum, the *independent set* produced by *DCA* is also a minimal dominating set.

WuLi is a very simple distributed procedure consisting of a few local rules, the execution of which creates the desired *CDS*. Every node *v* exchanges its neighbor list with all its neighbors. A node set itself has a dominating node if it has at least two unconnected neighbors. In order to reduce the size of a *CDS*, the original protocol presents two pruning rules. According to the first rule, a node deletes itself from the *CDS* when its close neighbor set, which includes all of its direct neighbors as well as itself, is completely included in the neighbor set of a neighboring dominating node and it has smaller ID than the neighboring dominating node. According to the second rule, a node deletes itself from the *CDS* when its open neighbor set, which includes all of its direct neighbors, is completely included in the neighbor sets of two connected neighboring dominating nodes and has the smallest ID.

The *DCA* protocol assumes quasi-stationary nodes with real-valued weights, which are initially *UNDECIDED*. Node weights induce a total ordering of the nodes without any ties because node weights reflect real numbers. During the execution of the algorithm each node will decide to be *IN* or to be *OUT*, of the *independent set*. A node decides when all its neighbors with larger weight have decided. When the time comes, a node decides to be *OUT* if one of its neighbors is *IN*. Otherwise it decides to be *IN*. The *IN* nodes form the minimal dominating set required for clustering. The execution time depends on possible chains of dependency between the nodes, a negative consequence of the reduced localization *DCA* presents. The node with the smallest weight has to wait for all other nodes in the chain.

3.3.1 Simulation Model

As in most studies in multihop wireless networks, we used unit disk random graphs to represent the network topologies used in the performance analysis. A unit disk graph is determined by node positions and a fixed *communication range R*, for all nodes. The produced topologies which more precisely are described as *Constrained - Connected Random Unit Graphs (C-CRUG)*, because of the constrains the participating nodes are obliged to adhere, were constructed by taking in to consideration two crucial independent variables, concerning the final network connectivity, that is the network density, which is perfectly expressed by the *degree d* of each node, and the node *population N*.

3.3.2 Simulation platform

The graphs were normally generated by placing nodes randomly and independently from each other with the help of a modified version of *Minimum degree proximity algorithm (MIN-DPA)* [6], a *C-CRUG* generator, which aims to distribute node degrees more uniformly while maintaining connectivity.

The idea is to place each new node in the vicinity of the node that has the smallest number of neighbors, while preserving a minimum distance to increase the probability of achieving a connected graph. The modified version of it though, involves the notion that the average degree of the topology and the transmission range of a node, were predefined.

The simulations refer to scenarios in which N static wireless nodes with maximum transmission range R are randomly and uniformly scattered in a geographic area of side *L*. We make the assumption that two nodes are neighbors if and only if their Euclidean distance is less than *R*.

We emphasize that nodes start the protocol execution at the same time and run the clustering and backbone formation algorithms to form a hierarchical multi-hop ad hoc network.

In our simulations R has been set to 50m, the number of nodes N has been assigned the values 100, 500, 1000 and 5000 while *L* has been set accordingly (see Table 1) to influence an average node degree of 4, 7, 10 and 15. This allowed us to test the protocols on increasingly dense networks, from (moderately) sparse to dense networks.

Table 1: L Parameter Values

Nodes	Degree#4	Degree#7	Degree#10	Degree#15
100	500	400	250	200
500	2000	1500	750	500
1000	5000	4000	1000	700
5000	7000	5000	2000	1500

In order to diminish any statistical errors all tests were repeated for 1000 times and the data used in our graphs are all average values of the results. A short description of the simulation parameters is given in Table 2.

Table 2: Interpretation of Used Symbols

Symbol	Usage	Default value
n	Number of Nodes in Network	
D	Intended Number of Neighbors	24
d	Average Node Degree	
M	Constant	32
R	Node Transmission range	
L	Axis Length (in meters) of *AOR* (Area Of Responsibility)	

3.3.3 Performance metrics

The assessment of the competing protocols were done along three dimensions: the first dimension portrays the cost (in delay and energy consumption) incurred by the protocols, the second concerns the characteristics of the resulting backbone, and the third one deals with the resilience of the resulting spanner to node failures.

Metrics for protocol cost. This family of measures, includes the *protocol duration* which is a direct measure of the delay incurred until the network spanner is established,

and the *total number of messages exchanged* among nodes which is a measure of the network congestion.

Protocol duration. It is the number of rounds required by the protocol to complete the procedure of backbone formation.

Total number of messages transmitted. The situation concerns the backbone construction operation, i.e., the selection of network nodes which will take on the role of a CH, and the declaration of 'attachment' of non-CH nodes to a CH. To uncouple our measurements from MAC particularities and focus only to the pure clustering protocol performance, we assume an ideal MAC layer with no provision for retransmissions in case of collisions exist.

Metrics for backbone description. This family of measures assess the backbone's 'layout'.

Backbone size. It is the number of the network nodes that comprise the backbone. A smaller backbone is desirable for minimizing the routing overhead. For instance, in sensor networks with nodes that can turn off their radio interface (energy saving sleep mode), the backbone size is a measure of how many nodes need to stay awake for data transport. Usually, the nodes in the backbone stay up or have a higher duty cycle for guaranteeing routes to the sink, and hence the smaller the backbone the more nodes can be in energy conserving mode. On the other hand, a very small backbone could result in situations where CH-to-member communication would require either excessive amount of energy to reach the node and vice versa, or multi-hop communication within the cluster.

Cluster cardinality distribution. It is the distribution of the number of members for each generated cluster.

Route length. If we consider the subgraph G_b of the network induced by the backbone construction (G_b is the graph where there are no links between ordinary nodes), then this metric gives a measure of how well a routing protocol can perform over the backbone.

Inter-clusterhead distance distribution. It records the distances among neighboring CHs. The larger this distance is the better for the performance of the clustering protocol, since, for instance, a three-hop distance among clusterheads would avoid hidden terminal problems and guarantees the 'independence' of clusterhead transmission.

3.3.4 Simulation results

In order to evaluate the performance of *RanMIS* , we selected two graph-theoretic ad hoc networks clustering algorithms, namely the *WuLi* and *DCA* , both of which create dominating sets, the former produces a connected dominating set and the latter produces a maximum independent set. These algorithms are simple, practical, extremely popular in the clustering community [1] and comprise the base for the development of many similar clustering algorithms [20]. We performed a series of experiments to compare the performance of *RanMIS* against these algorithms. The first set of experiments concerns the tuning of *RanMIS* with respect to its parameters, and then it follows its comparison against its competitors.

[1] They both have an excessive number of citations. See
http://scholar.google.com/scholar?cites=7672109277762563412&as_sdt=2005&sciodt=0,5&hl=en
and
http://scholar.google.com/scholar?cites=13992908027776879246&as_sdt=2005&sciodt=0,5&hl=en.

3.3.5 Tuning of *RanMIS*

The original article which introduced *RanMIS* [1], described two parameters that control its operation and affect its overall performance. The first of these parameters, D, concerns the estimated size of neighborhood of each node, while the second, M, is a constant related to the number of rounds required for the algorithm to complete the backbone construction. These parameters are independent and thus we have to make an initial choice for one of them (e.g., for D) and investigate the other (e.g., M), and vice versa.

Impact of parameter D. To evaluate the impact of parameter D on *RanMIS* 's performance, we conducted a series of experiments for various values of D, 6, 12, 24 36, and 48 and for various performance measures. D controls the maximum number of rounds *RanMIS* has in order to complete the backbone construction and it is consistent with the size of each node's neighborhood. That is because in a large neighborhood the intuition is that the likelihood to have a simultaneous transmission and a resultant collision between two adjacent transmitting nodes is increased, and consequently more protocol rounds will be required by the affected nodes in order to decide about their status.

In Figure 3, we see the number of rounds required by *RanMIS* to complete backbone construction for various values of D w.r.t. network size and its density.

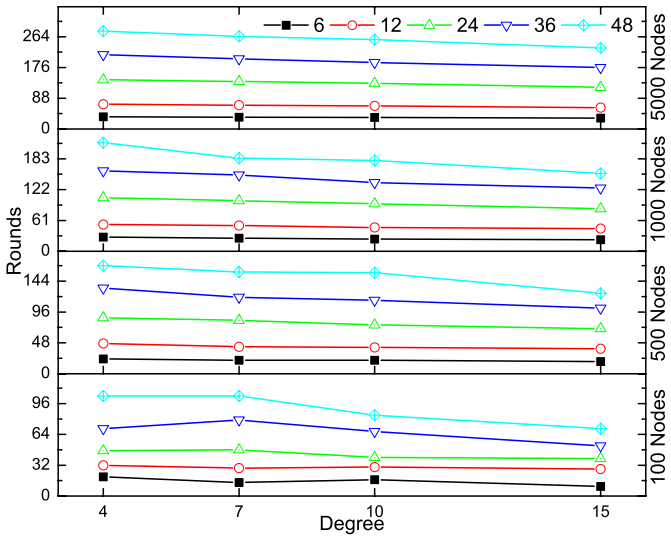


Figure 3: Impact of D on the number of Rounds required by *RanMIS* to complete backbone construction w.r.t. network size and its density.

The first observation is that, in accordance with our intuition, when the value of D increases, the number of rounds required by *RanMIS* to complete backbone construction increases for every network size. This consequence derives from the fact that there is a direct relation between the probability a node has in order to be chosen as a CH candidate and the setting of parameter D (cf. Figure 2). The higher the setting of D the lesser the probability for a node to become a CH and if so, the probability for a neighboring node to become a CH is also small (thus, we avoid any unwanted message retransmissions due to collisions). The second observation is that while the number of

rounds is almost irrelevant to network density for a given network size, the number of rounds increases linearly to the network size. This is due to the fact that more clusters will be developed and therefore the competition for becoming a clusterhead increases. It is obvious that the less the required rounds to complete the backbone is, the best for the protocol is (namely in Figure 3 this observation favors D setting of 6 and 12 to the other). Though, to conclude which is the best choice for parameter D, we must examine also the number of transmitted messages and the number of clusters created for the various choices of D (cf. Figure 4 and Figure 5). Thus, we do not conclude at this point that the best value for D is 6.

In Figure 4, we see the number of messages exchanged by *RanMIS* for backbone construction for various values of D w.r.t. network size and its density.

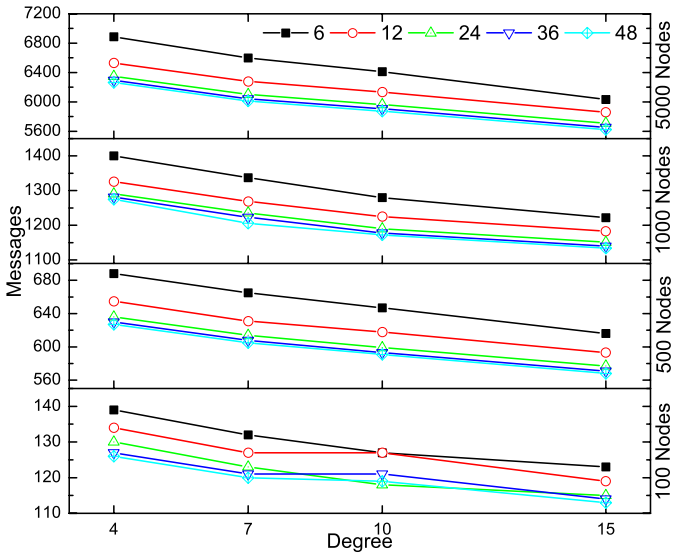


Figure 4: Impact of D on the number of messages exchanged by *RanMIS* for backbone construction w.r.t. network size and its density.

The situation here is reversed with respect to what we saw in Figure 3. In general, the number of messages required by *RanMIS* to complete the backbone construction process decreases for increasing values of D. Therefore, so far the best choice for D is the one that achieves a balance between the number of rounds and the messages transmitted, i.e., $D = 24$ (we will revisit this issue when commenting the next figure). Additionally, examining the figure, we observe a decreasing number of transmitted messages for increasing network density (for every network size). This is because when a node declares itself as a CH, then all of its neighbors (and there are too many nodes in dense networks) immediately attach themselves to this node. The final observation is that the relative ordering of the performance curves for various values of D is maintained across different densities (with some statistically insignificant variation for very small networks).

In Figure 5, we see clusters produced by *RanMIS* for backbone construction w.r.t. network size and its density for various values of D.

The main observation is that the number of clusters produced is not affected by parameter D. However, protocol execution drives to decreasing number of produced

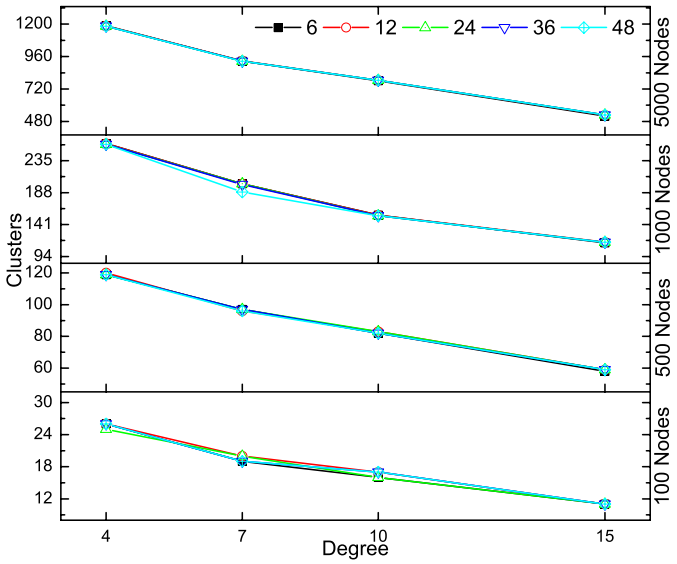


Figure 5: Impact of D on the number of clusters produced by *RanMIS* for backbone construction w.r.t. network size and its density.

clusters w.r.t. increasing network density for every network size, and the explanation for that is similar to the one for the previous figure.

Overall, *a value of* $D = 24$ *is the most appropriate choice* since it achieves a good tradeoff between network backbone construction delay (due to many rounds) and the transmitted messages.

Impact of parameter M**.** To evaluate the impact of constant M on *RanMIS* 's performance, we conducted a series of experiments for various values of $M = 2^x$ (x = 0 - 7), and for various performance measures. For these sets of experiments, the value of D was set equal to 24 (cf. subsection *Impact of parameter D*.)

In Figures 6, 7 we see the impact of M on the number of rounds required by *RanMIS* to complete the backbone construction. In Figure 6, it is clear that an M = 8 setting supersedes any other setting for practically all network instances (except for 100 nodes), since a small M means less work in the inner loop of *RanMIS* , that is less residency in a low probability state for a node to be chosen as a CH. Practically we bias the system to converge faster with small settings of M. The same observation holds in Figure 7, where for smaller values of M (namely 1, 2) the required number of rounds for convergence decreases.

On the other hand, a small M setting might create severe competition for the role of CH among neighboring nodes. Since M value affects the maximum number of rounds that are available for backbone construction, it is possible for a small value of M the protocol not to converge. In Figure 8, we see the impact of M on the % percentage of Max rounds used by *RanMIS* in order to complete the backbone construction. For small settings of M (e.g 1, 2) more than 80% of the available rounds were used by the protocol in order to converge.

In Figures 9, 10, we see the impact of M on the number of messages exchanged by *RanMIS* to complete the backbone. The observation is that for practically all network instances the number of transmitted messages is unrelated to M.

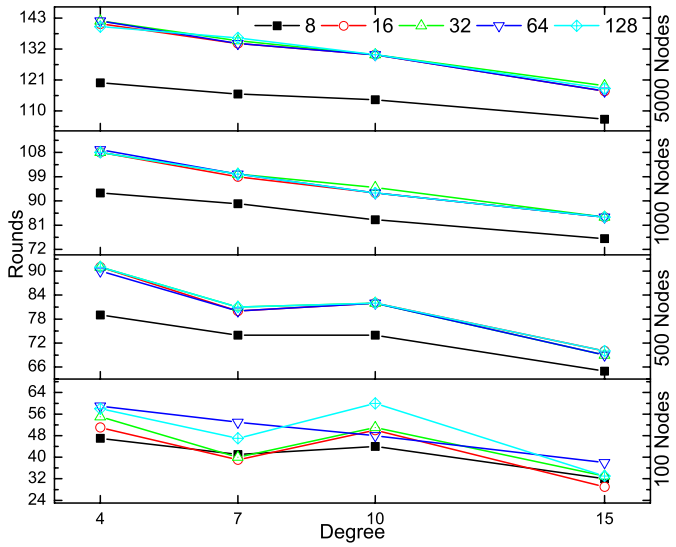


Figure 6: Impact of M on the number of rounds required by *RanMIS* to complete backbone construction w.r.t. network size and its density.

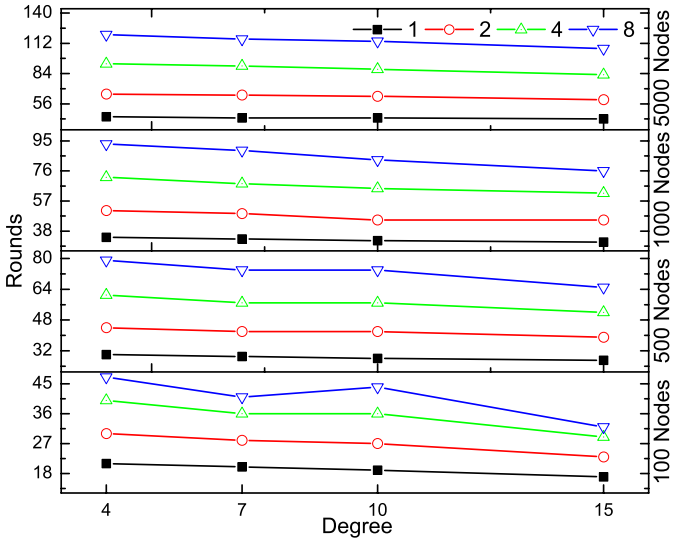


Figure 7: Impact of M on the number of rounds required by *RanMIS* to complete backbone construction w.r.t. network size and its density.

In Figure 11, 12, we see the impact of M on the number of clusters produced by *RanMIS* . Similarly, the number of clusters produced is not affected by M.

Overall, it seems that *the authors' suggestion of* $M = 34$ *is not the best choice*; it is only marginally good for small networks (100 nodes) when we want to minimize the number of transmitted messages – the difference though is not really significant. We concluded though, that we should be conservative with the setting of M, especially

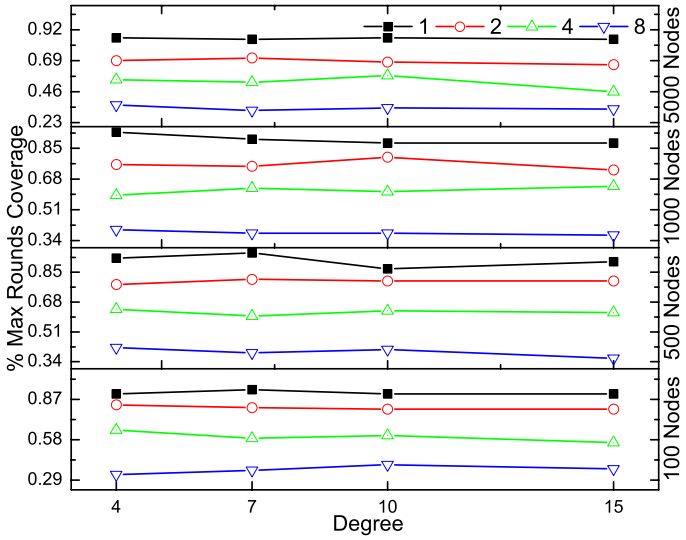


Figure 8: Impact of M on the % percentage of Max rounds required to complete the backbone construction w.r.t. network size and its density.

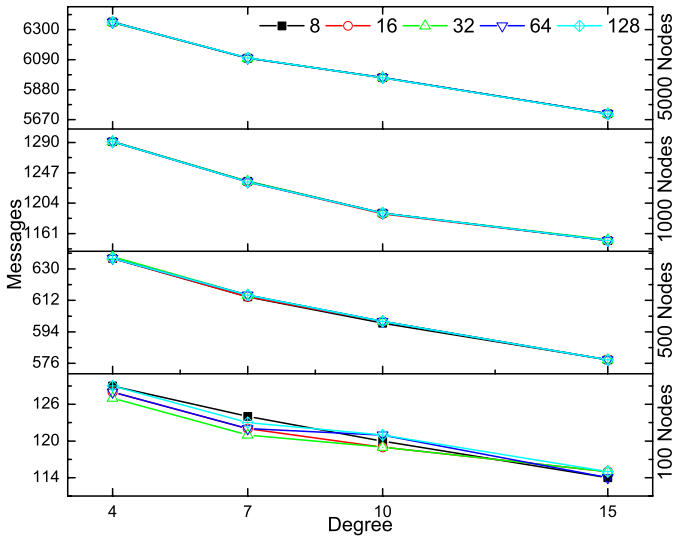


Figure 9: Impact of M on the number of messages exchanged by *RanMIS* for backbone construction w.r.t. network size and its density.

when we have to deal with real life network topologies. Nevertheless, in the rest of this article, we follow the *RanMIS* 's creators suggestion and set the value of M equal to 32.

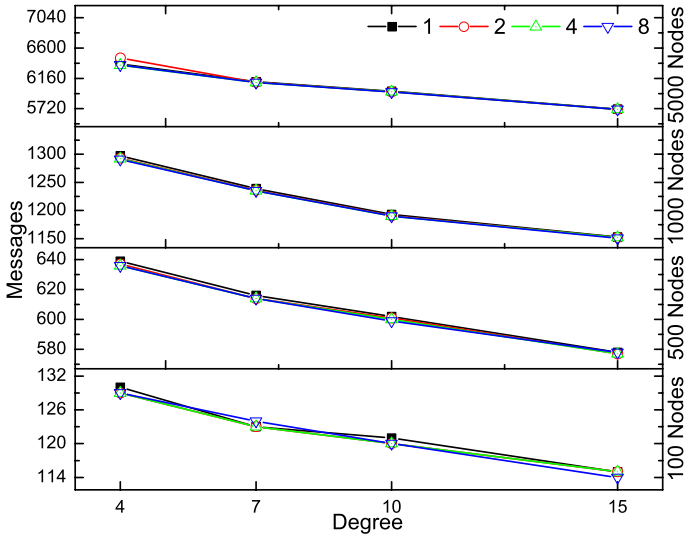


Figure 10: Impact of M on the number of messages exchanged by *RanMIS* for backbone construction w.r.t. network size and its density.

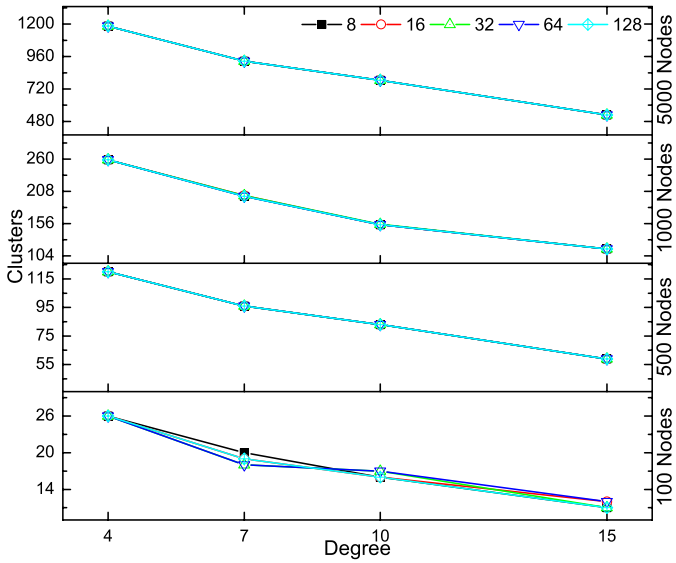


Figure 11: Impact of M on the number of clusters produced by *RanMIS* w.r.t. network size and its density.

3.3.6 Comparison of competing clustering protocols

Results concerning the protocol cost. In this series of experiments, we measured the number of rounds required by each protocol to complete and also the total number of messages transmitted. In Figures 13 and 14, we see the relative comparison of the

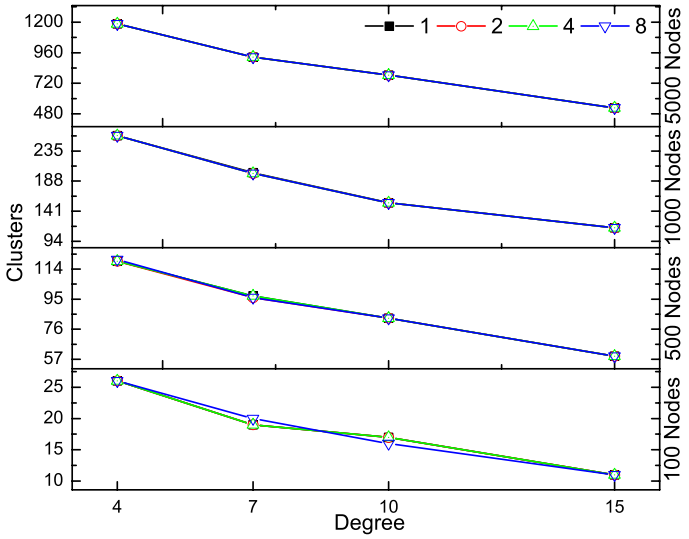


Figure 12: Impact of M on the number of clusters produced by *RanMIS* w.r.t. network size and its density.

algorithms.

As far as the number of protocol rounds is concerned, we observe that *DCA* is the clear winner for all network instances. The performance of the algorithms is consistent with the what the theory predicts for them; thus we do not comment further on this issue.

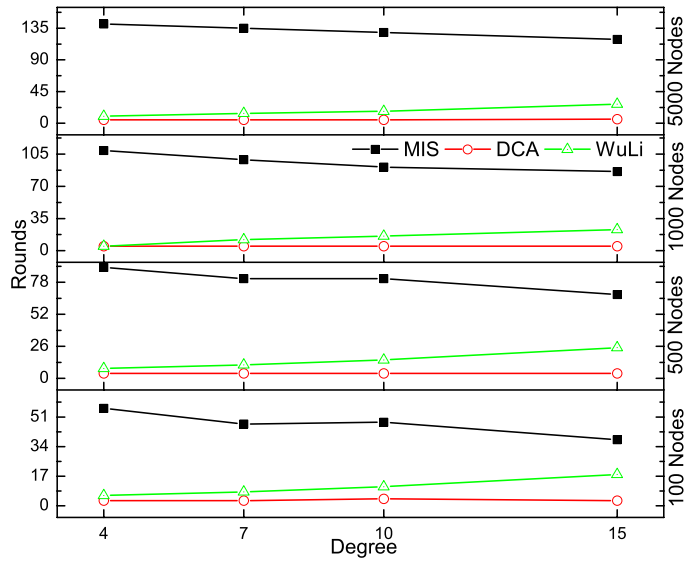


Figure 13: Rounds required by each competitor algorithm to complete backbone construction w.r.t. network size and its density.

As far as the number of transmitted messages for backbone construction is concerned, we observe that *RanMIS* is now the clear winner confirming its theoretical behavior. Its performance gap from the other competitors is stable across all network topologies. *DCA* sends around 40% more messages than *RanMIS* does, and *WuLi* sends around 60% more messages than *RanMIS* . Therefore, even though it takes more time to *RanMIS* to construct the backbone due to its probabilistic nature, it does this with far less messages than its competitors.

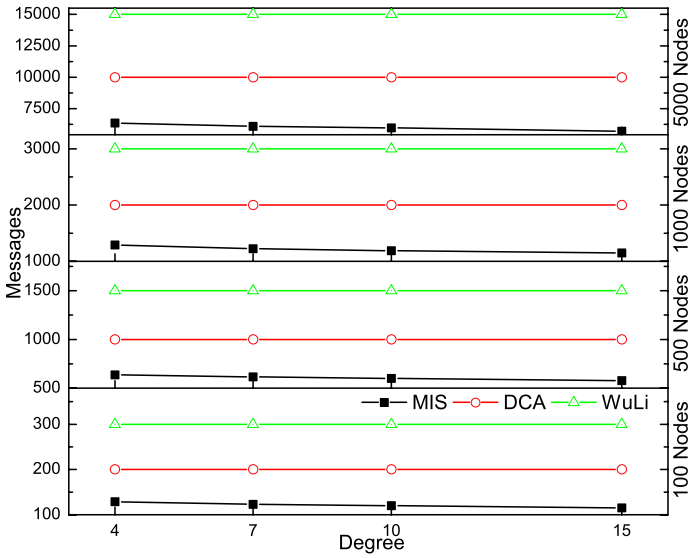


Figure 14: Messages required by each competitor algorithm to complete backbone construction w.r.t. network size and its density.

Results concerning the backbone description. In Figures 15–22 we present the performance of the protocols for the metrics belonging to the category of "backbone description".

The number of generated clusters by each protocol is presented in Figure 15. The first observation is that *RanMIS* and *DCA* produce almost the same number of clusters; even though their difference is indistinguishable in the figure, their difference is less than 1% which can be attributed to statistical variation. The reason for this resemblance is that both algorithms are building maximum independent sets using either a weight for each node (assigned dynamically or statically) in the case of *DCA* or using randomization in choice in the case of *RanMIS* . On the other hand, *WuLi* produces too many clusters, since it is mandatory to create *connected* dominating sets, whereas the former two algorithms are building plain dominating sets. Finally, we observe that the number of generated clusters reduces with increasing density, since in dense networks more nodes are able to find a CH (i.e., dominator) in their 1-hop neighborhood.

Next, we evaluated the cluster cardinality distribution for various densities of the network. The results are illustrated in Figures 16–19. At this point we should emphasize that this distribution should in general be 'bell-shaped' or look like any distribution with probability mass concentrated around the average node degree of the topology. In Figures 16 and 17, we see that this property is achieved by *RanMIS* and *DCA* for sparse

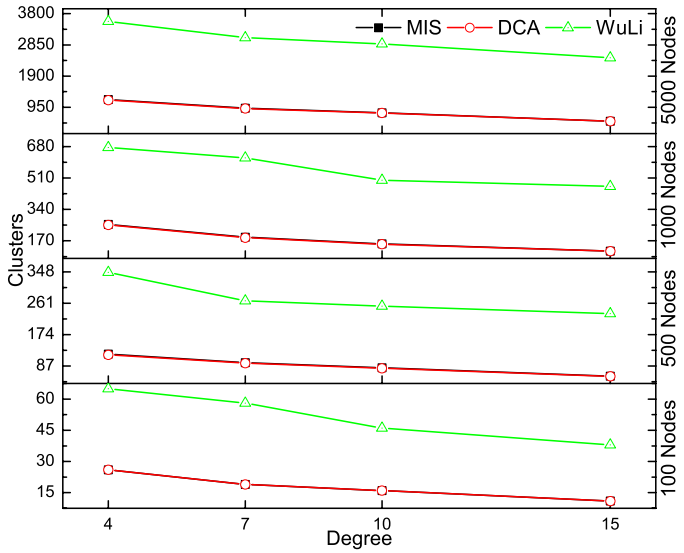


Figure 15: Clusters produced by each competitor algorithm w.r.t. network size and its density.

networks, but it is not achieved for dense networks (in Figures 18 and 19). In these last two figures, it seems that a significant percentage of clusters – around 22% of them – contain very few nodes, i.e., 1 or 2, including the clusterhead. On the other hand, *WuLi* exhibits an undesirable pattern across all network topologies, since more than 70% of its clusters contain 1 or 2 nodes, which is also a consequence of the large numbers of clusters produced (see Figure 15).

The diameter[2] length of the produced backbone by each protocol is presented in Figure 20 (in meters) and in Figure 21 (in hops). The common observation is that *WuLi* is the best performing algorithm which is due to the fact that it produces connected dominating sets and thus there are no gateway nodes intervening among CHs so as to increase the diameter. The second observation is that the diameter in general decreases with increasing network density, since in dense networks it easier to find short routes for any pair of nodes. In some cases where the topology is peculiar, the diameter increases in denser networks (e.g., network with 500 nodes with density equal to 7).

In Figure 22 we illustrate the percentage of adjacent CHs that reside at a distance of 3-hops from each other. Apparently, *WuLi* produces clusterheads at 1-hop distance from each other, and therefore we do not include it in the plot. For the other two algorithms, which produce maximum independent sets, it holds by the definition of the MIS, that any two adjacent CHs will be either at a distance of 2 hops or at a distance of 3 hops, with the latter being the preferred one (see paragraph on 'Inter clusterhead distance distribution' definition, Section 3.3.3). We see that both algorithms achieve approximately the same performance – the gap among them is statistically insignificant. In particular, we see that the majority (more than 50%) of adjacent CHs in almost all network topologies, are 3-hops away. The only exception appears for very small and very sparse networks (i.e., 100 nodes with degree 4), where this percentage drops to 48%; this is due to the fact that there are not too many links among the nodes to

[2]The largest of the shortest paths for any pair of nodes.

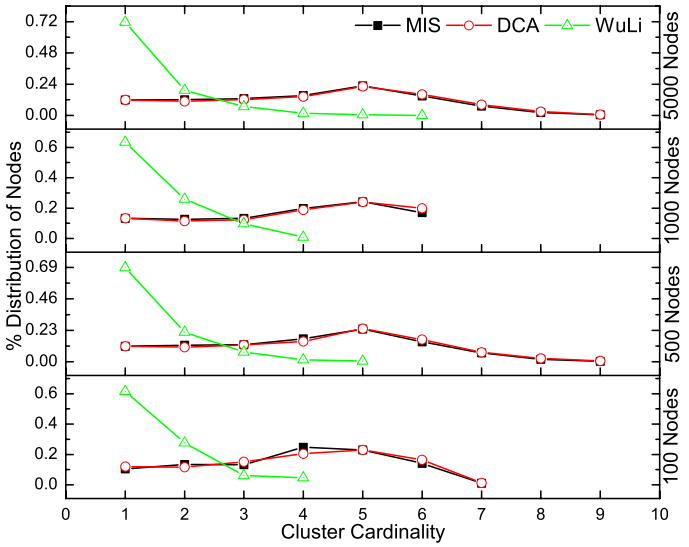


Figure 16: Distribution of nodes in clusters after backbone construction w.r.t. network size and a density of $D = 4$.

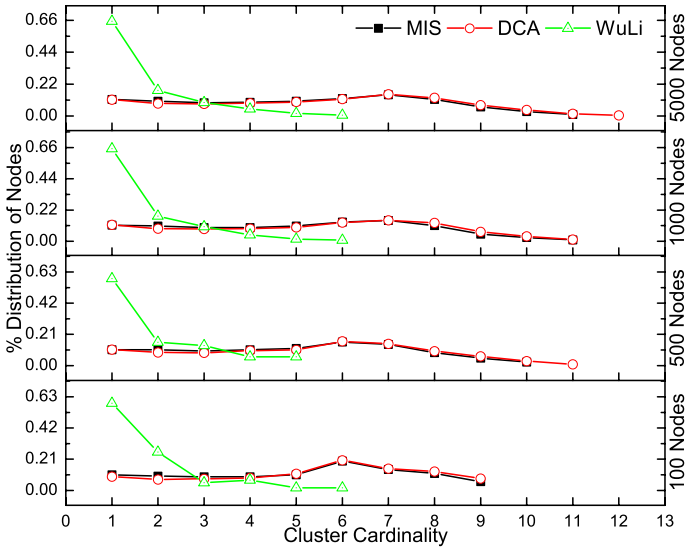


Figure 17: Distribution of nodes in clusters after backbone construction w.r.t. network size and a density of $D = 7$.

establish 3-hop distance among neighboring CHs.

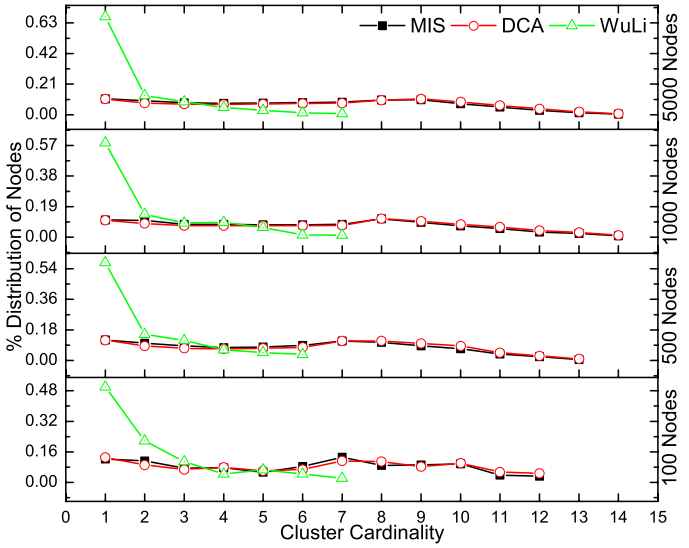


Figure 18: Distribution of nodes in clusters after backbone construction w.r.t. network size and a density of $D = 10$.

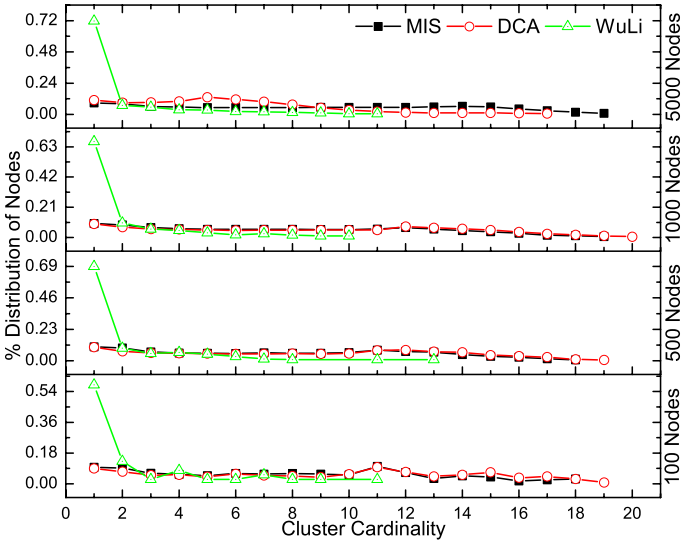


Figure 19: Distribution of nodes in clusters after backbone construction w.r.t. network size and a density of $D = 15$.

3.4 The *SPRING* clustering algorithm

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

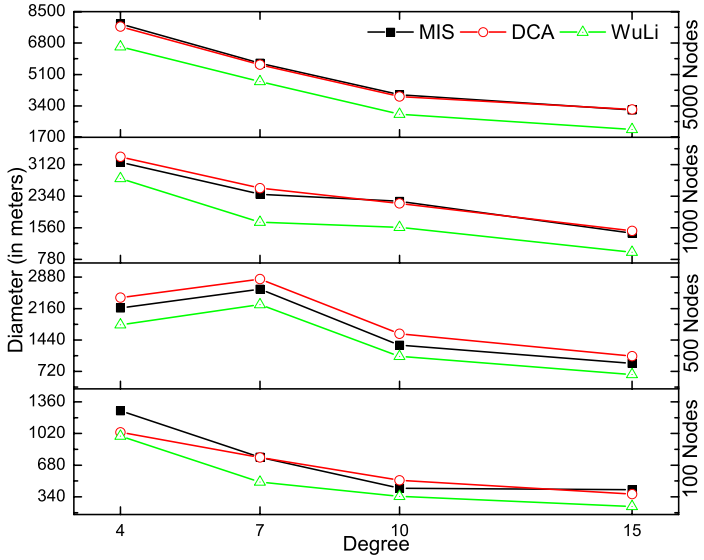


Figure 20: Network Diameter diversification (in meters) after backbone construction w.r.t. network size and its density.

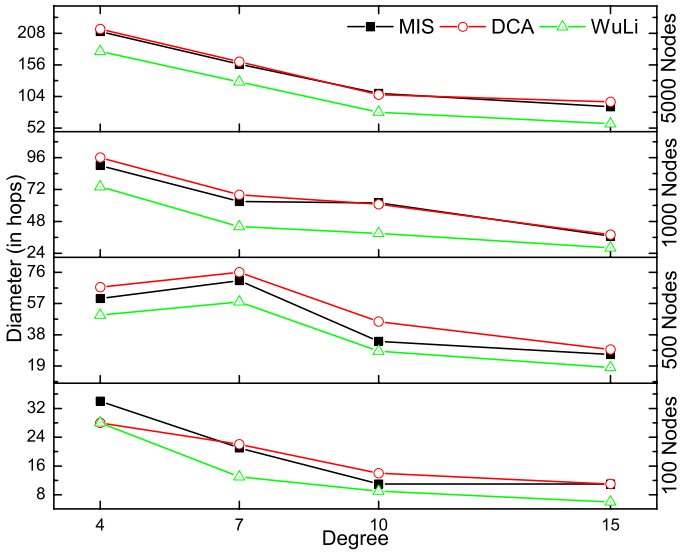


Figure 21: Network Diameter diversification (in Hops) after backbone construction w.r.t. network size and its density.

forces are applied to the nodes, pulling them closer together or pushing them further apart. Every node apply to its neighbors a force Frel 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 Frel along the east-west Fx and north-south Fy axes are calculated. In order

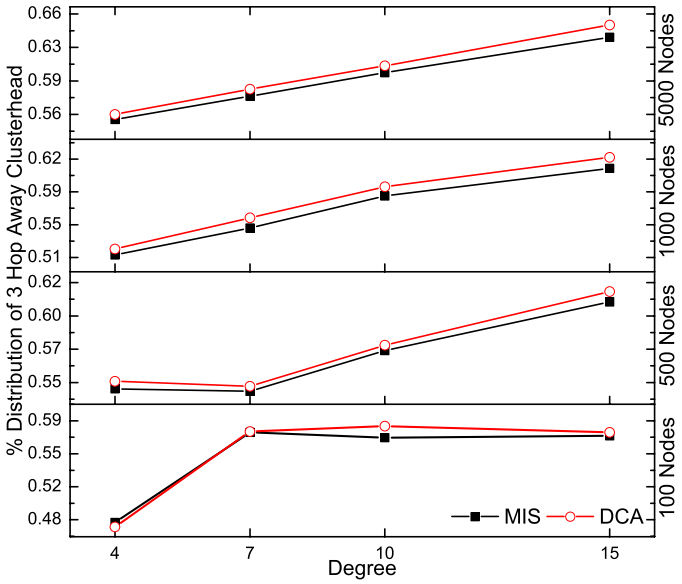


Figure 22: CH Distance Distribution after backbone construction w.r.t. network size and its density.

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 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. 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. The Maximum Transmission Range (R) of each node in the vehicular network environment is 250 meters.

3.4.1 Network identification

Neighborhood identification is the process whereby a vehicle/node identifies its current neighbours within its transmission range. For a particular vehicle, any other vehicle that is within its radio transmission range is called a neighbour. All vehicles consist of neighbour set which holds details of its neighbour vehicles. The neighbours 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,neighbour 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 timestamp. 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 intimate their presence by sending beacon messages frequently. After the reception of a beacon, each node will update its neighbour 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 neighbours by sending a beacon signal. If a known neighbour, times out, it will be removed from the neighbour set table. The total number of neighbors of a given vehicle is called the "active neighborhood set" N_i of the vehicle.

3.4.2 Clustering process and protocol structure

As described in the previous section the beaconing thread is responsible for exchanging information 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.

$$F_{relijx} = k_{ijx}\frac{q_i q_j}{r_{ij}^2}, \quad F_{relijy} = k_{ijy}\frac{q_i q_j}{r_{ij}^2} \tag{1}$$

where r_{ij} is the current distance among the nodes k_{ijx} (k_{ijy}) is a parameter indicating whether 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: (1) relative mobility parameters k_{ijx} and k_{ijy}.

Definition

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,

$$D_{cxij} = x_i - x_j, \quad D_{fxij} = x_i + dx_i - x_j - dx_j \tag{2}$$

$$D_{cyij} = y_i - y_j, \quad D_{fyij} = y_i + dy_i - y_j - dy_j \tag{3}$$

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. According to mobility in every axis relative mobility k_{ijx} and k_{ijy} are calculated according to:

$$if \quad D_{cxij} \leq D_{fxij} \quad then \quad k_{ijx} = -a_x dt. \tag{4}$$

$$if \quad D_{cxij} \geq D_{fxij} \quad then \quad k_{ijx} = a_x dt. \tag{5}$$

where a_x and a_y are given by

$$if \quad D_{cxij} \leq D_{fxij} \quad then \quad a_x = D_{fxij} - D_{cxij} \tag{6}$$

$$if \quad D_{cxij} \geq D_{fxij} \quad then \quad a_x = \frac{1}{D_{cxij} - D_{fxij}} \tag{7}$$

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 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 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.

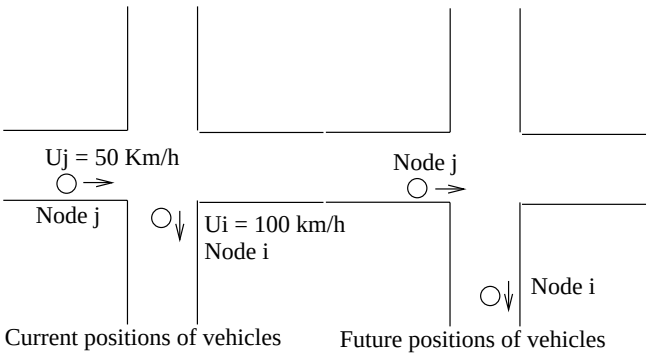


Figure 23: Relative mobility at node i with respect to node j.

3.4.3 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

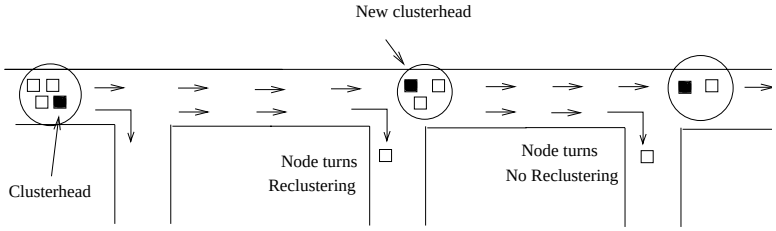


Figure 24: The correct choice of the clusterhead plays significant role

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 24).

Historical data about driver behavior may be used in order to select the appropriate clusterheads. In Sp-Cl parameter q_j is used to favour vehicles to become clusterheads. Using equation 1 to compute the relative force between two nodes, parameter q_j is used as follows:

$$if \quad k_{ijx} \geq 0 \quad then \quad q_i = 2, \qquad if \quad k_{ijx} \leq 0 \quad then \quad q_i = 1/2 \tag{8}$$

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.

3.4.4 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 timestamp. 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:

$$F_x = \sum_{j \in N_i} F_{relijx} \quad and \quad F_y = \sum_{j \in N_i} F_{relijx} \tag{9}$$

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:

$$F = |F_x| + |F_y| \tag{10}$$

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.

3.4.5 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.

3.4.6 Cluster maintenance

The cluster maintenance 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 are 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.

3.4.7 Demo

An extensive simulation study was conducted to evaluate the performance of our protocol. A custom simulator was used to evaluate our method. 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.

Scenario 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.

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.

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

Table 3: 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 4.

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

Table 4: Speed per lane.

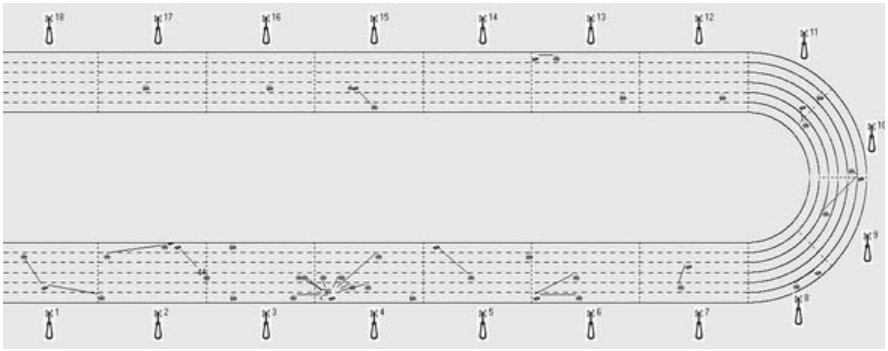


Figure 25: Simulation environment

3.4.8 Evaluation

Cluster stability.
In order to evaluate the stability of the clustering 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 and joins a nearby cluster or becomes free.
- A cluster-head merges with a nearby 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 26, 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.

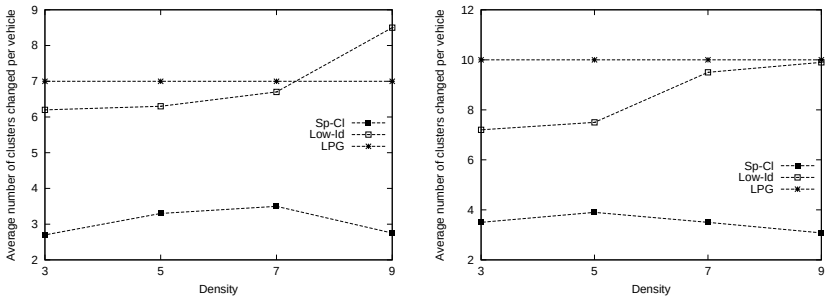


Figure 26: Average cluster change per vehicle for Sp-Cl, LPG and Low-Id methods for different transmission ranges(125m, 80m).

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 vanished over time. The total number of clusters created over a period of 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 vehicle's mobility defines its performance. In this simulation we counted the new clusters which are added to the system.

The figure 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.

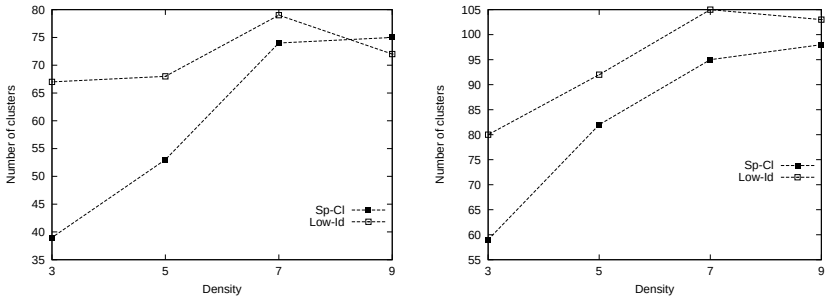


Figure 27: The average total number of formed clusters for Sp-Cl and Low-Id methods for different transmission ranges(125m, 80m).

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

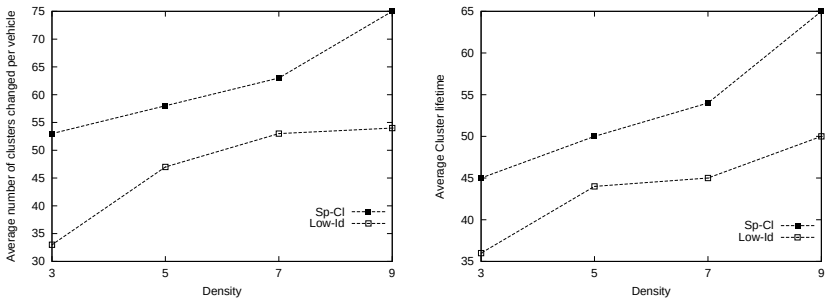


Figure 28: The average cluster lifetime Sp-Cl and Low-Id methods for different transmission ranges(125m, 80m).

4 Delay-aware cross-layer packet scheduling

The development of a throughput-optimal routing algorithm for packet radio networks which is also robust to topology changes was first presented in [61]; it is based on the Lyapunov drift theory, and it is known as the *backpressure* ($\mathscr{BP}$) packet scheduling algorithm. Subsequently, the original concept spawn several lines of research in the topic. 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 underwater acoustic sensor nets which are the target of our work.

To circumvent the delay problems of backpressure, the *shadow queue* algorithm [13] 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 [75] and thus the computational overhead, and as a side-effect, reduce the delay. To alleviate the delay problems of backpressure, scheduling based on the *combination of backpressure and shortest-paths* have been proposed [74], 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 [27, 43] could be incorporated in an *orthogonal way* to improve analogously the delay performance of all policies at the expense of throughput optimality, but this is beyond the scope of this paper. Here, we take an 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.

4.1 The quest for delay-aware packet scheduling protocols

In the shadow queues ($\mathscr{SQ}-\mathscr{BP}$) methods [7, 13], 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 can not be scheduled. This means that packets stay in queues longer, which can lead to higher delays. Indeed, we tested this intuitive result by evaluating the delay performance of these methods (see a sample performance in Figure 29) and confirmed this behavior. Therefore, since the $\mathscr{SQ}-\mathscr{BP}$ methods are not delay competitive, we do not consider them as viable competitors any more.

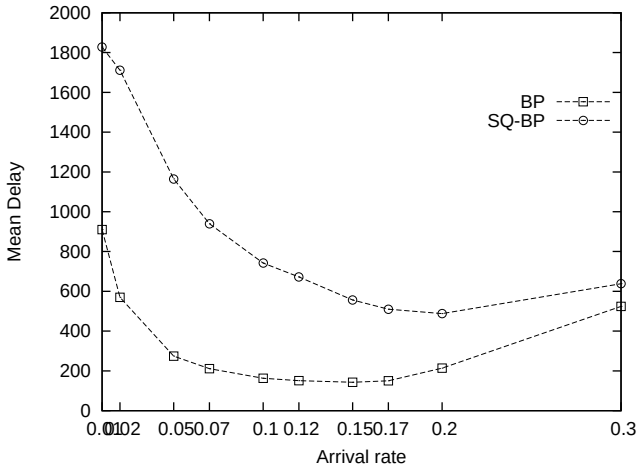


Figure 29: Performance of the $\mathscr{SQ}-\mathscr{BP}$ policy in a 4x4 grid network ($M = 2$).

The authors of [75] 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 $\mathscr{BP}$), and proposed the creation of clusters[3] of nodes so as to reduce the number of queues per node, which as a "side-effect" has the additional benefit of delay reduction. Their policy, namely *cluster-based backpressure* ($\mathscr{CB}-\mathscr{BP}$), requires maintaining one queue per *gateway* at each relay node which leads to an excessive number of gateways, which is 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 [3, 9] (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 $\mathscr{CB}-\mathscr{BP}$ approaches asymptotically the delays of the original $\mathscr{BP}$ (cf. Figure 30c). 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 $\mathscr{CB}-\mathscr{BP}$ 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.

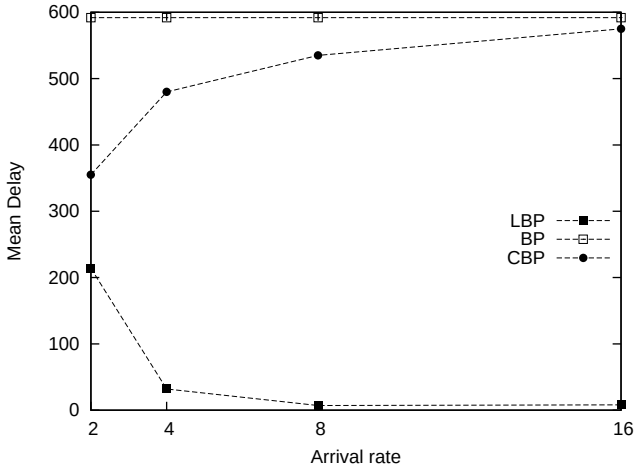


Figure 30: Impact of the number of clusters on the grid network topology.

The Shortest-paths backpressure ($\mathscr{SP}-\mathscr{BP}$) method [74] assumes the precomputation 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

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

where there is only one shortest-path per node pair (which is the most common case), $\mathscr{SP}-\mathscr{BP}$ 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 paths traverse the same set of nodes (i.e., nodes with high betweenness centrality [29]); in such topologies, these nodes deplete their energy so fast, that the network gets partitioned very rapidly.

4.2 Distributed backpressure-based packet scheduling

Max-weight scheduling, also known as backpressure routing, is a cross-layer control algorithm that is well-known to be throughput optimal since it provides queue stability within the network for all traffic injection rates within the network's capacity. Max-weight scheduling relies on global knowledge of full system state information of the network which is not realistic in ad-hoc networks. Implementing backpressure in a distributed way is a challenge. In order to implement the backpressure in ad-hoc networks, nodes need to disseminate some kind of information in order to make the optimal cross-layer control decisions.

The centralized scheduling algorithm selects the node with the highest differential queue backlog first and thus gives higher priority to a congested node for transmission. In distributed and random access based fashion, a node with higher differential queue backlog should be allowed to access the medium with higher probability.

Authors in [35] developed a distributed scheduling algorithm for multi-channel ad hoc wireless networks that automatically track the changes in the network topology and offered load. They introduce a two stage queueing mechanism for packet scheduling. The method has a significant lower capacity region than Greedy Maximal Scheduling.

In [49] authors explore the idea of disseminating only a limited amount of state feedback for this algorithm and evaluate the subsequent impact on performance. Queue backlogs are represented by K bits. The values are calculated using threshold values uniformly It is shown when using uniformly distributed threshold values, a tradeoff exists between the overhead required in information dissemination and performance gains from more effective scheduling The exact number of bits that is necessary for the correct behavior of the system is a major drawback of the proposed method.

In [55] when a packet is transmitted to any destination, the transmitting node inserts its queue length for that destination onto the packet. All one-hop neighbors overhear this transmission and update their queue length values for the node-destination pair. The algorithms are fully distributed and requires only one-hop information.

Lately backpressure has been used in traffic signal control. In [33] a distributed algorithm is presented where the signal at each junction is locally controlled independently from other junctions. Their approach relies on constructing a set of local controllers, each of which is associated with a junction. These local controllers are constructed and implemented independently of one another. The method seems to overcome SCATS, an adaptive traffic signal control systems that is being used in many cities.

4.3 The original Backpressure algorithm

Backpressure [61] 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:

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

  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:

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

  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).

4.4 Layered backpressure

This section describes a delay-efficient backpressure algorithm based on the creation of *layers*[4] 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 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 $\mathscr{L}ay\mathscr{B}\mathscr{P}$ scheduling policy, every node n keeps a separate queue for each destination 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 [75], because $\mathscr{L}ay\mathscr{B}\mathscr{P}$ does not require the knowledge of gateways' queue lengths. The "correct" partitioning of nodes into layers and a proximity-based naming of layers is

[4] 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.

of paramount importance in order to improve the mean end-to-end delay. 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. For the description of layer-creation and layer-naming algorithms, the reader is directed to [38], whereas subsection 4.4.1 presents the Layered Backpressure packet scheduling algorithm.

4.4.1 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 conrtoller observes the queue backlog matrix $Q(t) = (Q_n^d(t))$ and performs the following actions for routing:

Layered BackPressure at time slot t:

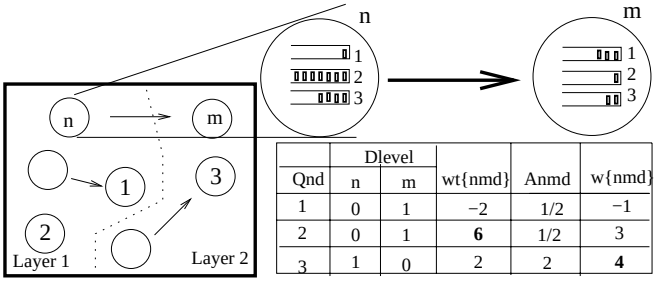
- Each link (n,m) is assigned a temporary weight according to the differential backlog $wt_{nmd}(t) = (Q_n^d - Q_m^d)$ and parameter A_{nmd} according to:

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

- Each link is assigned a final weight according to w_{nm} and parameter A_{mnd}:
  $w_{nm}(t) = \max_{d \varepsilon D}(wt_{nmd}(t) * A_{nmd})$
- The network conrtoler chooses the control action that solves the following optimization:
  $\mu^*(t) = \text{argmax} \sum_{n,m} \mu_{nm} w_{nm}(t)$
  subject to the interference model mentioned above where adjacent links are not allowed to be active simultaneously.

A simple example is illustrated in Figure 31. The network in Figure 31 consists of 7 nodes where all nodes are senders and only nodes with id's $1,2,3$ are destination nodes. In Figure 31 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 m, which according to the proposed layer backpressure policy will prevent the packet from returning to node n.

Theorem 1 (Throughput optimality) *The $\mathscr{L}ay\mathscr{B}\mathscr{P}$ algorithm is 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**

BP : Optimal commodity for link (n,m) on slot t is 2. w(n,m) = 6
LBP : Optimal commodity for link (n,m) on slot t is 3. w(n,m) = 4

Figure 31: An example of the $\mathscr{L}ay\mathscr{B}\mathscr{P}$ execution.

Proof.
The original backpressure algorithm is throughput optimal. In order to prove that $\mathscr{L}ay\mathscr{B}\mathscr{P}$ is also throughput optimal, the Lyapunov stability criterion 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.

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

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

This equation can be rewritten in the following form:

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

which is equivalent to :

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

where weights θ_i^d are used to offer priority service similar to [45].

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.

According to [22], we have:

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

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:

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

It is not difficult to show that:

$\Delta L(Q) \leq 2BN - 2\sum_{nd} \theta_n^d \varepsilon Q_n^d(t)$,
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:

$\Delta L(Q) \leq 2B'N\theta_{max} - 2\sum_{nd} \theta_n^d \varepsilon Q_n^d(t)$,
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 sum of all queues is:

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

4.5 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 $\mathscr{CBB}-\mathscr{BP}$, $\mathscr{LayBP}$) 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 $\mathscr{LayBP}$ 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 $\mathscr{LayBP}$ behavior can be affected of 'misplaced' nodes. In this case packet may be forwarded to layers different than the desired making the method inappropiate.

In order to cope with node mobility we incorporate in $\mathscr{LayBP}$ algorithm another step in which moving 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. Only moving nodes update the information on their counters in order to find the appropriate layer. The algorithm does not perform reclustering, but only 'helps' layers incorporate moving nodes.

4.6 Dynamic networks

4.6.1 Experimental setting

We evaluated the performance of the cluster-based algorithms in networks with mobile nodes in order to measure the adaptivity of the $\mathscr{EL}-\mathscr{BP}$ method. It is important here to mention that $\mathscr{EL}-\mathscr{BP}$ 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 $\mathscr{SP}-\mathscr{BP}$ 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 topologies, we considered that illustrated in Figure 32. 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). $\mathscr{BP}$, $\mathscr{CB}-\mathscr{BP}$ and $\mathscr{L}ay\mathscr{BP}$ are unaware of layer change while all the active links of the moving node are updated. In $\mathscr{EL}-\mathscr{BP}$ method, update algorithm runs at every time slot and moving nodes join the appropriate layer according to neighbor's information.

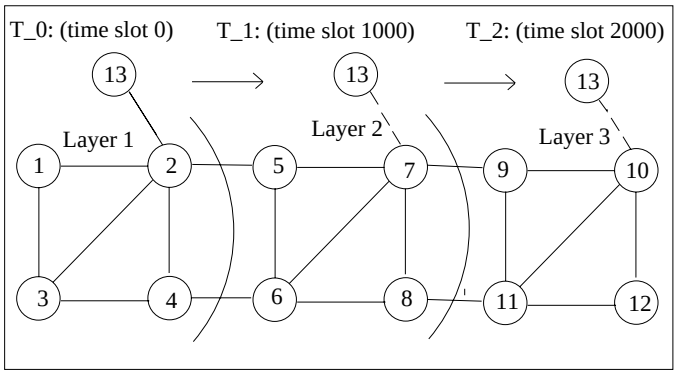


Figure 32: Dynamic network with 3 layers.

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.

4.6.2 Experimental results

Delay and throughput performance. It is clear that $\mathscr{CB}-\mathscr{BP}$ 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, makes it impossible for 'misplaced"' nodes to receive packets. Because of this behavior $\mathscr{CB}-\mathscr{BP}$ is not simulated in these experiments. The plots in Figure 33 show the delay performance of the

$\mathscr{L}ay\mathscr{BP}$, $\mathscr{EL}-\mathscr{BP}$ and the original backpressure. It is clear $\mathscr{L}ay\mathscr{BP}$ behaves worse than the original $\mathscr{BP}$ in cases of moving nodes. The $\mathscr{EL}-\mathscr{BP}$ 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 $\mathscr{L}ay\mathscr{BP}$ degrades, while $\mathscr{EL}-\mathscr{BP}$ retains its behavior.

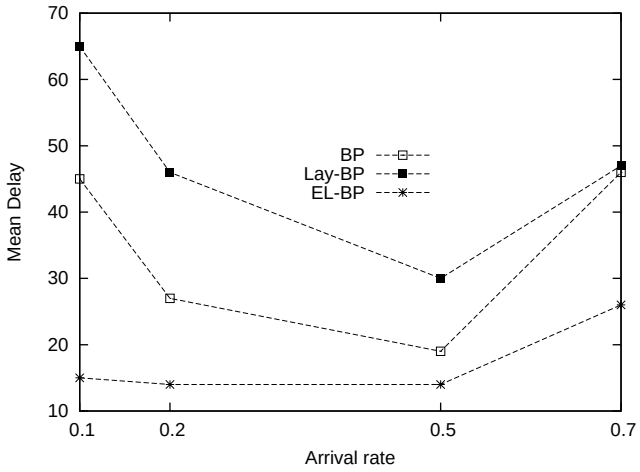


Figure 33: Impact of moving nodes on the delay performance of $\mathscr{BP}$, $\mathscr{L}ay\mathscr{BP}$ and $\mathscr{EL}-\mathscr{BP}$ for the dynamic network with the 3 layers.

The plots in Figure 34 depict the throughput performance of the competing methods[5]

5 Conclusions

Our work on the wireless communications and vehicular networking aspects of REDUCTION during the first year of the project revealed the benefits of the two layer architecture in combination with a packet scheduling/routing mechanism that guarantees throughput optimality and at the same time strives for low delay in packet dissemination. In the context of these efforts, we need some further work with respect to evaluating the considered and proposed clustering protocols, and efforts to tuning the developed backpressure protocols into a fully distributed algorithm, since now it depends a lot of a centralized controller.

References

[1] Y. Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331(6014):183–185, 2011.

[5] An extensive performance analysis of $\mathscr{EL}-\mathscr{BP}$ is described in [40].

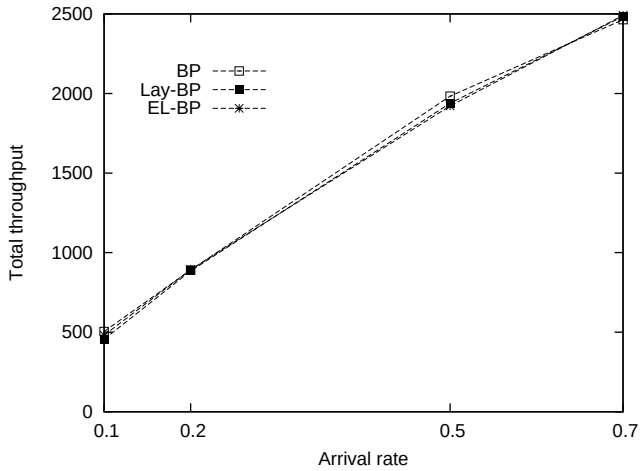


Figure 34: Impact of moving nodes on the throughput performance of $\mathscr{BP}$, $\mathscr{LayBP}$ and $\mathscr{EL-BP}$ for the dynamic network with the 3 layers).

[2] M. S. Almalag and M. C. Weigle. Using traffic flow for cluster formation in vehicular ad-hoc networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN)*, pages 631 –636, 2010.

[3] A. D. Amis and R. Prakash. Clusterhead selection in wireless ad hoc networks. US Patent 6,829,222 B2, 2004.

[4] A.D. Amis, R. Prakash, T.H.P. Vuong, and Huynh D.T. Max-min *d*-cluster formation in wireless ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 32–41, 2000.

[5] V.S. Anitha and M. P. Sebastian. (*k*,*r*)-dominating set-based, weighted and adaptive clustering algorithms for mobile ad hoc networks. *IET Communications*, 5(13):1836–1853, 2011.

[6] F. Atay, I. Stojmenovic, and H. Yanikomeroglu. Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks. In *Proceedings of the IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 597–615, 2008.

[7] 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.

[8] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, pages 310–315, 1999.

[9] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE Transactions on Parallel and Distributed Systems*, 17(4):292–306, 2006.

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

[11] A. Benslimane, T. Taleb, and R. Sivaraj. Dynamic clustering-based adaptive mobile gateway management in integrated VANET-3G heterogeneous wireless networks. *IEEE Journal on Selected Areas in Communications*, 29(3):559 –570, 2011.

[12] 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.

[13] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM) Mini*, 2009.

[14] M. Chatterjee, S. K. Das, and D. Turgut. WCA: A weighted clustering algorithm for mobile ad-hoc networks. *Journal of Cluster Computing*, 5:193–204, 2002.

[15] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks*, 8(5):481–494, 2002.

[16] B. Das and V. Bharghavan. Routing in ad hoc networks using minimum connected dominating sets. In *Proceedings of the IEEE International Conference on Communications (ICC'97)*, pages 376–380, 1997.

[17] M. K. Denko, J. Tian, T. K. R. Nkwe, and M. S. Obaidat. Cluster-based cross-layer design for cooperative caching in mobile ad hoc networks. *IEEE Systems Journal*, 3(4):499–508, 2009.

[18] 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.

[19] E. Dror, C. Avin, and Z. Lotker. Fast randomized algorithm for 2-hops clustering in vehicular ad-hoc networks. *Ad Hoc Networks*, 2012. To appear.

[20] K. Erciyes, O. Dagdeviren, D. Cokuslu, and D. Ozsoyeller. Graph-theoretic clustering algorithms in mobile ad hoc networks and wireless sensor networks - Survey. *Applied and Computational Mathematics*, 6(2):162–180, 2007.

[21] P. Fan, J. Haran, J. Dillenburg, and P. Nelson. Cluster-based framework in vehicular ad-hoc networks. In *Ad-Hoc, Mobile, and Wireless Networks*, volume 3738 of *Lecture Notes in Computer Science*, pages 32–42. 2005.

[22] L. Georgiadis, M. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1), 2006.

[23] M. Gerla and J.T.-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM Wireless Networks*, 1(3):255–265, 1995.

[24] J. Ghaderi, L.-L. Xie, and X. Shen. Hierarchical cooperation in ad hoc networks: Optimal clustering and achievable throughput. *IEEE Transactions on Information Theory*, 55(8):3425–3436, 2009.

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

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.

[27] 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.

[28] 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.

[29] 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):2–9, 2010.

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

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

[32] S. Leng, Y. Zhang, H.-W. Chen, L. Zhang, and K. Liu. A novel *k*-hop compound metric based clustering scheme for ad hoc wireless networks. *IEEE Transactions on Wireless Communications*, 8(1):367–375, 2009.

[33] I. Leontiadis, G. Marfiay, D. Mackz, C. Mascolo, G. Pauy, and M. Gerla. On the effectiveness of an opportunistic traffic management system for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1537–1548, 2012.

[34] F. Li and Y. Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology magazine*, pages 12–22, 2007.

[35] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 1118–1126, 2007.

[36] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee. Routing protocols in vehicular ad hoc networks: A survey and future perspective. *Journal of Information Science and Engineering*, 26:913–932, 2010.

[37] Y. Luo, W. Zhang, and Y. Hu. A new cluster based routing protocol for VANET. In *Proceedings of the International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, pages 176–180, 2010.

[38] L. Maglaras and D. 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.

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

[40] L. Maglaras and D. Katsaros. Layered backpressure routing for delay reduction in ad hoc networks, 2012. Submitted.

[41] C. Maihöfer. A survey of geocast routing protocols. *IEEE Communications Surveys & Tutorials*, 6(2):32–42, 2004.

[42] C.-Q. Mei, Huang H.-J., and Tang T.-Q. Improving urban traffic by velocity guidance. In *Proceedings of the IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 383–387, 2008.

[43] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali. Routing without routes: The backpressure collection protocol. In *Proceedings of the IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.

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

[45] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas on Communications*, 23(1):89–103, 2005.

[46] M. Ni, Z. Zhong, and D. Zhao. MPBC: A mobility prediction-based clustering scheme for ad hoc networks. *IEEE Transactions on Vehicular Technology*, 60(9):4549–4559, 2011.

[47] S. Panichpapiboon and W. Pattara-atikom. A review of information dissemination protocols for vehicular ad hoc networks. *IEEE Communications Surveys & Tutorials*, 2012. To appear.

[48] GeoNet EU FP7 project. http://www.geonet-project.eu/.

[49] S. T. Rager, E. N. Ciftcioglu, A. Yener, T. F. La Porta, and M. J. Neely. Distributed backpressure protocols with limited state feedback. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 693–698, 2011.

[50] B. Ramakrishnan, R. S. Rajesh, and R. S. Shaji. CBVANET: A cluster based vehicular adhoc network model for simple highway communication. *International Journal of Advanced Networking and Applications*, 2(4):755–761, 2011.

[51] E. Sakhaee and A. Jamalipour. A new stable clustering scheme for pseudo-linear highly mobile ad hoc networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pages 1169–1173, 2007.

[52] R. A. Santos, O. Alvarez, and A. Edwards. Performance evaluation of two location-based routing protocols in vehicular ad-hoc networks. In *Proceedings of the IEEE Vehicular Technology Conference - Fall (VTC-Fall)*, volume 4, pages 2287–2291, 2005.

[53] R. A. Santos, R. M. Edwards, and N. L. Seed. Using the cluster-based location routing (cblr) algorithm for exchanging information on a motorway. In *Proceedings of the International Workshop on Mobile and Wireless Communications Network*, pages 212–216, 2002.

[54] 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.

[55] U. K. Shukla. Backpressure policies for wireless ad hoc networks, 2010. Technical Report, Virginia Polytechnic Institute and State University.

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

[57] T. Song, W. Xia, T. Song, and L. Shen. A cluster-based directional routing protocol in VANET. In *Proceedings of the IEEE International Conference on Communication Technology (ICCT)*, pages 1172–1175, 2010.

[58] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[59] L. Sun, Y. Wu, J. Xu, and J. Xu. An RSU-assisted cluster head selection and backup protocol. In *Proceedings of the International IEEE Conference on Advanced Information Networking and Applications Workshops (AINAW)*, pages 581 –587, 2012.

[60] C. Suthaputchakun and Z. Sun. Routing protocol in intervehicle communication systems: A survey. *IEEE Communications magazine*, 49(12):150–156, 2011.

[61] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 1992.

[62] 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.

[63] 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-Fall)*, volume 1, pages 319–323, 2005.

[64] S.-L. Tsao and C.-M. Cheng. Design and evaluation of a two-tier peer-to-peer traffic information system. *IEEE Communications magazine*, 49(5):165–172, 2011.

[65] C. Tselikis, S. Mitropoulos, N. Komninos, and C. Douligeris. Degree-based clustering algorithms for wireless ad hoc networks under attack. *IEEE Communications Letters*, 16(5):619–621, 2012.

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

[67] S. Vodopivec, J. Bester, and A. Kos. A survey on clustering algorithms for vehicular ad-hoc networks. In *Proceedings of the International Conference on Telecommunications and Signal Processing (TSP)*, pages 52–56, 2012.

[68] T. Wang and G. Wang. TIBCRPH: Traffic infrastructure based cluster routing protocol with handoff in VANET. In *Proceedings of the Wireless and Optical Communications Conference (WOCC)*, 2010.

[69] A. Wegener, H. Hellbruck, C. Wewetzer, and A. Lubke. VANET simulation environment with feedback loop and its application to traffic light assistance. In *Proceedings of the IEEE Global Communications Conference Workshops (GLOBECOMW)*, 2008.

[70] S. Widodo, T. Hasegawa, and S. Tsugawa. Vehicle fuel consumption and emission estimation in environment-adaptive driving with or without inter-vehicle communications. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 382–386, 2000.

[71] 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.

[72] J. Wu and H. Li. A dominating-set-based routing scheme in ad-hoc wireless networks. *Telecommunication Systems*, 18:13–36, 2001.

[73] Y. Xia, C. K. Yeo, and B. S. Lee. Hierarchical cluster based routing for highly mobile heterogeneous MANET. In *Proceedings of the International Conference on Network and Service Security (N2S)*, 2009.

[74] L. Ying, S. Shakkotai, and A. Reddy. On combining shortest-path and back-pressure routing over multihop wireless networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2009.

[75] L. Ying, R. Srikant, and D. F. Towsley. Cluster-based back-pressure routing algorithm. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 484–492, 2008.

[76] O. Younis and S. Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.

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

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

[79] 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.