

Article

Epidemic and Timer-Based Message Dissemination in VANETs: A Performance Comparison

Pietro Spadaccino [†], Francesca Cuomo ^{*,†} and Andrea Baiocchi [†]

Department of Information Engineering, Electronics and Telecommunications (DIET), University of Rome La Sapienza, 00184 Rome, Italy; spadaccino.1706250@studenti.uniroma1.it (P.S.); andrea.baiocchi@uniroma1.it (A.B.)

* Correspondence: francesca.cuomo@uniroma1.it

† These authors contributed equally to this work.

Received: 26 February 2020; Accepted: 28 March 2020; Published: 31 March 2020



Abstract: Data dissemination is among the key functions of Vehicular Ad-Hoc Networks (VANETs), and it has attracted much attention in the past decade. We address distributed, efficient, and scalable algorithms in the context of VANETs adopting the paradigm. We introduce an epidemic algorithm for message dissemination. The algorithm, named EPIC, is based on few assumptions, and it is very simple to implement. It uses only local information at each node, broadcast communications, and timers. EPIC is designed with the goal to reach the highest number of vehicles “infected” by the message, without overloading the network. It is tested on different scenarios taken from VANET simulations based on real urban environments (Manhattan, Cologne, Luxembourg). We compare our algorithm with a standard-based solution that exploits the contention-based forwarding component of the ETSI GeoNetworking protocol. On the other hand, we adapt literature based on a connected cover set to assess the near-optimality of our proposed algorithm and gain insight into the best selection of relay nodes as the size of the graph over which messages are spread scales up. The performance evaluation shows the behavior of EPIC and allows us to optimize the protocol parameters to minimize delay and overhead.

Keywords: vehicular networks; data dissemination; epidemic algorithms

1. Introduction

In the framework of smart cities and environments, a fundamental role will be played by VANETs (Vehicular Ad-Hoc Networks). These networks are set up in an ad-hoc fashion by vehicles and provide direct communication among them, without the support of external infrastructure. Originally conceived essentially for vehicular safety, VANETs are also used for road monitoring and infotainment applications in the framework of the intelligent transportation system paradigm.

VANETs are designed to increase safety and driving efficiency and make the driving experience more comfortable. As for safety, it has been demonstrated that about 60% percent of accidents can be avoided by sufficient warnings. This can be pursued by using direct vehicular message dissemination in emergency situations. VANETs will also be used to gather Floating Car Data (FCD) for urban sensing and vehicular traffic control.

In all cases, Vehicle-to-Vehicle (V2V) communication systems provide a 360 degree view of similarly-equipped vehicles within communication range, and multi-hop dissemination messages can be spread or collected in different Regions of Interest (RoI). In this case, VANET vehicles themselves act as the relay nodes of the network, giving the possibility to forward the time-critical (e.g., emergency) messages, independently of the availability of external network infrastructures.

Many challenges are posed by the implementation of VANETs, which should provide first of all a high quality of services able to guarantee, with a very high probability, the correct forwarding of a message through the network. Another challenge is to achieve a minimum latency in the dissemination of information over an RoI. This is especially important in the case of critical messages, e.g., for emergency situations. When dealing with broadcast-based dissemination, also efficiency is a concern, i.e., the minimization of the number of replicas of the same message that are transported by the network to cover the RoI.

The VANET is a, particular type of “ad-hoc” [1] network; hence it is self-configuring and designed to operate autonomously. Each mobile node is free to move independently while communicating. The protocols used to communicate are specified in the standard IEEE 802.11p [2], which is an amendment of IEEE 802.11. On top of the IEEE 802.11p stack, one of the most important goals is to define and implement an efficient dynamic routing algorithm that can help disseminate a message to all vehicles roaming in a given RoI. Defining a simple, distributed, efficient routing algorithm is challenging because of the highly dynamic topology of the network, where each node is in constant movement. The challenge is to reduce the delay associated with passing the information from a node to another and making the algorithm as fast, reliable, and efficient as possible, while requiring the least possible local control information and overhead.

Epidemic paradigms can represent a valid model to follow. These models have been studied to analyze the behavior of infectious diseases and to model their spreading in the population, but they have turned out to be relevant and useful also for technological fields, e.g., to model the propagation of malware or, like in our specific case, to model the dissemination of a message in an ad-hoc network. There are different types of epidemic models. The simplest is the Susceptible-Infected (SI) model. In this model, each node can assume only two states: (i) susceptible, if it has not been infected yet; (ii) infected, if the node has got the disease. In terms of message dissemination in a VANET, infection consists of getting a message; hence, a vehicle is susceptible as long as it has not received a copy of the message yet. In this simple model, we assume that a vehicle remains in the status “infected” for the rest of the time after receiving the message. Another model is SIS (Susceptible-Infected-Susceptible), similar to SI, but with the difference that the “infected” state only lasts for a given time period. After that time, the node returns to a ‘susceptible’ state. Yet another variant, which is the one that inspired our protocol, is Susceptible-Infected-Recovered (SIR) [3]. Recovered means that a node cannot be infected any more (it is immune), nor does it contribute to the spreading of the disease. In the VANET application, this means that a vehicle that has received the message (thus turning its state from susceptible to infected) has to decide whether to relay the message or not. Once the decision is made, either way, the vehicle will not take part in the dissemination of the same message any more (it becomes recovered). The contribution of this paper is threefold: (i) we introduce EPIC, a protocol for disseminating messages in VANETS with the aim of being based on few assumptions and very simple to implement in a distributed fashion; (ii) we compare EPIC with a theoretical bound derived from the literature on the connected cover set that allows us to assess the near-optimality of our proposed algorithm; (iii) we compare EPIC with the Contention-Based Forwarding (CBF) approach, which is defined by the ETSI standardization bodies, and with an evolution of it (named CBF+); we show, in real urban scenarios, the performance gain of our approach both in terms of capability to reach a great number of vehicles and in the reduced number of involved relay nodes to attain this satisfactory result. The rest of the paper is structured as follows. Section 2 discusses key related works on epidemic dissemination in VANETs. The EPIC model and algorithm are presented in Section 3, while the relevant performance assessment is in Sections 4 and 5. Finally, conclusions are given in Section 6.

2. Related Work

A VANET dissemination logic is used to select a sub-set of vehicles that are situated along the road, to act as relay nodes. Message dissemination in VANETs has been discussed in several papers in the last few years, where different challenging aspects have been highlighted [4]. Data dissemination

is fundamental to transport information to intended receivers while meeting certain design objectives, e.g., high delivery ratio. In pure ad-hoc VANETs, where a network infrastructure does not exist, messages move from one vehicle to another in order to reach the intended receivers. While flooding is a possibility to disseminate data in a VANET, it may result in being very inefficient due to the high load generated in the system. The dissemination logic has to select only a sub-set of vehicles to act as relay nodes to avoid the broadcast storm problem [5]. As an example, the paper [6] aimed at selecting as relaying vehicles those that were located at preferred positions, while inhibiting others. The protocols in this category can be identified as cluster-based relay where the dissemination logic operates by identifying clusters and cluster heads for an efficient dissemination. The paper [7] proposed different solutions to collect real-time FCD information efficiently in Dedicated Short-Range Communication (DSRC)-enabled VANETs. The goal was to improve the efficiency of the FCD collection operation while keeping the impact on the DSRC communication channel as low as possible. We do this by exploiting a slightly modified version of a standardized data dissemination protocol to create a backbone of relaying vehicles that, by following local rules, generate a multi-hop broadcast wave of collected FCD messages. The proposed protocols are evaluated via realistic simulations under different vehicular densities and urban scenarios.

Another way to face the dissemination problem is to rely on probabilistic approaches. One branch of these approaches is represented by the epidemic models that were proposed in the past to solve the probabilistic communication and information dissemination in ad-hoc networks [8] and in distributed systems [9]. In [10], M.Nekovee et al. showed the benefits of using epidemic algorithms in vehicular ad-hoc networks. One key advantage that epidemic algorithms offer is that they do not need any information about the network topology. This perfectly fits the requirements of VANETs because of their dynamic topology and the absence of an infrastructure to which to refer. An epidemic algorithm, as explained in [11], mimics the spread of a contagious disease: Each vehicle relays the information it has received to randomly chosen peers, rather than to a specific node. Each node decides if it has to re-transmit the message or not, according to some infection rules. The simplest epidemic example is flooding, where each node rebroadcasts all messages it receives. Many variations have been proposed to minimize the redundancy of flooding, for example probabilistic, distance-based, and location-based algorithms. An adaptive bio-inspired epidemic dissemination protocol for wireless sensor-actuator networks was presented in [12]. An analytical model of message dissemination optimal control to minimize the accumulated network cost was provided in [13], with applications to mobile and social networks. Epidemic modeling has been also applied extensively to malware spreading in networks (e.g., see [14]).

In [15], it was shown how the performance of a VANET degraded as the packet flow increased, and the authors proposed a probabilistic algorithm as a solution to reduce the overload. One fundamental issue during flooding is the spurious forwarding, that is when multiple vehicles are committed to forward the same packet when it is not necessary, i.e., when a vehicle forwards a message even if all its neighbors have already received it. Situations like this have a catastrophic impact on the network in the case of a high density of nodes and channel congestion, limiting performance and functionality. With probabilistic approaches, the impact of spurious forwarding is reduced by introducing a parameter p (decimation probability), so that a node gives up forwarding with probability p and becomes a forwarder with probability $1 - p$.

Furthermore, the work in [16] addressed probabilistic dissemination algorithms, defining an analytical model to infer optimal re-broadcast probabilities. Our approach is based instead on timers, rather than probabilistic forwarding. Reliable message broadcast was the focus of [17]. The authors aimed at providing a reliable broadcasting of messages in a vehicular network. The potential of non-orthogonal multiple access for V2V message dissemination was explained in [18]. The paper in [19] targeted reliable delivery of emergency messages. The authors focused on coding and the optimal number of repetitions of the message to combat the unideal vehicular radio channel. In [20],

the authors defined a message dissemination protocol suited for urban environments. It exploited knowledge of the street layout, e.g., adapting its behavior to the presence of intersections.

In this paper, we present EPIC, as the preliminary proposed in [21], and we provide an extensive performance evaluation by comparing it with both a theoretical bound provided by leveraging the graph theory and with an algorithm, Contention-Based Forwarding (CBF), defined by the ETSI standardization bodies [22].

Epidemic approaches applied to the data dissemination in VANETS can be found also in [23]. In [23], the proposed algorithm was based on the SIR approach enhanced with a selection of the relay only among the ones sending back a passive acknowledgment message and at the farthest distance. Compared with this approach, EPIC, does not use control messages, apart from the beaconing service, which is used by the vehicles to update the positions of neighbors, since all the metadata needed for the algorithm to work are contained in the short header of the message being disseminated. Moreover, another important aspect is that EPIC is evaluated in real urban scenarios.

3. EPIC Model and Algorithms

The leading idea of EPIC is as follows. A vehicle A , which receives a message, estimates if one or more of its neighbors have already received a copy of the same message, broadcast by other relay nodes. If the majority of the neighbors of A are believed to have already received the message, then A will not relay the message itself. A can estimate the coverage of its neighbors thanks to: (i) a list of previous relays carried in the header of the message; (ii) the knowledge of the position of its neighbors. The execution of the algorithm does not require any dedicated control message. Each vehicle executes the algorithm autonomously, i.e., based only on: (i) events that it observes at its network interface; (ii) information stored in the vehicle node; (iii) information contained in the disseminated message.

In the following, we assume that each vehicle is equipped with GPS (hence knows its position) and maintains an updated database of its neighbor vehicles, the so-called Local Dynamic Map (LDM) [24]. The LDM is updated thanks to beaconing [25] (notice that the LDM includes only the neighbors of a vehicle, so it gives no knowledge about the overall network topology).

3.1. Message Structure

The dissemination protocol header carried by the message contains the following main fields:

1. *ID* (four bytes), a message identifier;
2. *EMITTERS* (variable size), a list of records, one for each previous relay node of the message;
3. *TTL* (one byte), the number of remaining hops before stopping the dissemination at $TTL = 0$. *TTL* is decremented by one by each relay node and set to a given initial value by the application at the originator of the message.

Each record in the *EMITTERS* list is comprised of: the MAC address of the previous relay node (*ADDR* = 6 bytes), its GPS coordinates (*COORD* = 12 bytes), a timestamp of when the relaying took place (*TS* = 4 bytes). The *EMITTERS* list is completed by a one byte field carrying the current number of records listed. The first element in the list corresponds to the message originator, the last one is the last relay node from which the message has been received.

Table 1 lists the data type and size of the message fields. The parameter h indicates the number of nodes that have already relayed the message whose coordinates are stored in the *EMITTERS* message header. The maximum value of h is denoted as N_{eml} . As a consequence, the message length is $L = 4 + 1 + 1 + (12 + 6 + 4) \cdot h + \ell = 6 + 22 \cdot h + \ell$, where ℓ is the length of the message payload.

To reduce the overhead, it is possible to shorten the list of relay nodes (*EMITTERS*) recorded in the message header to $N_{eml} < TTL$ nodes, trading off the performance of the dissemination protocol in terms of efficiency (reduced number of redundant copies of the message) with the overhead carried by each relayed message.

Table 1. Message header: field, type, and size.

Field	Type	Size (Bytes)
<i>ID</i>	Integer	4
<i>EMITTERS</i>	List	$1+(12+6+4) \cdot h$
<i>TTL</i>	Integer	1

3.2. EPIC Algorithm

The EPIC algorithm is conceived of as the spreading of an “infection”, given the analogy of message dissemination to a population of vehicles roaming in the RoI with the spreading of a disease in a population of susceptible individuals. This spread follows the SIR model described in [3]. EPIC uses the same states and transitions of the SIR model, as shown in Figure 1, and these states refer to each vehicle for a given message M . This means that for two different messages, M and M^* , i.e., $M^* \neq M$, the same vehicle can be in different states. On the contrary, for the same message M , a vehicle is in exactly one state at any given time. The time spent in a state depends on the events described in the caption of Figure 1.



Figure 1. States and transitions of the Susceptible-Infected-Recovered (SIR) model. The same are used in EPIC, and they refer to a specific message. Let us identify it as M , which is sent in the VANET. Each vehicle that never received the message M is in the susceptible state. When it receives the message M , it makes a transition to the infected state, setting a timer Δt . Once the timer expires, the vehicle decides whether to rebroadcast a message or not (this is part of the EPIC algorithm and is described in the following) and makes a transition to the recovered state. When a vehicle is in this latter state, it discards all the next copies of M .

Algorithm 1 describes the behavior of a vehicle every time it receives a message. For the sake of simplicity, we will consider the dissemination of a single message.

Each vehicle starts in the *SUSCEPTIBLE* state, meaning that it has not yet received a copy of the message. Let us consider a tagged vehicle A . Once a copy of the message arrives at A , the procedure is executed, changing A 's state from *SUSCEPTIBLE* to *INFECTED* and starting a timer of duration Δt :

$$\Delta t = \max \left\{ T_{min}, T_{max} \left(1 - \frac{d}{R_{max}} \right) \right\} \quad (1)$$

where:

- T_{max} and T_{min} are respectively the maximum and minimum waiting time before sending a message in broadcast.
- R_{max} : the maximum communication range, i.e., the maximum range within which message delivery is successful with a high probability.
- d : distance between the receiver and transmitter of the message.

An additional parameter is used in the EPIC procedure, *EvaluatePositions*, namely R_{min} . R_{min} is the minimum communication range considered by the Algorithm 2, i.e., it is assumed that two nodes within distance R_{min} can communicate successfully with probability one.

The timer value depends on the distance d between A and the vehicle node B from which A has received the message. The coordinates of B are known to A , since they are carried in the message header (last element of the *EMITTERS* list).

Algorithm 1 Procedure to be executed upon message receipt.

```

procedure ONMESSAGERECEIPT
  input:
  msg ← received message
  state ← vehicle state, can be SUSCEPTIBLE, INFECTED, or RECOVERED

  main:
  if state = RECOVERED then
    return

  else if state = SUSCEPTIBLE then
    rcv_messages ← empty list
    state ← INFECTED
    timer ← new timer( $\Delta t$ )
    rcv_messages.append(msg)

  else if state = INFECTED then
    rcv_messages.append(msg)

  on timeout:
  do_relay ← EvaluatePositions
  if do_relay and msg.TTL > 0 then
    msg ← updateMessage(msg)
    relayMessage(msg)
  state ← RECOVERED

```

While the timer is running, *A* remains in the *INFECTED* state. In that state, *A* collects all other possibly received copies of the same message (*A* can recognize that a received packet contains a copy of a previously received message by looking at the message *ID* and originator source address). *A* appends all received copies in the “rcv_messages” list.

Upon timer expiry, *A* uses the content of the “rcv_messages” list to understand whether it should relay the message. For that purpose, *A* runs the procedure *EvaluatePositions* described in Algorithm 2. If this evaluation returns TRUE and if the current hop counter of the message is less than *TTL*, then *A* updates the message header fields and broadcasts the message. If instead, the outcome of the procedure *EvaluatePositions* returns FALSE (i.e., relaying the message is not needed, since all neighbors of *A* should have already received it), node *A* does not relay the message and drops it (after having delivered the data to the upper layer).

After the decision about whether to relay the message or not, the protocol state machine of *A* moves to the *RECOVERED* state. Once in this state, the vehicle will ignore all further copies of the message.

Algorithm 2 uses a heuristic to identify whether or not a vehicle has to relay a message. The heuristic is based on the knowledge of the position of the previous emitters. This position is used to infer whether the message has been or not received by the neighbors of the current vehicle. Notice that being a heuristic, it may return a result that does not always match the reality. In the following, we present more detail of the steps performed to implement this heuristic. In the initialization step, the current vehicle creates a set of all emitters found in the headers of all received copies of the message (in case *EMITTERS.ADDR* is duplicated, the emitters list picks only the emitter with the recentest *TS*field). Then, in the main phase, the heuristic iterates over each emitter *E* in the set and checks whether any neighbor *V* of *A* is within R_{\min} of *E*. If this is the case, the heuristic infers that *V* has already received the message from *E*. Hence, *A* removes *V*’s position from the “not_reached” list that it had initialized with the full list of *A*’s current neighbors. The algorithm returns TRUE if, at the end of the execution, the number of unreached neighbors is greater than a threshold fraction α of all the neighbors of *A*. Otherwise, it returns FALSE. It is apparent that, if we increase/decrease R_{\min} , a neighbor has more/less chances to be removed from the list; thus, it is more/less likely that the

vehicle relays the message. We notice again that, EPIC being a heuristic, Algorithm 2 can declare that: (i) a neighbor of A has received the message, while in fact it has not, or that (ii) a neighbor of A has not received the message, while in fact, the message was received by it. In the former case, there is a possibility that the EPIC node A does not act as the relay, while on the contrary, it is important to relay the message to reach some unreached nodes. In the second case instead, the EPIC node A may relay a message, and this relay is not necessary. If we set a limit on the *EMITTERS* list, the whole procedure's execution has a time complexity that is linear with respect to the number of neighbors of a vehicle node.

Algorithm 2 *EvaluatePositions*

Returns whether or not a vehicle node should re-broadcast a message (*dist* is the Euclidean distance).

procedure *EvaluatePositions*

input:

rcv_messages \leftarrow list of msgs received during timer

neighbor_pos \leftarrow list of positions of my neighbors

initialization:

not_reached \leftarrow neighbor_pos

emitters = \emptyset

for msg **in** rcv_messages **do**

 emitters = emitters \cup msg.EMITTERS

main:

for emitter **in** emitters **do**

for pos **in** neighbor_pos **do**

if $dist(emitter.COORD, pos) < R_{min}$ **then**

 not_reached.remove(pos)

if not_reached.length $> \alpha \cdot neighbor_pos.length$ **then**

return True

else

return False

Figure 2 illustrates an example of the execution of EPIC. Node A is *INFECTED* (EPIC vehicle, in red), waiting for the timer to expire, and receives copies of the message from two emitter nodes (yellow ones in the figure). Once the timer expires, the EPIC node has to decide whether or not to relay. It knows the positions of the emitters, retrieved from the *EMITTERS* field of each received message, and the positions of its neighbors from the LDM (*Neighbor 1*, *Neighbor 2*, and *Neighbor 3* in the figure). The EPIC node gives up relaying the message since all of its neighbors are within R_{min} of at least one emitter node. In Figure 2a, the distance of *Neighbor 3* from the closest emitters is greater than R_{min} ; thus, the EPIC node decides to relay the message. On the other hand, in Figure 2b, all of EPIC's neighbors are within the R'_{min} range ($R'_{min} > R_{min}$) of a previous emitter node; thus, the node gives up relaying.

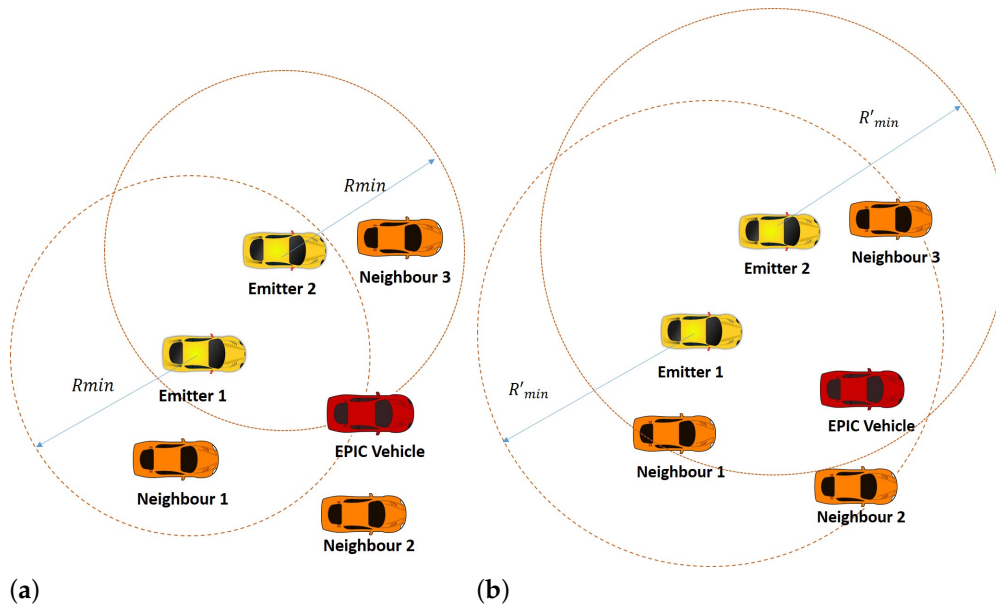


Figure 2. The red vehicle is the node currently running EPIC: (a) In this case, EPIC decides to relay, because one of the three neighbors (Neighbor2 in the figure) is not in the R_{min} range of any emitter; thus, the EPIC vehicle, from its perspective, considers it as not reached by the message. (b) The EPIC vehicle does not relay because the new R_{min} (R'_{min} in the figure) is such that all three of its neighbors are assumed to be reached by the message sent by the other two emitters.

4. Performance Evaluation Setting

To assess the performance of EPIC and compare it to other approaches, we set up simulations of vehicular networks using VEINS [26] and three different scenarios. The considered scenarios refer to three urban maps of Cologne, Luxembourg, and Manhattan in New York. For each of these three scenarios, we defined vehicle flows feeding the maps, with a square RoI having a side of a few km. VEINS allowed an accurate simulation of the micro-mobility of a vehicle through the road map, thanks to all the metadata that describe the road lanes, traffic lights, the right of way, and vehicle routing. Simulation of the radio channel and of the communication stack was taken care of by OMNET++. The considered radio channel was the two-ray ground with the additional attenuation due to obstacles (the description of obstacles was included in the metadata of the urban maps).

For each scenario, we ran simulations where vehicles sent hello messages in the broadcast. Once all vehicles had sent their hello message, we recorded the LDMs of vehicles. In the LDM of vehicle i , we found the list of neighbors of i , i.e., those vehicles from which i received a message successfully. This allowed us to build a connectivity graph of the vehicular network. If vehicle i was reachable from vehicle j , we set $a_{ij} = a_{ji} = 1$. We forced the connectivity matrix to be symmetric, since the radio channel was reciprocal (time-division duplexing). The elements a_{ij} defined the adjacency matrix \mathbf{A} of the connectivity graph. Table 2 shows the main characteristics of the considered graphs.

Figure 3a–c contains an overview of the geographical distribution of the network nodes (in red) and the connections between them (in green) derived by running simulations on the wireless communication via VEINS. We selected these graphs since they provide heterogeneous test settings for EPIC. Luxembourg (Figure 3a) had a high number of nodes, a high number of edges, and a very high average degree. Cologne (Figure 3b) had a low number of nodes, a low number of edges, and a high diameter. New York (Manhattan) (Figure 3c) had a high number of nodes, a relatively high number of edges, and a very low average number of hops between nodes.

Table 2. Metrics of connectivity graphs. In order from top to bottom: number of nodes, number of edges, average node degree, standard deviation of the degree, average distance between any two nodes (measured as the number of hops), standard deviation of the distance, diameter.

Scenario	Luxembourg		Cologne		New York	
	<i>High</i>	<i>Low</i>	<i>High</i>	<i>Low</i>	<i>High</i>	<i>Low</i>
Density						
# of Nodes	790	787	436	220	672	432
# of Edges	17,316	4589	3766	1217	12,462	4447
Avg. degree	43.84	11.66	17.27	11.06	37.09	20.59
Degree std. dev.	24.13	7.34	9.62	5.99	22.51	10.22
Avg. distance	4.07	7.10	8.11	6.25	3.29	3.45
Distance std. dev.	0.82	1.13	1.37	1.35	0.49	0.50
Diameter	10	16	21	15	8	9

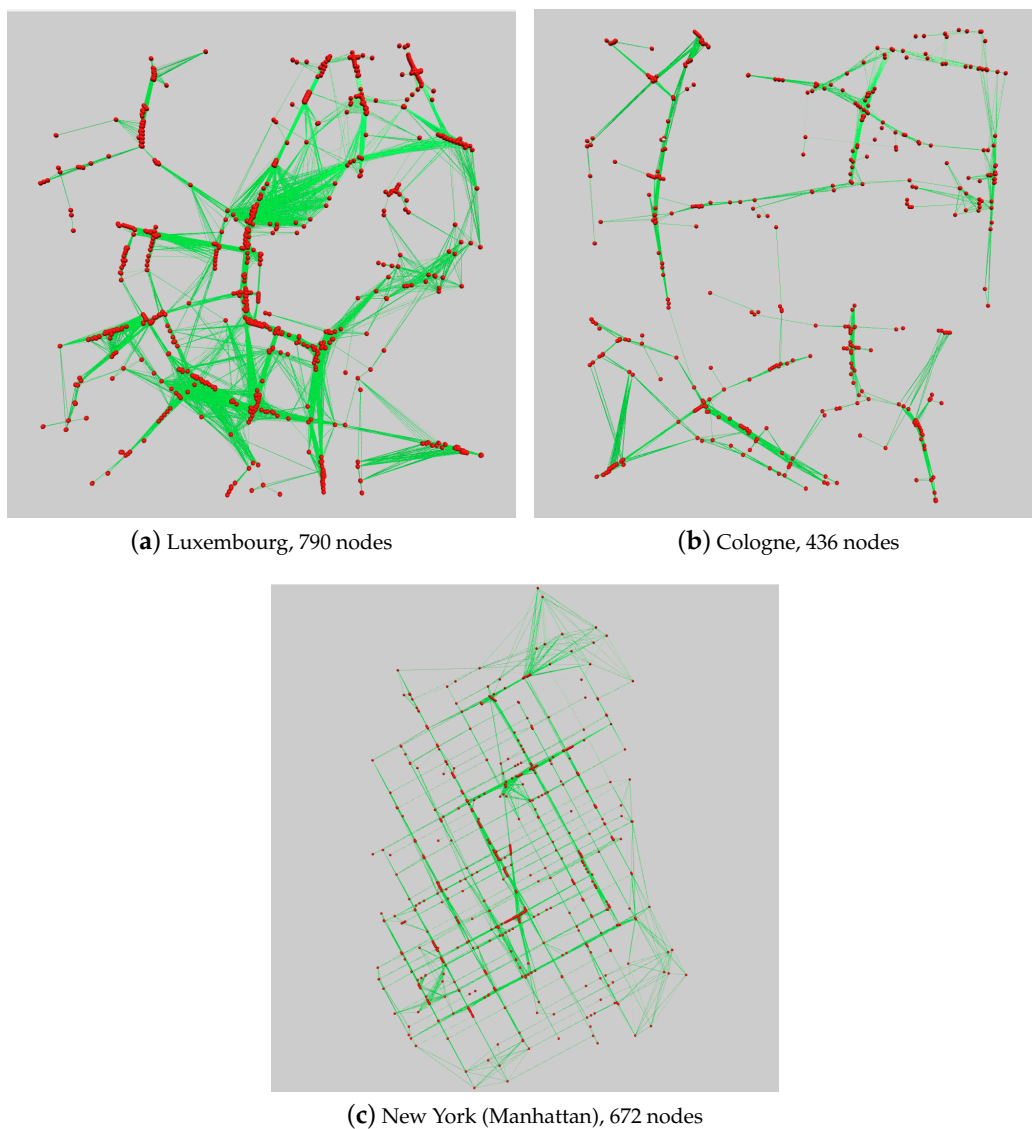


Figure 3. Overview of the network graphs. They will provide heterogeneous test settings for EPIC.

The evaluation of EPIC's performance was carried out on those graphs. The simulation experiments consists of picking an initial node at random and making it start message dissemination. The EPIC algorithm was run in each node of the network. To account for the effect of the underlying radio network, we defined a delay θ required for the transmission of the message, which was the

sum of the overhead implied by lower layers plus the time to transmit the data. Since the packet size variation as it propagated was minor, we set θ equal to a fixed value, namely $\theta = 2.5$ ms. This time stemmed from the following calculation. Accounting for back-off slot count down (in the worst case), frame overhead in IEEE 802.11p amounts to 0.386 ms. If we assumed a node transmitted a payload of 800 bytes at air bit rate 3 Mbit/s (the most robust modulation and coding set of IEEE 802.11p), it turned out that $\theta = 0.386$ ms + $8 \cdot 800$ bytes / 3000 kbit/s ≈ 2.519 ms.

According to the message passing model, if a node started transmitting a message at time t_0 (upon its timer expiry), its neighbors received that message only at time $t_0 + \theta$. This implied that a neighbor whose timer expired after time t_0 , but before time $t_0 + \theta$, would not be inhibited, thus realizing a so-called “spurious forwarding” [27].

As a comparison, we considered the Contention-Based Forwarding (CBF) algorithm of the ETSI GeoNetworking protocol [22]. CBF is a simple forwarding scheme based on timers. Each node receiving a new message starts a timer given by:

$$T_{CBF} = \begin{cases} T_{min} + (T_{max} - T_{min}) \left(1 - \frac{d}{R_{max}}\right) & d < R_{max} \\ T_{min} & d \geq R_{max} \end{cases} \quad (2)$$

where the meaning of the parameters is the same as for EPIC. If the tagged node receives a new copy of the same message while the timer is running, it gets inhibited. It deletes the timers and gives up forwarding it. If instead, the timer expires and no new copy of the message has been received, the message is forwarded. A qualitative representation of this mechanism is reported in Figure 4.

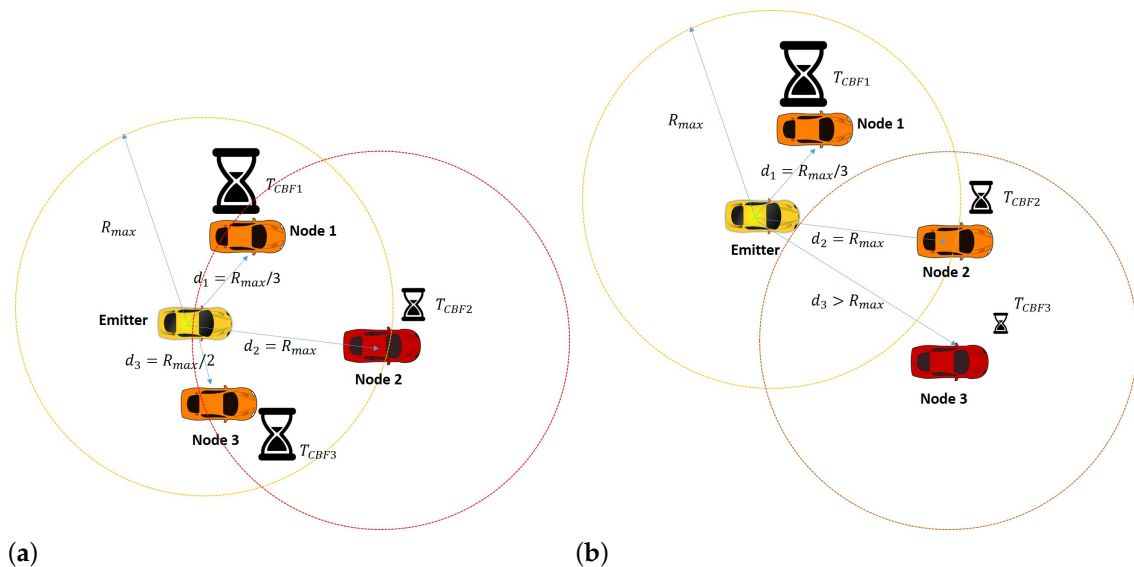


Figure 4. Example of the Contention-Based Forwarding (CBF) behavior where the emitter is the yellow vehicle and its transmission range is R_{max} . Cases: (a) There are three nodes receiving the message. The timers are set in accordance with Equation (2), and the first forwarder is the red vehicle (Node 2). The transmission of Node 2 will be received for the second time by Node 1 and Node 3, who are inhibited from forwarding the message again. (b) In this case, Node 3 receives the original message (being at a distance greater than R_{max}), and its timer is the lowest, so it re-transmits. The transmission by Node 3 inhibits Node 2, but not Node 1, which is no longer in the area of Node 3.

The version of CBF described above is the one specified as an ETSI standard. We introduced a modified version, with a new parameter that could be tuned to improve performance trade-offs, namely the fraction of vehicles reached by a disseminated message against the number of relay vehicles.

The modification affected the inhibition rule. During the timer count-down, a vehicle node was inhibited and canceled message forwarding, if it received at least $K \geq 1$ copies of the same message.

The standard corresponded to the special case $K = 1$. For the sake of clarity, we called the extended algorithm with a general parameter $K > 1$ CBF+ and reserved the name CBF for the standard one ($K = 1$).

5. Performance Results

We present the numerical results of the EPIC algorithm in the three scenarios described in the previous section. The presentation of the numerical results is organized as follows. In Section 5.1, we introduce the performance indicators and list the main parameters of the considered protocols. In Section 5.2, we give an exhaustive evaluation of the message delivery and message delays of EPIC, from which we derive an optimized setting of the EPIC parameters. In Section 5.3, we briefly examine CBF/CBF+'s performance, to derive the optimized setting of the parameter K and select the best configuration of CBF+ to compare with EPIC.

5.1. Metrics

We define the following performance metrics:

- PDR, the Packet Delivery Ratio, i.e., the fraction of the network nodes in the considered graph that were reached by at least one copy of the message. This was also the probability that a node received the disseminated message.
- NR, the Number of Relays, i.e., the number of network nodes that forwarded the message.
- RR, the Ratio of Relays, i.e., the ratio of NR over the number of network nodes, reported in Table 2.
- D, the Delay, that is the time elapsing between the beginning of the dissemination process and the and the last message reception (including duplicates). This was a measure of the time required for the dissemination process to die out. Thanks to the inhibition rule and to the finite number of nodes belonging to a graph, D was bounded above to a finite value.

Each metric was obtained by running several dissemination experiments, each one by selecting a random initial node that triggered message dissemination.

The result for each metric was obtained by taking the average of the obtained values over the different simulations.

Results were produced for three scenarios (Luxembourg, Cologne, and New York) and for two levels of vehicular density (high and low).

In the ensuing analysis, we removed all isolated graph nodes, so that the considered graphs were fully connected. This way, an ideal message dissemination algorithm should reach 100% of the nodes.

5.2. EPIC Performance

Figure 5a,b plot the PDR (dark colored/blue bars) and RR (light colored/orange bars) as percentages vs. R_{min} for the Luxembourg scenario with high and low vehicle densities, respectively.

The horizontal dashed line represents the number of relay nodes estimated according to a heuristic algorithm [28] to compute the cardinality of the minimum connected cover set of a graph, given that it is comprised of a given node (the node that starts the dissemination).

Several remarks are in order.

As for the effect of the parameter R_{min} , the bigger it was, the fewer the number of relay nodes. Correspondingly, also the coverage of the disseminated message exhibited a slight decrease, decreasing from about 99% to about 95%. Sizing R_{min} meant finding a good compromise between coverage (PDR close to 100%) and overhead (number of relay nodes). A sensible value appeared to be $R_{min} \approx 190$ m for high density and $R_{min} \approx 90$ m for low density.

The fact that the achieved number of relay nodes became smaller than the bound predicted by the heuristic algorithm for the minimum connected cover set should not be a surprise. First, it was a heuristic algorithm, not giving the exact minimum. Second, the algorithm found the connected cover

set comprising all nodes belonging to the graph, while EPIC only reached part of the nodes of the graph (even if it was the vast majority).

Finally, the percentage of required relay nodes was quite high, ranging between 30% and 40%. As expected, it was higher for the low density scenario.

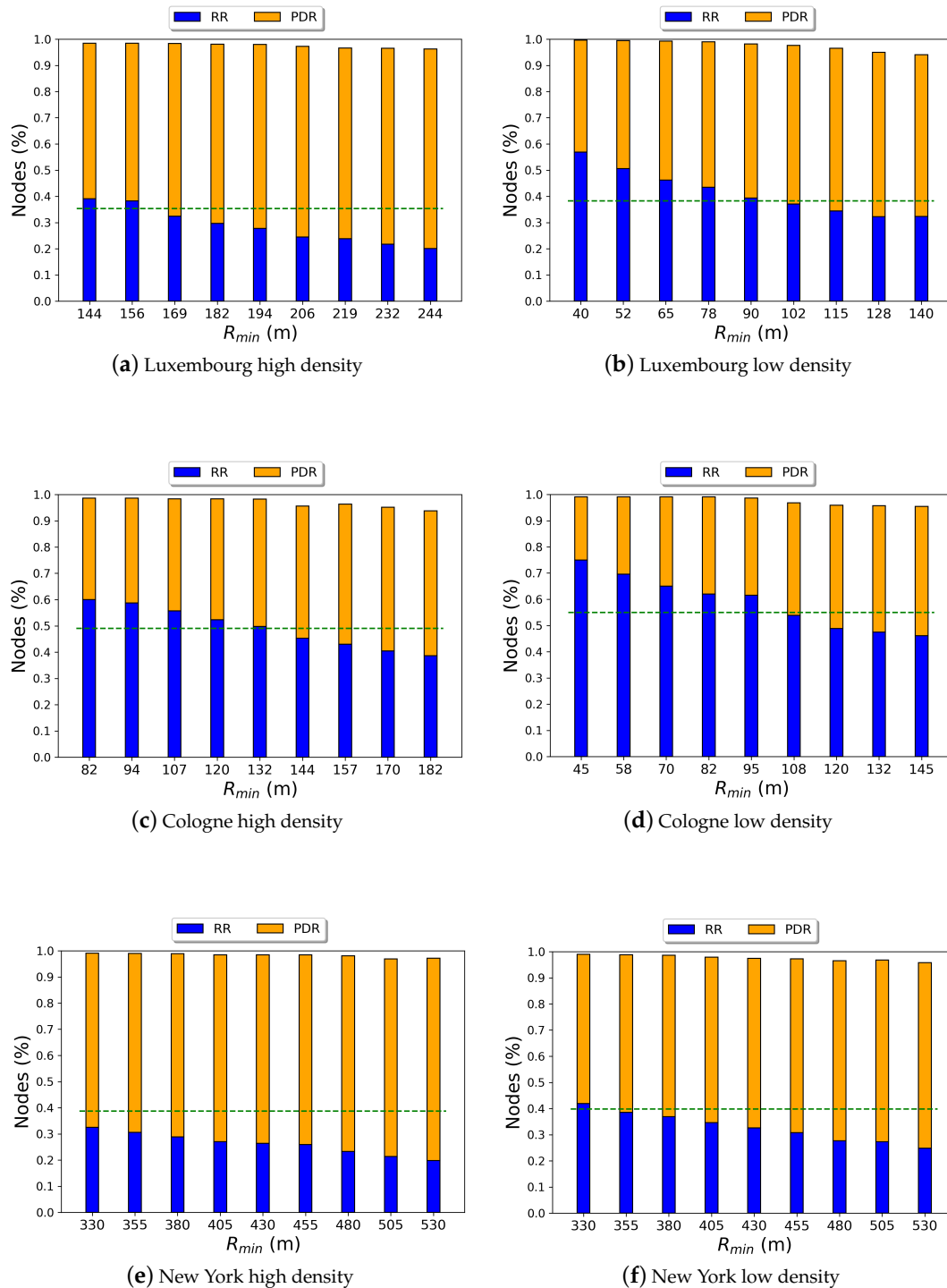


Figure 5. Comparison between network nodes reached by the disseminated message and nodes that broadcast the message, as a function of R_{min} . The overall bar reports the PDR ratio and the blue part is the Ratio of Relays (RR), as described in Section 5.1. The horizontal green line represents the RR using the heuristic connected set cover algorithm described in [28].

Figure 5c,d plot the PDR (dark colored/blue bars) and RR (light colored/orange bars) as percentages vs. R_{min} for the Cologne scenario with high and low vehicle densities, respectively. New York scenario is plotted in Figure 5e,f, according to the same color scheme as for the other scenarios.

Similar comments apply to the Cologne and New York scenarios as for Luxembourg.

The right choice for R_{min} appeared to be $R_{min} \approx 130$ m and $R_{min} \approx 95$ m for the high and low density Cologne scenario. As for New York, we found $R_{min} \approx 450$ m and $R_{min} \approx 380$ m, respectively, for high and low density.

We see that the best choice of the parameter R_{min} depended on the considered urban scenario. A marked difference could be noticed between Luxembourg and Cologne on one side and New York on the other side. In the first case, the right value for low density was slightly below 100 m, while it should be between 100 m and 200 m for high density. In the case of New York, the best R_{min} value was much bigger. The main reason was the regularity of the road pattern of New York (Manhattan) with respect to Luxembourg and Cologne. The average length of open, straight streets is much bigger for New York. The effect of this higher regularity was a low diameter and a low average number of hops between nodes in the connectivity graph, as shown in Table 2, which in turn was also a performance improvement both for the coverage (PDR) and the number of relay nodes (RR).

Figure 6a–f show respectively PDR and RR as a function of the ratio T_{max}/θ , for the three considered scenarios and two levels of vehicular density.

Since R_{min} was set in an optimal way for each scenario, the percentage of covered nodes was quite high. The percentage of relay nodes decreased monotonously with T_{max}/θ , tending to an asymptotic stable value, which was essentially reached for $T_{max} \approx 10 \cdot \theta$.

This set of results gave indications to set a proper maximum timer level T_{max} , once we had an estimate of the amount of time θ required to transmit the packet carrying the message.

Figure 7a,b plot the normalized dissemination delay D/θ as a function of the normalized maximum value of the forwarding timer T_{max}/θ , for all urban scenarios and high and low vehicular density, respectively.

The average dissemination delay increased with T_{max}/θ , as expected. We saw that we needed to set $T_{max}/\theta \approx 10$ to reduce the number of forwarders to the lowest level achievable by EPIC, so as to limit the number of redundant forwarded messages copies.

Going from high to low density, we saw that the performance in the New York scenario did not change, since the considered road topology was highly regular and the communication graph of vehicles was connected in both cases. The irregular road topology of the other two scenarios had contrasting effects. In Luxembourg, we observed a significant increase of dissemination delay when the density was lower. This was due to the reduced effectiveness of the inhibition rule, so that forwarding actions lasted longer to explore the whole vehicle graph. On the contrary, delay in the Cologne scenario become slightly lower with lower density. This correlated with the lower PDR of this last scenario, i.e., dissemination delay was somewhat shorter only because fewer vehicles (in percentage) were reached. Those that were left out were just those that had few connections, i.e., those at which would take more time to arrive.

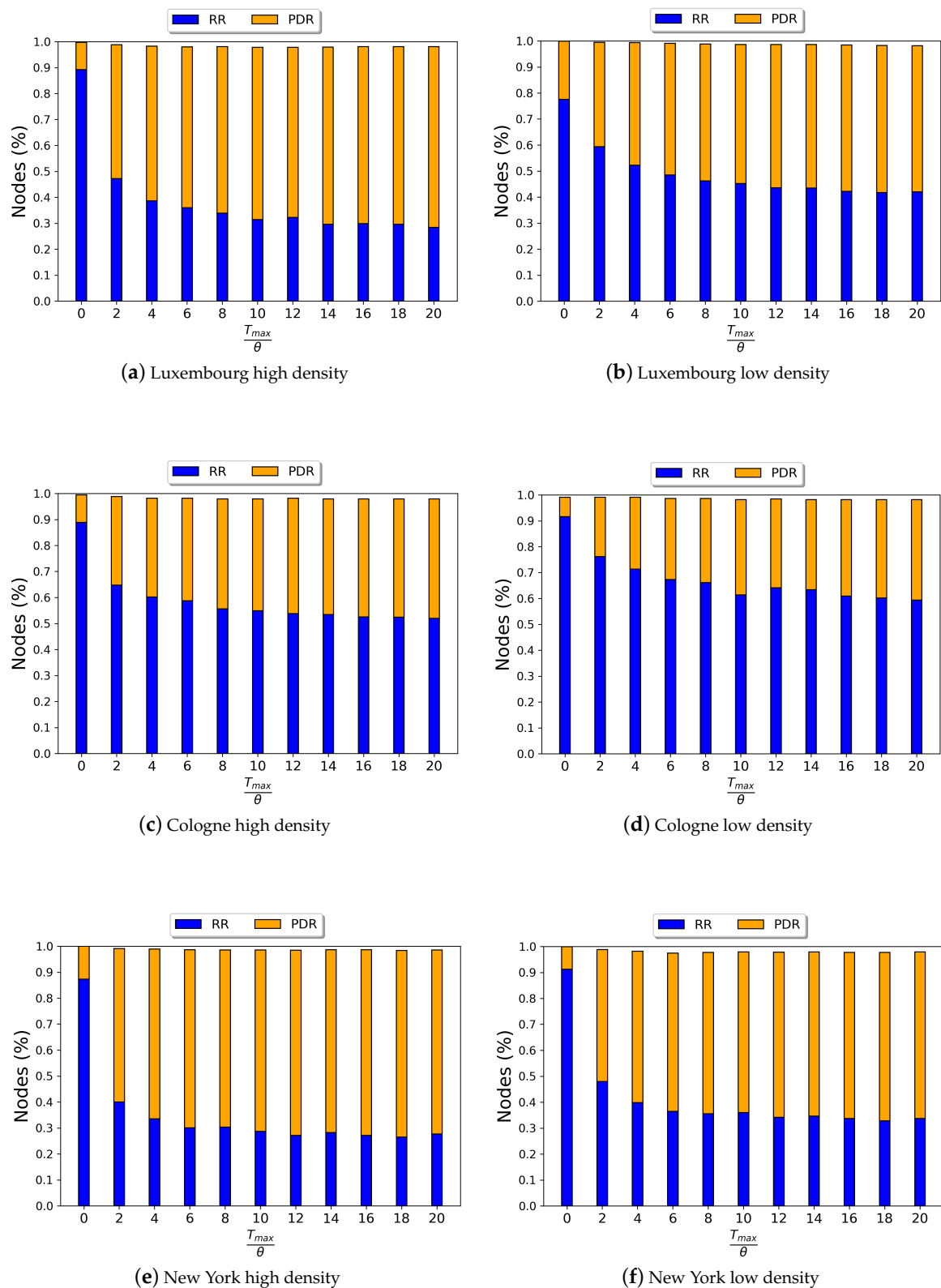


Figure 6. Comparison between network nodes reached by the disseminated message and nodes that broadcast the message by varying T_{max}/θ . If $T_{max} \approx \theta$, network nodes did not have time to receive transmissions from neighbors, and they relayed the message themselves with high probability, making RR reach unnecessary high values. This effect was mitigated if T_{max} was at least approximately ten times θ , achieving high values of PDR with a lower RR.

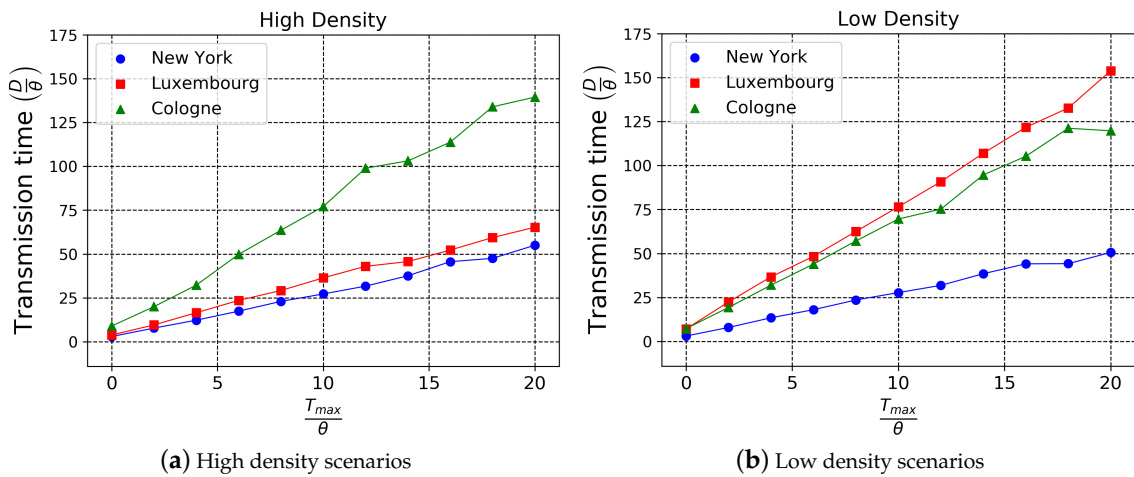


Figure 7. The increase of the transmission time D as a function of T_{max} . If T_{max} is set too low, the overall performance may be degraded, as shown in Figure 6.

5.3. CBF Performance and Parameter Tuning

Figure 8a–c plots the PDR and RR ratios for the Luxembourg, Cologne, and New York scenarios achieved by the CBF+ algorithm vs. the parameter K described in Section 4.

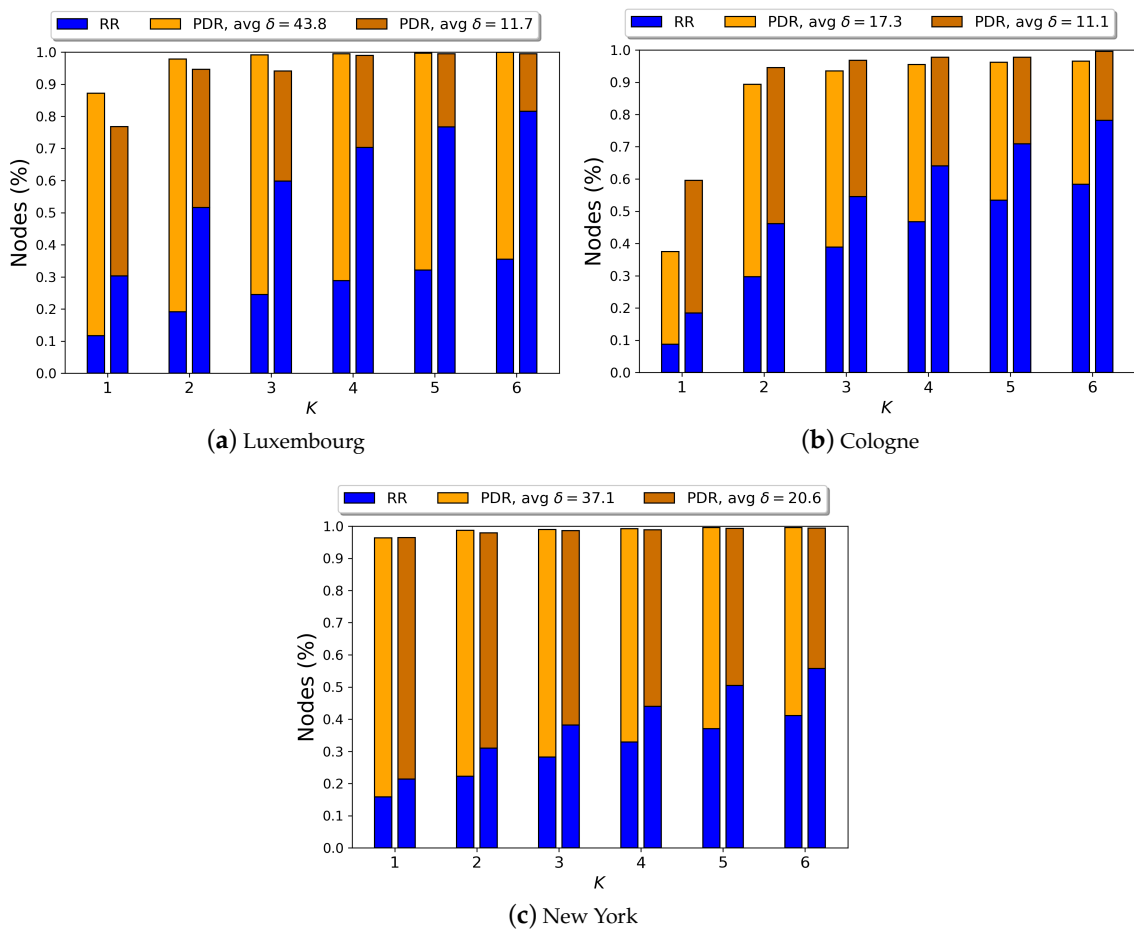


Figure 8. Performance of CBF/CBF+ in terms of PDR and RR as a function of K (standard CBF with $K = 1$). As K increases, the probability that a network node relays a message becomes higher, thus increasing the number of relay nodes and the overall coverage alongside.

The plain CBF algorithm ($K = 1$ in the figures) worked well in the New York scenario (Figure 8c), both in the high density and low density case, achieving high coverage with few relay nodes. As we said previously, the low diameter and low average distance between nodes in the New York graph had the effect of increasing the performance of both PDR and RR.

On the other hand, CBF had poor performance in the Luxembourg (Figure 8a) and Cologne (Figure 8b) scenarios, achieving a ratio of covered nodes of $\approx 87\%$ and $\approx 38\%$ in the high density case and $\approx 86\%$ and $\approx 60\%$ in the low density case, respectively.

To improve CBF’s performance, we could use the CBF+ generalization with $K > 1$. By the definition of K given in Section 4, the bigger K was, the higher the probability that a node relayed a message. This had the effect, in turn, of increasing the number of covered network nodes.

In order to perform the comparison between EPIC and CBF+, we chose an optimal value for K , which meant finding a good compromise of covered nodes and relay nodes for each scenario. We saw that K depended heavily on the scenario and on the density setting. A reasonable choice seemed to be $K = 3$ and $K = 4$ for Luxembourg, $K = 5$ and $K = 5$ for Cologne, and $K = 2$ and $K = 3$ for New York, for the high and low density cases, respectively.

5.4. Comparison of EPIC and CBF/CBF+

In Table 3, we report the values of the parameters that were optimized for each scenario (R_{min}, R_{max}, α for EPIC and K for CBF+). All the simulations were executed by using these parameters and with a packet drop rate of 1%.

Table 3. Parameters used during the simulations.

Scenario	Luxembourg		Cologne		New York	
	High	Low	High	Low	High	Low
R_{min}	194 m	90 m	132 m	95 m	430 m	430 m
R_{max}	500 m	500 m	500 m	500 m	1000 m	1000 m
α	5%	0%	5%	0%	10%	10%
K	3	4	5	5	2	3

In Tables 4–6, we report the simulation results, using both EPIC and CBF+ as the dissemination algorithm, for the scenarios of New York Luxembourg, and Cologne, respectively. In each table, we report the number of relay nodes, the ratio of relay nodes, the transmission delay, and the PDR.

Table 4. Results obtained for the New York scenario. NR, Number of Relays.

Algorithm	CBF+	EPIC	CBF+	EPIC
Density	High		Low	
NR	150	176	171	139
RR	22.32%	26.19%	39.58%	32.17%
D	335 ms	247 ms	301 ms	252 ms
PDR	98.14%	98.59%	98.61%	97.69%

Table 5. Results obtained for the Luxembourg scenario.

Algorithm	CBF+	EPIC	CBF+	EPIC
Density	High		Low	
NR	207	228	547	320
RR	26.20%	28.86%	69.50%	40.66%
D	403 ms	313 ms	791 ms	736 ms
PDR	99.05%	98.23%	98.98%	98.28%

Table 6. Results obtained for the Cologne scenario.

Algorithm	CBF+	EPIC	CBF+	EPIC
Density	<i>High</i>		<i>Low</i>	
NR	202	215	155	129
RR	51.95%	49.31%	70.45%	58.64%
D	692 ms	765 ms	627 ms	621 ms
PDR	96.01%	98.17%	97.73%	98.18%

As for New York (Table 4), we noticed that both EPIC and CBF+ achieved a satisfactory ratio of covered network nodes, with a PDR close to 100%. As for RR, EPIC used about 4% more relay nodes than CBF+ in the high density scenario, while in the low density case, EPIC used about 7% less than CBF+. Both algorithms worked well in the New York scenario, mainly due to its low diameter and low average number of hops between network nodes.

For Luxembourg (Table 5), we observed similar values of PDR and RR across the high density case. Instead, for the low density case, the relay ratio dropped from 69.50% for CBF+ to 40.66% for EPIC, a relative decrease of more than 40%.

This high number of relay nodes of CBF+ was due to the high value of K , which was set in order to have a PDR ratio comparable to EPIC.

In EPIC, the inhibition rule indicated to a node whether or not to act as a relay based on the geographical coordinates of the previous broadcasts. This was not possible using CBF+, which simply counted the number of received copies of the same message, independently of from which the position they were broadcast.

The power of the inhibition rule designed in EPIC was measured by achieving a high coverage while maintaining a low number of relay nodes.

As for the Cologne scenario (Table 6), EPIC and CBF+ achieved in the high density case similar values of covered and relay nodes, with EPIC showing slightly higher coverage (98.17% vs. 96.01%) and a slightly lower relay ratio (49.31% vs. 51.95%). In the low density scenario, the RR ratio went from 70.45% for CBF+ to 58.64% for EPIC, a relative decrease of more than 15%.

As for the dissemination delay, we observed that in most of the scenarios, EPIC needed less time to disseminate the message. While only for Cologne, high density CBF+ was faster (692 ms vs. 765 ms), for Luxembourg, high density EPIC was $\approx 25\%$ faster than CBF+ and more than 15% faster for New York, both high and low density.

6. Conclusions

We defined an epidemic algorithm, EPIC, for message dissemination in vehicular networks with direct V2V communications. The EPIC algorithm was completely distributed and executed autonomously by each vehicle node, based on proper parameter settings and events collected at the network interface. We provided a simulation-based performance analysis to tune EPIC's parameters. Simulations were based on accurate modeling of the radio path, physical layer, and vehicular environment, obtained by means of VEINS and three different urban scenarios.

We compared EPIC with the ETSI standard algorithm for message dissemination, Contention-Based Forwarding (CBF), and a heuristic performance bound on the number of relay vehicle nodes, based on the minimum connected cover set cardinality on the vehicle connection graph.

Performance analysis showed that EPIC offered a good performance trade-off between the coverage of message dissemination (which reached more than 98% of nodes in high density scenarios with the number of relay nodes below 30–40%, depending on the analyzed city), overhead due to multiple forwarding nodes/message copies, and dissemination delay. Extension of this work may address adaptive self-tuning of protocol parameter, to adapt to the specific vehicular scenario (e.g., vehicle density, pattern of roads).

Author Contributions: Conceptualization, F.C. and A.B.; methodology, A.B.; software, P.S.; validation, P.S.; formal analysis, A.B.; investigation, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Valerio Fenni and Andrea Sorrentino for their implementation of the code in [28] to derive the minimum connected set covered to measure the optimal number of relay nodes in our vehicular scenarios.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Toh, C.K. *Wireless ATM and Ad-Hoc Networks: Protocols and Architectures*; Kluwer Academic Publishers: Norwell, MA, USA, 1996.
2. Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In Proceedings of the VTC Spring 2008—IEEE Vehicular Technology Conference, Singapore, 11–14 May 2008.
3. Kermack, W.O.; McKendrick, A.G. A contribution to the mathematical theory of epidemics. *Proc. R. Soc. Lond. Ser. A* **1927**, *115*, 700–721.
4. Aparecido, L. Data dissemination in vehicular networks: Challenges, solutions, and future perspectives. In Proceedings of the 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 27–29 July 2015; pp. 1–5, doi:10.1109/NTMS.2015.7266482. [[CrossRef](#)]
5. Wisitpongphan, N.; Tonguz, O.K.; Parikh, J.S.; Mudalige, P.; Bai, F.; Sadekar, V. Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wirel. Commun.* **2007**, *14*, 84–94. [[CrossRef](#)]
6. Cuomo, F.; Rubin, I.; Baiocchi, A.; Salvo, P. Enhanced VANET Broadcast Throughput Capacity via a Dynamic Backbone Architecture. *Ad Hoc Netw.* **2014**, *21*, 42–59, doi:10.1016/j.adhoc.2014.04.008. [[CrossRef](#)]
7. Turcanu, I.; Salvo, P.; Baiocchi, A.; Cuomo, F.; Engel, T. A multi-hop broadcast wave approach for floating car data collection in vehicular networks, *Veh. Commun.* **2020**, *24*, 100232, doi:10.1016/j.vehcom.2020.100232. [[CrossRef](#)]
8. Musolesi, M.; Mascolo, C. Controlled Epidemic-Style Dissemination Middleware for Mobile Ad Hoc Networks. In Proceedings of the 2006 3rd Annual International Conference on Mobile and Ubiquitous Systems—Workshops, San Jose, CA, USA, 17–21 July 2006; pp. 1–9, doi:10.1109/MOBIQW.2006.361736. [[CrossRef](#)]
9. Eugster, P.; Guerraoui, R.; Kermarrec, A.M.; Massoulié, L. Epidemic Information Dissemination in Distributed Systems. *IEEE Comput.* **2004**, *37*, 60–67, doi:10.1109/MC.2004.1297243. [[CrossRef](#)]
10. Nekovee, M. Epidemic algorithms for reliable and efficient information dissemination in vehicular. *IET Intell. Transp. Syst.* **2009**, *3*, 104–110, doi:10.1049/iet-its:20070061. [[CrossRef](#)]
11. Ganesan, D.; Krishnamachari, B.; Woo, A.; Culler, D.; Estrin, D.; Wicker, S. *An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks*; Technical Report IRB-TR-02-003; Intel Corporation: Santa Clara, CA, USA, 2002.
12. Byun, H.; So, J. Node Scheduling Control Inspired by Epidemic Theory for Data Dissemination in Wireless Sensor-Actuator Networks With Delay Constraints. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 1794–1807, doi:10.1109/TWC.2015.2496596. [[CrossRef](#)]
13. Chen, P.; Cheng, S.; Chen, K. Optimal Control of Epidemic Information Dissemination Over Networks. *IEEE Trans. Cybern.* **2014**, *44*, 2316–2328, doi:10.1109/TCYB.2014.2306781. [[CrossRef](#)] [[PubMed](#)]
14. Ajelli, M.; Lo Cigno, R.; Montessoro, A. Modeling Botnets and Epidemic Malware. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–5, doi:10.1109/ICC.2010.5502265. [[CrossRef](#)]
15. Salvo, P.; Cuomo, F.; Baiocchi, A.; Rubin, I. Probabilistic relay selection in timer-based dissemination protocols for VANETs. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 2725–2730, doi:10.1109/ICC.2014.6883736. [[CrossRef](#)]
16. Saeed, T.; Mylonas, Y.; Pitsillides, A.; Papadopoulou, V.; Lestas, M. Modeling Probabilistic Flooding in VANETs for Optimal Rebroadcast Probabilities. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 556–570, doi:10.1109/TITS.2018.2828413. [[CrossRef](#)]

17. Zhang, X.; Yan, L.; Zhang, H.; Sung, D. A Concurrent Transmission Based Broadcast Scheme for Urban VANETs. *IEEE Trans. Mob. Comput.* **2019**, *18*, 1–12, doi:10.1109/TMC.2018.2827930. [[CrossRef](#)]
18. Benabdallah, F.; Hamza, A.; Becherif, M. On the use of non-orthogonal multiple access for V2V message dissemination. *IET Intell. Transp. Syst.* **2019**, *13*, 1125–1129, doi:10.1049/iet-its.2018.5378. [[CrossRef](#)]
19. Benrhaïem, W.; Hafid, A.; Sahu, P.K. Reliable Emergency Message Dissemination Scheme for Urban Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1154–1166, doi:10.1109/TITS.2019.2902850. [[CrossRef](#)]
20. Chaqfeh, M.; El-Sayed, H.; Lakas, A. Efficient Data Dissemination for Urban Vehicular Environments. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1226–1236, doi:10.1109/TITS.2018.2850068. [[CrossRef](#)]
21. Spadaccino, P.; Conti, P.; Boninsegna, E.; Cuomo, F.; Baiocchi, A. EPIC: An Epidemic Based Dissemination Algorithm for VANETs. In *Proceedings of the 1st ACM MobiHoc Workshop on Technologies, MOdels, and Protocols for Cooperative Connected Cars*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–6, doi:10.1145/3331054.3331546. [[CrossRef](#)]
22. ETSI EN 302 636-4-1 v1.2.1. *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical Addressing and Forwarding for Point-to-Point and Point-to-Multipoint Communications; Sub-Part 1: Media-Independent Functionality*; ETSI: Valbonne, France, 2014.
23. Chitra, M.; Siva Sathya, S. SEIR epidemic spreading model to suppress broadcast storm in vehicular ad hoc networks. *Int. J. Veh. Saf.* **2017**, *9*, 228–252, doi:10.1504/IJVS.2017.085204. [[CrossRef](#)]
24. ETSI EN 302 895 v1.1.1; *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*. 2014. Available online: <http://www.etsi.org> (accessed on 26 February 2020).
25. Shah, S.; Ahmed, E.; Xia, F.; Karim, A.; Shiraz, M.; Md. Noor, R. Adaptive Beaconing Approaches for Vehicular Ad Hoc Networks: A Survey. *IEEE Syst. J.* **2018**, *12*, 1263–1277, doi:10.1109/JSYST.2016.2573680. [[CrossRef](#)]
26. Hagenauer, F.; Dressler, F.; Sommer, C. A Simulator for Heterogeneous Vehicular Networks. In *Proceedings of the 6th IEEE Vehicular Networking Conference (VNC 2014)*, Paderborn, Germany, 3–5 December 2014; pp. 185–186, doi:10.1109/VNC.2014.7013339. [[CrossRef](#)]
27. Baiocchi, A.; Salvo, P.; Cuomo, F.; Rubin, I. Understanding Spurious Message Forwarding in VANET Beaconless Dissemination Protocols: An Analytical Approach. *IEEE Trans. Veh. Technol.* **2016**, *65*, 2243–2258, doi:10.1109/TVT.2015.2422753. [[CrossRef](#)]
28. Ren, W.; Zhao, Q. A note on ‘Algorithms for connected set cover problem and fault-tolerant connected set cover problem’. *Theor. Comput. Sci.* **2011**, *412*, 6451–6454, doi:10.1016/j.tcs.2011.07.008. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).