Analysis and Simulation of a Content Delivery Application for Vehicular Wireless Networks [1]

Marco Fiore Claudio Casetti Carla-Fabiana Chiasserini
Michele Garetto

Dipartimento di Elettronica, Politecnico di Torino
Corso Duca degli Abruzzi 24, I-10129 Torino (Italy)
e-mail: `<lastname>@tlc.polito.it`

Abstract

We propose and analyze an information-sharing application for wireless intervehicular networks, called Infoshare. Infoshare leverages the broadcast nature of the wireless medium to achieve maximum spreading of information queries among vehicles, while a smart caching policy limits the overhead resulting from useless queries and duplicated replies. We first evaluate the performance of Infoshare through the simulator ns2, highlighting the impact of various system parameters on the information spreading dynamics. Then, we develop an analytical model of the information exchange dynamics, which allows us to identify the system parameters that guarantee a sustainable information exchange on the network scenario under study.

Key words: Wireless intervehicular networks; Content delivery; Performance Evaluation

1 Introduction

It is now commonly acknowledged that cars in the not-too-distant future will more and more resemble a communication hub, sporting arrays of GPS navigators, DVD players, videogame consoles and sleek LCD screens flashing the latest traffic alerts, breaking news or local sightseeing information. Of course, the exchange of audio and video clips through an InterVehicular Network (IVN) would require that vehicles be equipped with a high-bit-rate radio interface and gigabytes of storage. However, although some already envision peer-to-peer networks on wheels, it is quite likely that the first cooperative networked environment among moving vehicles will feature short-lived data exchanges, mostly about traffic warnings or maybe adverts for a local store running the latest seasonal sale. In such highly impersonal context, most pieces of information are likely to be of general use, and therefore a sensible dissemination and caching policy would be desirable.

[1] Partially supported through the European NoE EuroNGI

In this paper we focus on such an environment: a few access points, or gateway nodes, along the road and far from each other, feed passing cars the information they require. We envision a set of information categories that users may be interested in, and that they "pull" from the network. To maximize the chance of getting fresh information, we assume that vehicles are capable of cooperating to disseminate the information that was pulled from gateways, in order to reach farther vehicles by forming an ad hoc network. Such cooperation is aided by on-board caches that keep track of what nearby users have exchanged, and that feed the required information upon a query from a user out of range from the gateway node. We develop an information sharing application, based on such a "pull" paradigm, named *Infoshare*, to test the effects of the various system parameters on the spreading dynamics of the shared information. By investigating its performance through simulation and analysis, we try to answer the following questions: (i) Which are the parameters that significantly affect the performance of content delivery in IVNs? (ii) Aiming at modeling content delivery in IVNs, which aspects can be neglected and which simplifications would be acceptable? (iii) How should the system parameters be set to guarantee a sustainable information exchange?

The present work, which extends (9), fits into a flurry of recent efforts by researchers focusing their attention on IVNs. A seminal work in this field is (1), where small, high-data-rate communication islands over a highway systems (Infostations) are proposed as an alternative to cellular coverage. The performance of Infostations are further investigated in (16; 17). The technical challenges of delivering multimedia and safety information to cars forming an ad hoc network are outlined in (11). In (10) an in-vehicle entertainment system is considered, through which users can download audio and video traffic. The problem addressed there concerns the availability of the information and how to predict such availability. A cooperative strategy for content delivery and sharing is proposed in (7; 13). The latter, named SPAWN, addresses several issues: peer discovery, content selection, and content discovery. Peer discovery uses a centralized method as well as a distributed approach. The distributed technique leverages the broadcast nature of the wireless medium that allows nodes to overhear information about the content availability at neighbors. Content selection is determined by a proximity-driven piece selection strategy, where proximity estimation is based on hop-count, while content discovery is implemented by making peers communicate which information they own to all their neighbors. A cooperative approach to provide data services in MANETs is proposed in (5), where a context-aware group communication middleware is used to select collaborating partners and schedule messages. Three cooperative caching schemes explicitly designed for MANETs are explored in (3), highlighting the challenges of caching strategies for wireless users, compared to traditional wired operations. With regards to routing protocols, in our work we do not assume any specific scheme; however, we would like to mention that several on-demand routing protocols for MANETs have been proposed: they distribute queries to network nodes and establish paths for unicast traffic (e.g., (12)). Also, several solutions have been proposed to reduce the routing overhead of on-demand protocols. For instance, Location Aided Routing (2) and Query Localization (6) limit the query flood by decreasing the number of

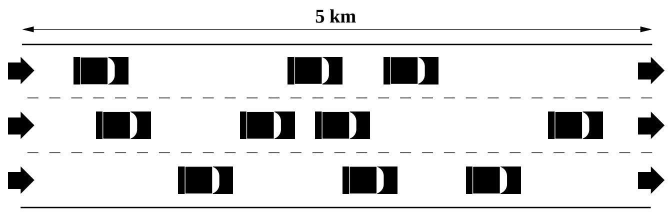

Fig. 1. Reference network scenario

nodes receiving route queries. The mechanism in (2) restricts the flooding of the query using Global Positioning System (GPS), while in (6) route requests are forwarded only in those areas where old paths existed.

The remainder of this work is organized as follows. In Section 2 we lay out the scenario and the assumptions regarding the communication technology; a detailed description of the Infoshare application can be found in Section 3. Section 4 describes the simulation set-up, while simulation results are presented and discussed in Section 5. Section 6 introduces the analytical model of the information exchange dynamics; the model is validated and exploited in Section 7. Section 8 draws some conclusions.

2 Network scenario

We consider an IVN deployed along a unidirectional, straight road, such as a single- or a multi-lane highway. Vehicles are randomly spread along the road, and we assume that those traveling on the same lane cruise at a constant speed. An example of the network scenario is depicted in Figure 1.

Each vehicle is equipped with a radio interface and a data cache. Users on a vehicle wish to access information made available by fixed gateway nodes, located on the roadside, that are connected to the Internet and are broadcasting along the road. We assume that N distinct pieces of information are available and may be requested by the users. Connectivity between vehicles and gateway nodes is, however, spotty and cooperation among vehicles is highly desirable. We focus on the road segment between two gateways nodes, and investigate the performance of the IVN. We assume that a gateway node is placed at the beginning of the road and that vehicles entering the road segment have managed to download some information from the gateway, i.e., they already carry some pieces of cached information. Then, we consider that all vehicles are equipped with an information-sharing application, named *Infoshare*, that allows vehicles to efficiently exchange data.

We select IEEE 802.11 as wireless communication technology since several experiments (8; 14; 15; 4) have shown that it is suitable for both inter-vehicular and infrastructure-to-vehicle communications. We assume that radio transmissions take place at 2 Mb/s, and set the node radio range to 100 m. These values are consistent with real-world measurements for vehicular networks, performed through 802.11b cards on board of vehicles moving over highway scenarios at speeds comparable to those used in our simulations (14; 15).

3 Information sharing application

Infoshare is a lightweight application intended for a context in which multiple different small pieces of information with fast time dynamics are shared by vehicles moving along the road. The general behavior of the application is the following: a set of N different pieces of information is available for sharing, each type of information being identified by a unique *id*. A vehicle queries other vehicles for information pieces it does not have. Queries are broadcast by the source vehicle and relayed by receiving nodes, so that the request is propagated in a multihop fashion until a vehicle carrying the desired data is reached. A flow chart detailing these operations is depicted in Figure 2. Once found, the information is returned to the query source through an application-driven, unicast path. Upon reception of the message containing the requested data, the query source vehicle caches the information for a certain time, after which the data is dropped and may be requested again [2]. Operations that follow the reception of an information message at a vehicle (either a source or a relay node) are illustrated in Figure 3. In the following paragraphs a detailed description of the Infoshare application functioning is provided.

3.1 Query generation

A *query* message, carries the following information:

- **source address** (*srcAddr*): the application-level address of the node that generated the query
- **information ID** (*id*): the identifier of the requested piece of information
- **sequence number** (*seqNo*): the overall number of requests performed by this source. This value is incremented at the generation of a new query by the source vehicle
- **next hop address** (*nextAddr*): the address of the node that physically transmitted the query message. When the query is generated, this field matches the source address. On the other hand, since a multihop mechanism is employed by the application to spread the query, this address becomes different from the source one as the message is forwarded by intermediate nodes acting as relays
- **time to live** (TTL): the remaining number of hops allowed for the current message. This field is decremented at each query forward by relaying vehicles.

Note that the $\{srcAddr, id, seqNo\}$ triplet identifies the query in a unique way within the whole ad hoc network. Thus, it is possible for relay nodes to correctly forward new requests and discard those which are duplicate (i.e., already relayed) or out-of-date (i.e., queries for a piece of information for which a query from the same source and with an higher sequence number was

[2] The main reason why we chose to drop the cached data after a certain time is to avoid the problem of stale information at the vehicles

already received and relayed). Also, the name of the *nextAddr* field may sound misleading, since it actually refers to the previous node in the forward path; however, it plays a crucial role in tracing the steps back to the query source, as discussed later, hence the “next hop” tag.

3.2 Query spreading

The query spreading is performed employing broadcast transmissions, and exploits a *query list* structure at each node to manage the relaying process.

The *query list* keeps track of the received queries, each list element containing the following fields:

- **source address**: the application-level address of the node that generated the query, obtained from the *srcAddr* field of the received query
- **information ID**: the identifier of the requested piece of information, obtained from the *id* field of the received query
- **sequence number**: the overall number of requests performed by this source, obtained from the *seqNo* field of the received query
- **next hop address**: the address of the node that physically transmitted the query message. This is the address of the node from which the query was received, and is obtained from the *nextAddr* field of the received query
- **status**: the status of the request, either PENDING, i.e., waiting to be solved, or SOLVED, that is, already fulfilled.

When a vehicle requests one piece of information, it broadcasts the new query message, but also adds an entry to its own query list. The status of the query added to the list is obviously set to PENDING [3]. Note that, to avoid an uncontrolled growth of the list dimension, entries can be deleted after an expiration threshold, which we set to 20 s in our tests.

As illustrated in Figure 2, each node receiving the query message first searches its query list for the same query. If the request is not already present in the list, the message is discarded; otherwise a new query list entry is created and the status of such query list element is set to PENDING. Next, the node checks whether it owns the piece of information the source node is requesting. If the application does not have the requested information in its cache, and the TTL field in the query message is greater than zero, it acts as relay. However, before retransmitting the query, the node waits for a *query lag* interval of time, at the end of which it checks the query status: the query is only forwarded if the status is still PENDING. The introduction of the delay when relaying query messages limits query flooding, as explained later in this section. The *query lag* value must be consistent with the application time dynamics, and, in our study, we have chosen a random time value uniformly distributed between 13 and 17 ms. Such procedure is then repeated at each hop, until a vehicle storing the requested information is found, or the TTL reaches zero.

[3] Clearly, the same application can have multiple PENDING queries at the same time

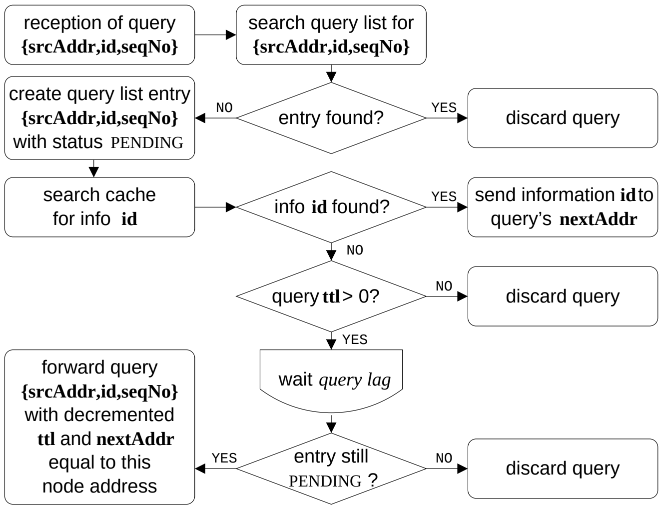

Fig. 2. Flow chart of operations following the reception of a query

We make the following remarks. (i) The next hop address fields stored at each forwarding application build up *a multihop unicast "return path"* to the source of the query. (ii) The presence of a query list at each application and the control on the query sequence number prevent useless duplication of queries. (iii) In spite of the query lag mechanism, query duplicates may still be generated in the network since *all* nodes receiving the query and not owning the requested information act as relays. Such redundancy has a twofold motivation. First, query spreading only relies on broadcast messages, which are by their nature unreliable[4] : the duplication of queries can be exploited to balance the unreliability of broadcast transmission. Secondly, we want to keep the application as simple as possible, while limiting the number of forwarding nodes would imply a relay selection which is usually based on the use of localization systems (e.g., GPS) and complex computations.

3.3 Information retrieval and transmission

When the query is received by a node owning the desired piece of information, the application at such node immediately sends a *unicast* message containing the information to the vehicle it received the request from, which is then charged with the task of forwarding it towards the query source.

Typically, in mobile ad hoc networks unicast transmissions rely on a routing protocol. Infoshare instead does not require multihop capability at the network layer: the aforementioned "return path" is used at the application layer to transmit the information back to the vehicle that generated the request.

More specifically, the application header attached to the information payload replicates the structure of the query message, containing the $\{srcAddr, id, seqNo\}$

[4] Note that, according to the IEEE 802.11 MAC layer specifications, broadcast messages are not acknowledged by the receivers, and they are never retransmitted by the sender

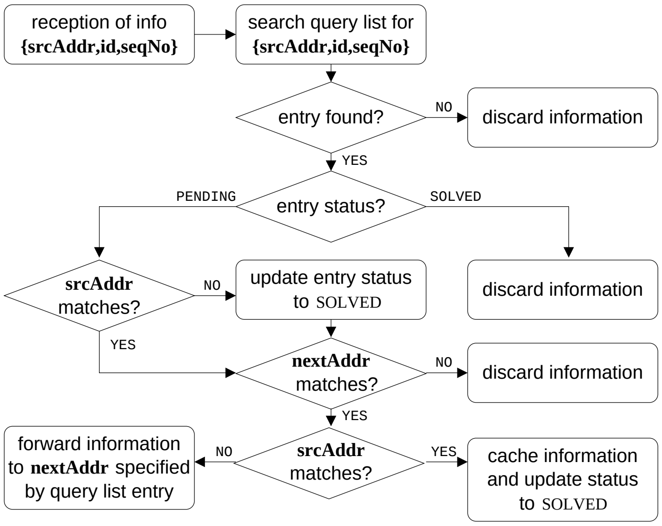

Fig. 3. Flow chart of operations following the reception of an information message

triplet which enables the identification of the query the message replies to. Thus, a node receiving an information message can look up its query list for the corresponding query, update the status of such entry to SOLVED [5], and retrieve the *nextAddr*, i.e., the address of the next hop in the application layer "return path". In other words, the application can indicate the next hop node to be used to feed the information back to the requesting vehicle, so that transmissions at the network layer are always single-hop ones. This mechanism is then iterated, as the next hop vehicle that receives the information message sets the status of the corresponding entry in the query list to SOLVED, and forwards in its turn the information to *its* next hop, until the source of the query is reached. The operations described above are detailed in Figure 3.

Note that, firstly, unicast reply messages via the return path ensure a reliable transmission of the information back to the source of the query, thanks to retransmissions at MAC layer. Secondly, this application-driven routing feature of Infoshare can be leveraged jointly with the broadcast nature of wireless communications to improve the system performance. As a matter of fact, replies containing the information are unicast, which improves their reliability, but they can be sensed by other nodes within range, due to the nature of the wireless medium. Exploiting the *promiscuous mode*, which is implemented in 802.11 wireless LAN cards (18), also applications located at vehicles other than the intended destination can receive the information without penalty in terms of traffic load on the channel. We now examine drawbacks and advantages of this solution.

The drawbacks are as follows:

- a slightly increased message size with respect to performing multihop routing at network layer, since the additional *nextAddr* field in the application

[5] Recall that, upon receiving a new query, a node creates an entry in the query list, no matter whether the node has the requested information cached or not. This allows the application to recognize duplicated queries, thus preventing generation of multiple answers to the same request

header is needed to identify the intended recipient of the information, and
- additional processing effort by all applications located at nodes within transmission range, which are not the actual recipients of the message and must therefore discard it.

The advantages are twofold:

- there is no need for routing at the network layer, which could be very expensive in terms of both computational resources and channel overhead in presence of dense, highly dynamic ad hoc networks, like IVNs;
- it is possible to exploit the fact that nodes not in the "return path" can receive the information message as well, to further reduce the number of useless transmissions. By listening to the channel, such nodes can learn that the data requested by a particular query is on its way back to the query source and set to SOLVED the status of that query entry in their list. Thus, at nodes waiting to forward the query, when the query lag expires, the entry in the query list is found not to be PENDING anymore and the query is not rebroadcast; in practice, the query is not propagated any further than the first node having the requested information in cache, in each direction.

3.4 Information caching

When the node that originated the query receives the information message, it updates the relative query list entry to a SOLVED status, and caches the data. After some time, the information is discarded and can be requested again with a new query. In our simulation we assumed the cache to be large enough to store all the N pieces of information, thus no information is discarded due to a full cache.

4 Simulation scenario

Simulations were conducted using the network simulator ns2. The Infoshare application was implemented in ns2, and a road mobility model allowing the use of several statistical distributions for vehicle speed and interarrival time was developed.

The simulation scenario consists in a unidirectional straight road (e.g., a highway lane) with vehicles entering the road according to a random distribution and moving at constant speed up to the end of the road, where vehicles are removed. The road length is set to 5 km. An example of the simulated scenario is illustrated in Figure 1. Each new vehicle entering the road holds an initial number of information pieces, which is the same for all nodes and varies across the simulation experiments. Requests for non-cached pieces of information are performed by each node according to i.i.d. Poisson processes with rate $\lambda = 0.006$, while a cached piece of information is dropped after a random time which is exponentially distributed with mean $1/\mu$. The TTL parameter in the query messages is set to infinity.

At the transport layer, we adopt the simple *Message Passing* agent provided by the ns2 simulator. This is equivalent to the UDP protocol, since it does not perform any packet retransmission or congestion control. Moreover, the Message Passing agent does not add any header to the data it receives from the application. We set the application header size to 32 bytes, and we considered it to include the transport layer 8-byte (UDP-like) header. The payload of information messages is set to 480 bytes.

At the MAC layer the IEEE 802.11 scheme is employed: the RTS-CTS handshaking is used for the transmission of information messages, while, since queries are broadcast messages, no RTS-CTS is used for them and they are never retransmitted. The data transmission rate and the radio range are set to 2 Mbps and 100 m, respectively.

All simulations have a duration of 2000 s. However steady-state statistics (such as the fraction of solved queries) are collected only from 500 s on, to avoid transient effects.

5 Simulation results

The general goal of our simulation experiments is to understand how different mobility conditions affect the information sharing process and therefore the capability of the IVN to support an application characterized by fast dynamics such as Infoshare. In particular, we are interested in the *information sustainability*, i.e., the ability of the system as a whole to propagate and maintain information without losing it. The sustainability of information is affected by several parameters, such as query sending rate, cache dropping rate, initial number of information pieces owned by a vehicle, vehicle mobility. We explored the impact of each of these parameters, although we could not examine all possible combinations and correlations among them.

5.1 Road Snapshots

A first set of simulations, which we dubbed “road snapshots”, was aimed at investigating the effectiveness of the query/cache mechanism *without* vehicle mobility. The goal is to determine whether the choice of parameters related to queries and caching allows the sustainability of information in a simple setting in which vehicles do no enter or exit the road segment. We set the simulation parameters as shown in Table 1.

We assume that, at the beginning of simulation time, each node holds 10 random pieces of information out of 50. Figure 4 presents the temporal evolution of the information density along the road, expressed in copies/m. The results

Table 1
Simulation parameters for road snapshots

Parameter	Value
Arrival distribution	static
Arrival rate	n.a.
Speed	0 m/s
Intervehicular distance	30 m
Initial cache content (unless differently specified)	10 out of 50

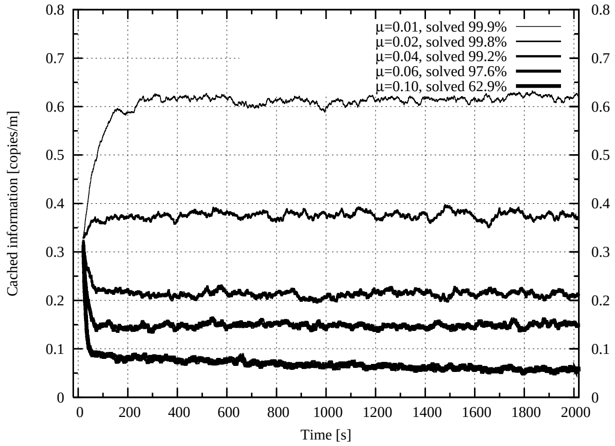

Fig. 4. Cached information density as a function of simulation time, for different cache dropping rates, initial cache content 10 out of 50 - road snapshot

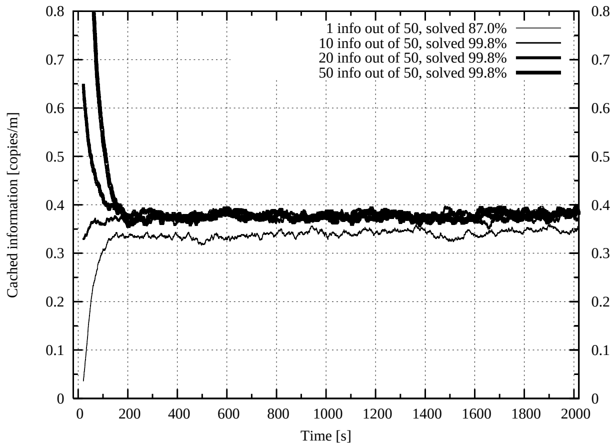

Fig. 5. Cached information density as a function of simulation time, for different initial cache content, $\mu = 0.02$ - road snapshot

show that all considered values of cache dropping rate, μ, yield a sustainable network traffic within the simulation time span (i.e., the nodes' caches are never drained and information exchange continues throughout the simulation). However, in the case of $\mu = 0.1$ we observe that the cached information density decreases with the passing of time, thus the system eventually gets depleted of all information pieces.

The impact of different initial cache content is shown in Figure 5, for the case of $\mu = 0.02$. The results in the plot suggest that, regardless of the initial number

Table 2
Simulation parameters for deterministic vehicle arrivals

Parameter	Value
Arrival distribution	deterministic
Arrival rate	0.5 vehicles/s
Speed	15 m/s
Initial cache content (unless differently specified)	10 out of 50

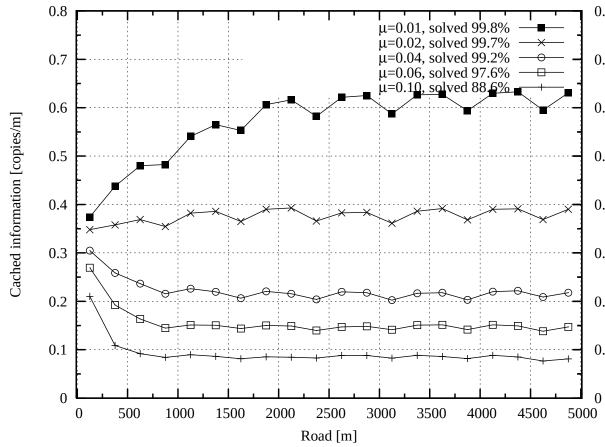

Fig. 6. Cached information density along the road for different cache dropping rates and initial cache content of 10 out of 50 - deterministic arrivals

of pieces in the cache, a "sensible" choice of μ is able to sustain the network through the whole duration of the simulation. Interestingly, the steady-state value of information density along the road is approximately 0.36-0.4 copies/m, which corresponds to a steady-state content of caches of approximately 11-12 pieces per vehicle. The only exception is the 1-out-of-50 case, in which the steady-state cache content is about 10: this is due to the fact that a few pieces were permanently lost during the initial transient.

5.2 Deterministic vehicle arrivals

In the second set of simulations we introduce mobility, assuming at first a deterministic arrival of vehicles at the beginning of the road segment, so that no connectivity gaps ever occur along the road (recall that vehicle move at constant speed). Table 2 summarizes the choice of parameters.

Figure 6 shows that, setting the initial cache content to 10 pieces out of 50, as done in the experiment of Figure 4, yields sustainable information exchange for all considered values of μ, including $\mu = 0.1$ for which cache depletion was observed in the static case. The arrival of fresh information on vehicles entering the network guarantees that even high cache dropping rates do not affect the information sustainability. The probability density function (pdf) of the amount of cached information is shown in Figure 7.

For the same reason, as shown in Figure 8, the inflow of fresh information fills

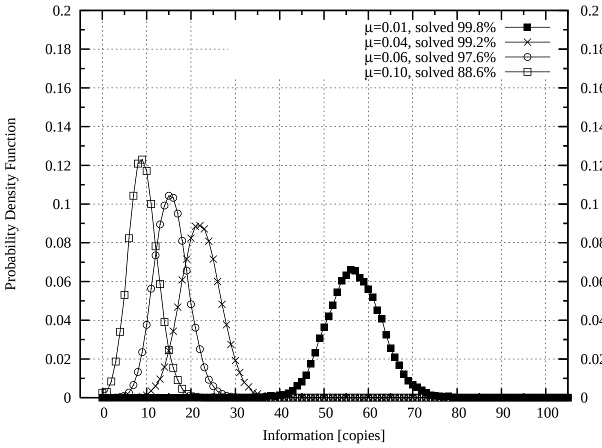

Fig. 7. Probability density function of amount of cached information, for different cache dropping rates - deterministic arrivals

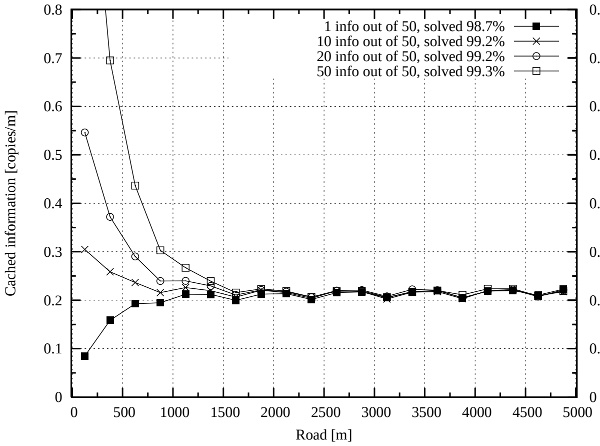

Fig. 8. Cached information density along the road, for different initial cache content and $\mu = 0.04$ - deterministic arrivals

the loss of pieces of information when the initial cache content is only 1 out of 50; when $\mu = 0.04$, for any initial cache content, the mean information density settles at around 0.22 copies/m (approximately 7 pieces per vehicle cache).

5.3 Exponential vehicle arrivals

The third set of simulations features the same scenario considered in the previous experiments, with the exception of *exponential* vehicle interarrivals. Exponential interarrivals shed the seamless connectivity assumption, allowing gaps to be formed if the distance between two vehicles is larger than the radio range.

The effect of gaps is quite disruptive, as witnessed by Figure 9: cache dropping rates that did not affect the information sustainability in the deterministic arrival case now barely allow the data to propagate for a few kilometers before the cache at vehicles loses all content. Although we only simulated a 5-km stretch of road for the sake of simulation scalability, the downward trend of all curves suggests that, unless the information is rebroadcast by access points

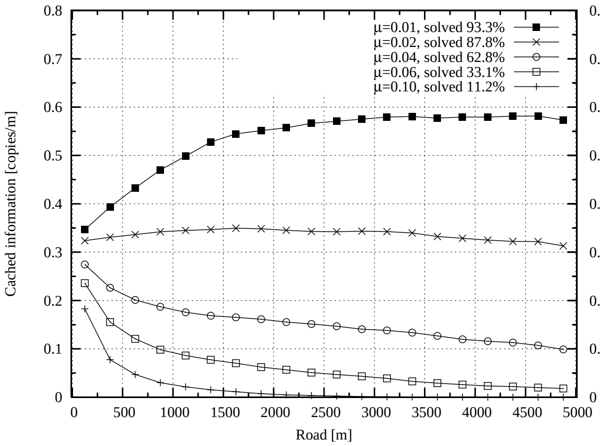

Fig. 9. Cached information density along the road, for different cache dropping rates - exponential arrivals

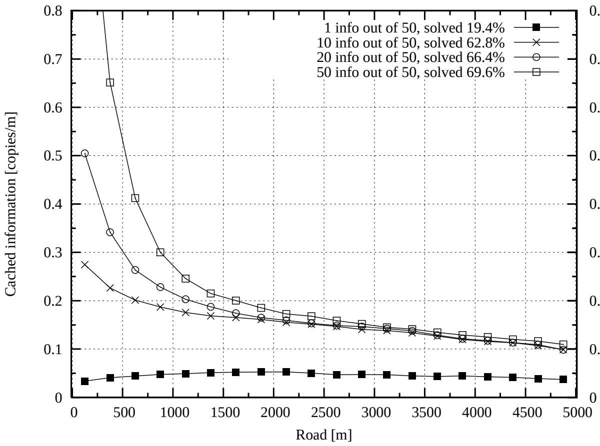

Fig. 10. Cached information density along the road, for different initial cache content and $\mu = 0.04$ - exponential arrivals

along the road, the system is destined to be depleted in the long run.

Similarly, if a dropping rate of $\mu = 0.04$ (which does not guarantee sustainability) is selected and different initial cache contents are tested (Figure 10), all curves of cached information density exhibit a decreasing, convergent behavior along the road. The 1-out-of-50 pieces case does not join the other curves at any point within the selected experiment, its value at or below 0.05 copies/m. Failure in reaching the common behavior can be explained considering that (i) exponential arrivals create disconnected "clusters" of vehicles, and (ii) vehicles enter the road caching a single piece of information each. The combination of these two factors leads to a segmented network, where each group of intercommunicating nodes only owns a small fraction of the complete information set, being at the same time unable to acquire any of the remaining pieces of information. The chosen μ allows the groups of vehicles not to loose the few pieces they hold along the road: this is consistent with the fact that the same μ is capable to provide higher quantities of cached information when initial conditions are less prohibitive.

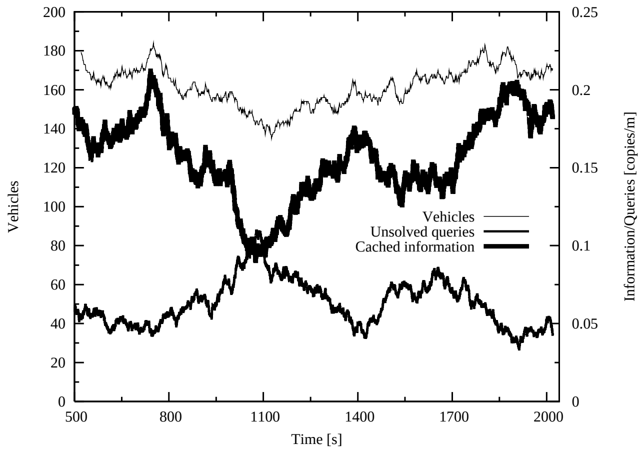

Fig. 11. Number of vehicles, unsolved queries and cached information density on a single-lane road as functions of simulation time

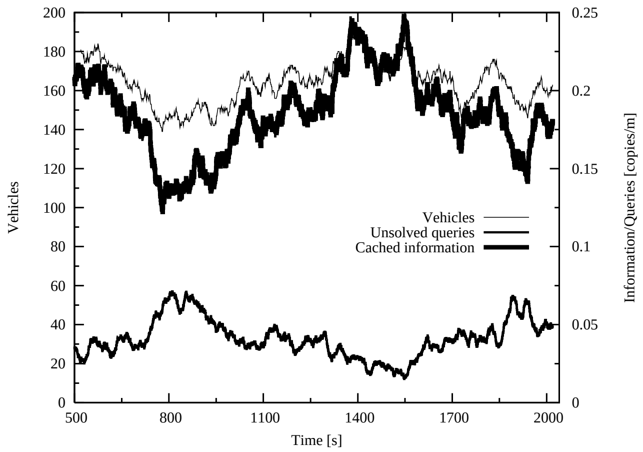

Fig. 12. Number of vehicles, unsolved queries and cached information density on a three-lane road as a function of simulation time

5.4 Lane speed

We now look at the impact of lane speed and compare a single-lane scenario (Figure 11) to a three-lane scenario (Figure 12) in terms of three metrics: number of vehicles, unsolved queries and cached information density.

Vehicles entering the single-lane road travel at a constant speed of 15 m/s (around 50 km/h), while the three-lane case has vehicles traveling at 15, 25 and 30 m/s, respectively (no lane change allowed while vehicles are on the stretch of road under examination).

Lanes have different exponential arrival rates, set in a way that the average number of vehicles is the same on each lane. Also, arrival rates are chosen so that the overall average number of vehicles on the road segment is the same as that in the exponential case we studied before.

The plots in Figures 11 and 12 report the number of unsolved queries normalized to the road length, the cached information density and the number of vehicles on the road, as functions of simulation time. Note that unsolved queries are requests that were not satisfied because they could not reach a vehicle possessing the information [6] : this may happen because of collisions involving query messages or connectivity gaps which prevent the query spreading.

We observe the following: firstly, a general strong correlation exists between the number of nodes on the road, the number of unsolved queries and the number of cached pieces of information. As a matter of fact, a common derivative sign is present in all of the three curves, regardless of the speed of vehicles on each lane. Such sign is the same for the number of nodes and the quantity of stored information, while it is the opposite for unsolved queries, whose curve is thus in counterphase with respect to those of the previous two metrics. The key parameter among the three is however the number of vehicles, from which the other two depend upon: the larger the number of nodes in the network, the higher the network connectivity and the information availability, since, on average, all nodes carry the same quantity of information. In other words, a higher number of vehicles reduces the probability that a query remains unsolved because of lack of connectivity or unavailability of the requested information. From that, we can infer that a higher arrival rate or a more regular distribution can positively affect the system performance.

By taking a comparative look at each figure, we observe that, when vehicles move at different speed on the road, the cached information density is higher and the number of unsolved queries decreases. We can notice that in Figure 11, small reductions of the number of nodes lead to remarkably negative performance peaks in terms of unsolved queries. This is caused by the zero relative speed of nodes: as a matter of fact, with no relative mobility of vehicles, the neighbors of a certain node are always the same from the beginning to the end of the road, thus there is the risk that isolated groups of nodes lose a particular piece of information which can no longer be retrieved within the group. Moreover, the relative rigidity of node movements makes it difficult also for connected vehicles to reach pieces of information which are stored at distant nodes, and this condition holds throughout the road, affecting multiple separate requests for the same information. Instead, a three-lane mobility of nodes (Figure 12) reduces such problems, thus enhancing the distribution of information among the vehicles.

5.5 Information restoring

The last set of tests was focused on the presence of elements that restore lost pieces of information: fixed Access Points (APs) or moving servers (e.g., buses). Both APs and buses are assumed to permanently hold all pieces of information and to reply to any query they receive with the desired piece of information. Therefore, the closer to an AP or a bus, the more likely that a query is satisfied. Simulation settings are identical to the exponential arrival case with no APs or buses.

[6] Thanks to retransmissions, the probability that an information message is lost is negligible

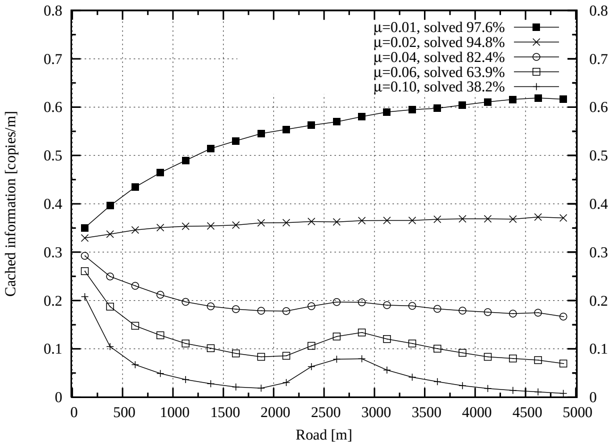

Fig. 13. Cached information density along the road, for different cache dropping rates - single AP

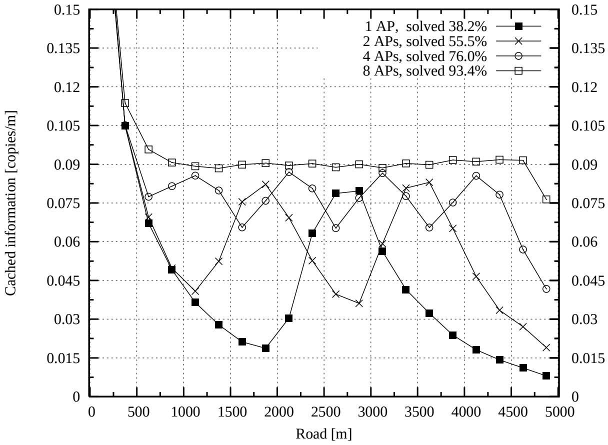

Fig. 14. Cached information density along the road, for different number of APs - $\mu = 0.1$

Figure 13 illustrates the effect of a single AP, placed halfway along the road, on the cached information density. By comparing it with Figure 9, it is straightforward to observe that the AP is responsible for a sudden surge halfway along the road, but its effects wane quickly because of the cache dropping rate, as soon as the vehicles leave the AP radio range. Thus, the AP turns out to be just a temporary relief to information starving.

Similarly, Figure 14 takes a non-sustainable cache dropping value under exponential arrivals ($\mu = 0.1$) and plots the progress of cached information density when 1, 2, 4 or 8 APs are evenly scattered along the road, highlighting that there is a critical distance between access points (approx. 500 m in our case, corresponding to the case with 8 APs) that defeats the losses caused by cache dropping.

A different, smoother behavior appears when moving buses are carrying and spreading the information upon being queried (Figure 15). In this case, where μ is still equal to 0.1, sustainability is always guaranteed, as long as there is a sizable bus density. Indeed the level of information density is strictly correlated to the ratio between buses and vehicles (hence, the bus density).

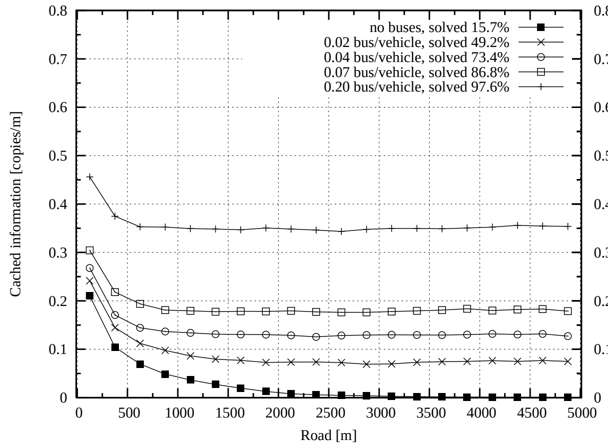

Fig. 15. Cached information density along the road, for different ratios of buses and vehicles - $\mu = 0.1$

Table 3
Notation used

Symbol	Meaning
R	Radio range
L	Road length
H	Average hop length
n	No. of information pieces initially stored by a vehicle
N	Total number of information pieces
V	Number of vehicles on the road
α	Vehicle arrival rate
ω	Vehicle exit rate
Δ	Inter-vehicle distance
σ	Duration of a backoff slot
λ	Request rate per information piece by a vehicle
Λ	Total request rate of information pieces by a vehicle
Λ_q	Total sending rate of queries by a vehicle
Λ_p	Total sending rate of information messages by a vehicle
μ	Information dropping rate
$P_s(i)$	Probability of information successful retrieval given that there are i copies
$\bar{P}_s$	Average prob. of information successful retrieval

6 Modeling information exchange in Infoshare

To study the dynamics of the information exchange and the system ability to maintain a given information density along the road, we have developed an analytical model of the IVN.

We consider that there are no APs along the road segment and all vehicles implement the Infoshare application. Our model is based on the following assumptions:

(i) vehicles enter a single-lane road according to a Poisson process with parameter α; each of them has n out of N initial pieces of information;
(ii) the number of vehicles on the road is constant and equal to V;
(iii) vehicles stay on the road for an exponentially distributed time with mean $1/\omega$;
(iv) the TTL parameter is set to infinity;
(v) transmission errors due to the wireless channel are neglected.

Assumption *(ii)* is based on the following observations. Considering Poisson arrivals of vehicles, one can think of the road as an $M/G/\infty$ queue, for which the number of customers is Poisson distributed. Then, given the large number of vehicles in the system, the Poisson distribution can be approximated by its mean value $V = \alpha/\omega$. Assumption *(iii)* is justified by the fact that, as the number of customers in an $M/G/\infty$ queue does not depend on the service time distribution, the number of vehicles is insensitive to the time they spend on the road.

As before, let us denote with λ and μ the request and dropping rate, respectively, for a generic piece of information. For simplicity, in the following we assume that the rates λ and μ are the same for all information pieces, however extending the analysis to the case where each information is associated with different request and dropping rates is straightforward. Also, we define $P_s(i)$ as the probability of successful retrieval of a generic piece of information given that there exist i copies on the road. Probability $P_s(i)$ will be computed in the next subsection.

Being the behavior of all information pieces identical, let us focus on a single information piece and study the dynamics of the number of its copies in the system. The dynamics of the number of copies can be represented by the continuous-time, birth-and-death process shown in Figure 16. The number of states is equal to V, i.e., the number of vehicles on the road (see Assumption *(ii)*). Indeed, all vehicles can potentially get a copy of the information piece; it follows that there can be at most V information copies in the system for a given piece.

The state transitions are determined by the following dynamics. Consider the generic state $i > 0$; the number of copies can increase thanks to successful queries or newly arrived vehicles that possess the tagged information. Note that in the former case only vehicles that do not already have the tagged information in their caches can request it. For $i = 0$, instead, no queries can be satisfied, i.e., only new vehicle arrivals can bring the tagged information into the system. The transition probability f_i from state i to state $i + 1$ is therefore given by:

$$f_i = \begin{cases} \alpha \frac{n}{N} & i = 0 \\ \alpha \frac{n}{N} + (V - i)\lambda P_s(i) & 1 \leq i < V \end{cases} \tag{1}$$

The number of copies can decrease either because a vehicle storing the tagged information exits the road or because the information is dropped. Hence the transition probability b_i from state i to state $i-1$ is given by:

$$b_i = i(\omega + \mu) \tag{2}$$

Given the transition probabilities above, the steady-state distribution $\pi = \{\pi_i\}$, $0 \leq i \leq V$ can be easily computed through standard techniques.

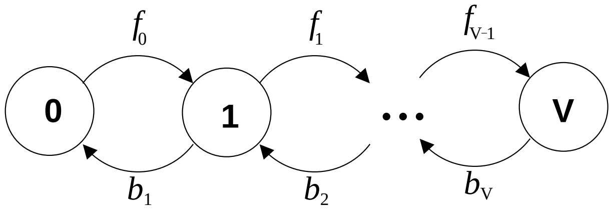

Fig. 16. Birth-and-death process describing the information copies dynamics for a single piece of information

6.1 Computation of the probability of successful information retrieval

In this section we describe how to compute the probability $P_s(i)$ that a piece of information is retrieved successfully by a vehicle requesting it, given that there are i copies of the given piece in the system.

Having fixed the number of copies i, we denote by $\rho_i = i/L$ the spatial density of the piece, which is assumed to be constant along the road. We consider an arbitrary vehicle on the road, and denote by $m = \lfloor R/\Delta \rfloor$ the number of vehicles that are in transmission range on either side of it. Assuming that m is small with respect to the 802.11 backoff window size and that stations do not operate in saturated conditions, we neglect collisions among vehicles in radio range of each other (these collisions occur when two or more vehicles attempt to transmit at exactly the same time slot). Indeed, in our system transmission failures are primarily due to the hidden terminal problem, i.e. to simultaneous transmissions from vehicles not in range of each other, which overlap at the receiver.

In the diagram of Figure 17, the tagged vehicle is indexed by 0, and vehicles on the right hand side of it, equally spaced from each other, are numbered progressively. Vehicle m is the leftmost one in radio range of vehicle 0. Let Λ_q and Λ_p be the rate at which a node attempts to transmit a generic query or a generic information message, respectively (we do not distinguish, for simplicity,

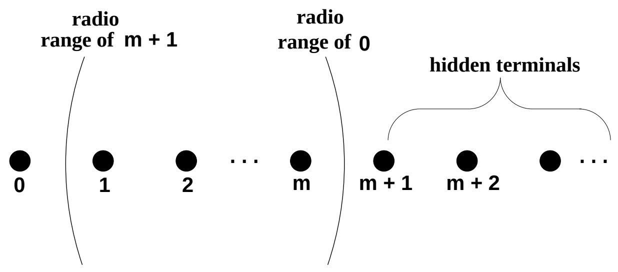

Fig. 17. Numbering of nodes on one side of a tagged vehicle

among packets locally generated or relayed on behalf of other nodes). These sending rates will be computed later in the section.

In our model, we assume that transmission attempts at the vehicles occur according to independent Poisson processes. Suppose that at time $t = 0$ the tagged vehicle starts transmitting a query. We first compute the probability Q_c that none of the m vehicles within radio range of the tagged node (on one side of it) receive the query successfully. This occurs when all of the m nodes simultaneously receive a packet sent by node $m + 1$. When this event takes place, two situations are possible: (1) the query collides with another query message; (2) the query collides with an information message. Case (1) occurs when vehicle $m + 1$ starts transmitting a query in the time interval $[-T_q, T_q]$, where T_q is the duration of a query message. Case (2) occurs in three different subcases: (2-i) when vehicle $m+1$ starts sending the RTS packet to a node on its left, in the interval $[-T_{\text{RTS}}, T_q]$; (2-ii) when vehicle $m+1$ starts sending the RTS packet to a node on its right, in the interval $[-T_{\text{RTS}}-T_{\text{CTS}}-T_d, T_q]$, where T_d is the duration of the data packet; (2-iii) when vehicle $m + 1$ sends a CTS or an ACK packet in response to an information message sent by some other node $k > m+1$, in the intervals $[-T_{\text{CTS}}, T_q]$ and $[-T_{\text{ACK}}, T_q]$, respectively. Let $T_s = T_{\text{RTS}}+T_{\text{CTS}}+T_d+T_{\text{ACK}}$ be the total duration of the four-way handshake. Considering that only one half of the information messages sent by a node is directed to the left or to the right of it, we obtain:

$$Q_c = 1 - e^{-\Lambda_q 2T_q} e^{-\frac{\Lambda_p}{2}(T_s+3T_q+T_{\text{RTS}}+T_{\text{CTS}})}$$

Probability Q_c can be used to compute other two important variables. The first one is the probability Q_{1h} that the query does not propagate beyond the first hop (we are still considering only one side of the propagation). Let $Q_s = (1 - Q_c)$. Probability Q_{1h} is given by:

$$Q_{1h} = Q_c + \sum_{k=1}^{m-1} \left(Q_c Q_s^k \prod_{j=1}^{k} (1 - Q_s^j) \right) + Q_s^m \prod_{j=1}^{m} (1 - Q_s^j) \tag{3}$$

In (3), the first term represents the probability that none of the m neighbors on one side of the requesting node receives correctly the query due to collisions with hidden terminal $m + 1$; the following terms account for the cases where k neighboring nodes ($1 \leq k \leq m$) successfully receive the query but they are unable to further propagate it due to collisions with hidden terminals $(m + 1), \ldots, (m + k)$.

The second one is the probability Q_h that the message stops propagating after a given hop (different from the first one). To compute this probability, we impose that m consecutive nodes fail to propagate the query beyond the last vehicle reached by the query:

$$Q_h = \prod_{k=1}^{m}(1 - Q_s^k)$$

Considering that nodes in radio range of a transmitter are equally likely to act as relay, we obtain that the average hop length is equal to $H = \Delta(1+m)/2$. The number of hops done by a query after the first hop is described by a geometric distribution of parameter Q_h, hence the distribution of the physical distance reached by the query can be approximated with an exponential distribution of parameter $\gamma = Q_h/H$.

We can now compute the probability $P_s(i)$ of successful information retrieval. On each side of a requesting node, three conditions must be met in order to retrieve the requested piece of information: (i) there is at least one copy of the requested piece on the road; (ii) the query propagates beyond the first hop[7]; (iii) a copy is reached before the query stops propagating due to collisions. With regards to condition (i), we observe that, given that there are i copies of the requested piece of information in the system ($i > 0$), with probability $\frac{2}{i+1}$, all of them are located just on one side of the requesting node; condition (ii) is satisfied with probability Q_{1h} computed above; the occurrence of condition (iii) can be easily computed thanks to the fact that both the distance reached by a query and the distance from the first copy are described by an exponential distribution, of parameters γ and ρ_i, respectively. Thus, with probability $\frac{\gamma}{\gamma+\rho_i}$, a query stops before reaching the nearest copy of the requested piece. Putting all together, we obtain

$$\begin{aligned} P_s(i) = 1 - \frac{2}{i+1}\left[Q_{1h} + (1 - Q_{1h})\frac{\gamma}{\gamma+\rho_i}\right] - \left(1 - \frac{2}{i+1}\right) \\ \left[Q_{1h} + (1 - Q_{1h})\frac{\gamma}{\gamma+\rho_i}\right]^2 \end{aligned}$$

As a special case, $P_s(0) = 0$. To quantify the impact of collisions on the system performance, we compute the average probability $\bar{P}_s$ to solve a query, conditioned to the presence of at least one piece of the requested information in the system. Notice that, in absence of any collision, this quantity would be equal to 1. The proposed metric can be computed as,

$$\bar{P}_s = \frac{\sum_{i=1}^{V} \pi(i)\Lambda(V-i)P_s(i)}{\sum_{i=1}^{V} \pi(i)\Lambda(V-i)} \tag{4}$$

[7] We neglect the case where the piece of information is within one hop from the requesting node. Indeed the probability of such an event is very low, unless the number of copies on the road is very large, but in this case the probability of unsolved query is in any case negligible

We still need to evaluate the aggregate sending rates Λ_q and Λ_p of queries and information messages, respectively, at a given vehicle. We condition on the number i of copies present in the system for a generic piece of information, and compute the conditioned sending rates $\lambda_q(i)$ and $\lambda_p(i)$ of queries and information messages, respectively. Then we have:

$$\Lambda_q = \sum_{i=0}^{V} \pi(i)\lambda_q(i) \quad ; \quad \Lambda_p = \sum_{i=0}^{V} \pi(i)\lambda_p(i)$$

We start considering the propagation of queries. The average distance $D_q(i)$ traveled by a query on each side of the source (in case it is propagated beyond the first hop) is equal to the minimum of three quantities: (i) the distance from the first copy of the requested piece of information; (ii) the distance from the end of the road; (iii) the distance reached by the query before it stops propagating due to collisions. To simplify the analysis, we assume that the length of the road segment on the left or on the right of a tagged vehicle is exponentially distributed with parameter $\beta = 2/L$. It follows that $D_q(i) = (\rho_i^{-1}+\beta^{-1}+\gamma^{-1})^{-1}$. Notice that all vehicles within distance $D_q(i)$ are expected to forward the query once. Thus we obtain

$$\lambda_q(i) = 2\Lambda(V-i)(1-Q_{1h})\frac{D_q(i)}{\Delta}$$

The propagation of information messages is more difficult to study. Indeed, we must consider that not all sent queries reach a copy of the requested information, however, if they reach a copy, information messages are propagated back reliably, because they are sent in unicast mode (we have verified by simulation that information messages are dropped with negligible probability while being transferred to the requesting vehicle).

The average number $M(i)$ of information messages delivered back to a requesting vehicle is,

$$\begin{aligned} M(i) &= \frac{2}{i+1}(1-Q_{1h})\frac{\rho_i}{\rho_i+\gamma} + 2\left(1-\frac{2}{i+1}\right)(1-Q_{1h})\frac{\rho_i}{\rho_i+\gamma} = \\ &= \frac{2\,i}{i+1}(1-Q_{1h})\frac{\rho_i}{\rho_i+\gamma} \end{aligned}$$

and each of them travel an average distance $1/(\rho_i+\gamma)$.

We also need to compute the collision probability P_m of an information message. Following the same reasoning used above to compute Q_c, we neglect collisions between in-range transmitters, and evaluate the probability Q_m that an information message collides with transmissions originated from the vehicle at position $m+1$:

$$Q_m = 1 - e^{-\Lambda_q(T_q+T_{\mathrm{RTS}})}e^{-\frac{1}{2}\Lambda_p(3T_{\mathrm{RTS}}+2T_s)}$$

Referring to the diagram in Figure 17, we need to consider that a transmission from node 0 to a *specific* node j, $1 \le j \le m$ is subject to collision with the transmissions originated at all hidden terminals k, $m < k < m + j$. As a consequence, long hops are more vulnerable to collision than short hops. Assuming that the hop length is uniformly distributed in $[1 \ldots m]$, we obtain

$$P_m = 1 - \sum_{j=1}^{m} \frac{1}{m} (1 - Q_m)^j$$

Finally, accounting for the retransmission of information messages, we obtain

$$\lambda_p(i) = \frac{\Lambda(V - i)M(i)}{\rho_i H(1 - P_m)} \qquad (i > 0)$$

As a special case, $\lambda_p(0) = 0$.

A fixed point approximation is needed to solve for the entire system, since some parameters of the model depend on each other in a nonlinear way. The convergence to a unique solution has been verified experimentally in several different cases.

7 Model validation and exploitation

In this section we validate our model comparing analytical predictions with simulation results in a few different cases. Then we show how the model can be exploited to design an Infoshare system achieving a desired performance. In Figure 18 we show the average number of copies present in the system for a given piece of information, as a function of the initial number of pieces given to a vehicle entering the road segment, for different values of the dropping rate μ. We observe excellent agreement between model (continuous lines) and simulation (marks) in all considered cases.

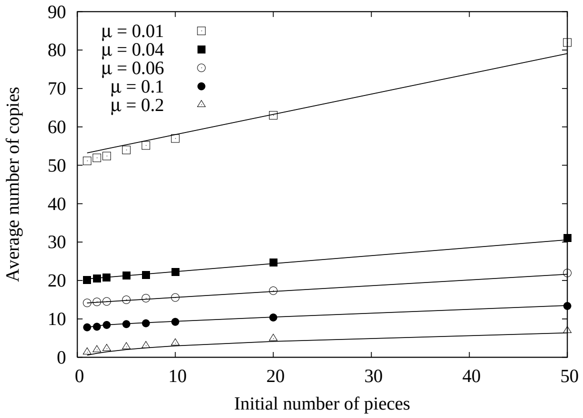

Fig. 18. Average number of copies for a given piece of information. Comparison between model (lines) and simulation (marks)

Indeed, the model is able to accurately predict not only the average number of copies in the system, but also the distribution of the number of copies. Figure 19 presents the pdf of the number of copies for different values of μ and initial number n of pieces equal to 10. All distributions reported in the

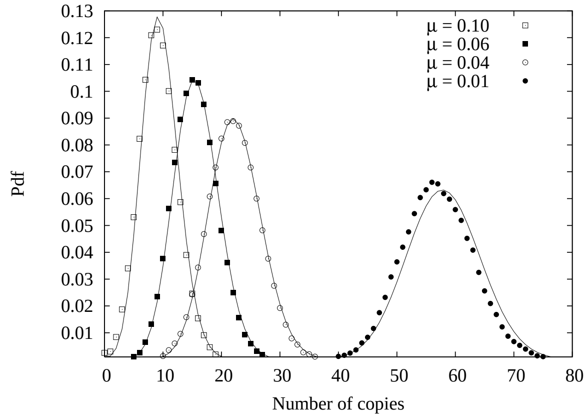

Fig. 19. Distributions of the number of copies for a given piece of information. Comparison between model (lines) and simulation (marks)

plot correspond to well-behaved systems in which the probability of having at least one copy of a given piece of information is close to 1 (i.e., the probability of zero copies is negligible).

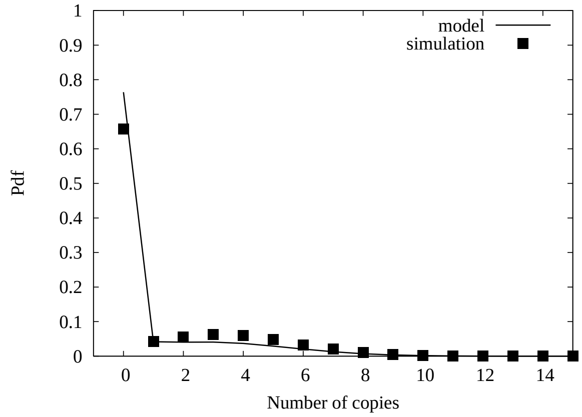

Fig. 20. Distribution obtained when $\mu = 0.2$ and $n = 1$. Comparison between model (line) and simulation (marks)

Figure 20 shows, instead, the pdf obtained in the case of $\mu = 0.2$ and $n = 1$. The distribution suggests that most of the time (i.e., with probability of about 0.75) it is not possible to find any copy of the requested information piece. Occasionally, the information piece is able to propagate in the system being replicated in a few number of copies. The model obtains a good match with simulation results also in this case, allowing to estimate with a fairly good degree of accuracy the probability $\pi(0)$ that a given piece of information is not available in the system.

In Figure 21 we compare analytical prediction and simulation results for the average probability to solve a query, conditioned to the presence of at least one piece of information, i.e., the quantity $\bar{P}_s$ computed in (4). The plot refers to the case of $n = 5$. We observe that $\bar{P}_s$ decreases for increasing values of the dropping rate μ. This is due to the increased amount of query traffic, both because vehicles produce more queries, and because each query has to go through a larger number of hops before reaching a copy of the requested information.

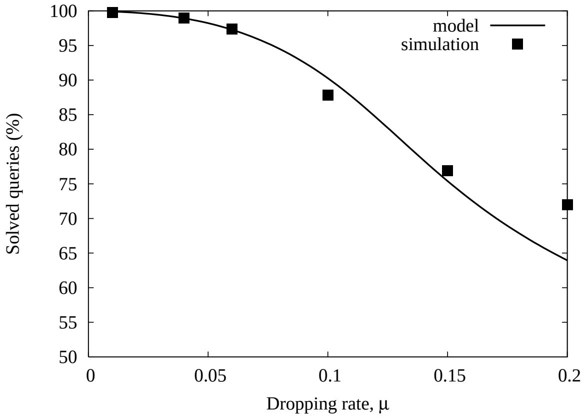

Fig. 21. Conditioned probability to solve a query, when $n = 5$. Comparison between model (line) and simulation (marks)

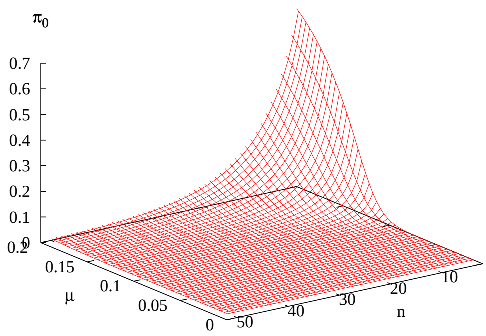

Fig. 22. Probability that a given piece of information in not present in the system, as a function of μ and n, as computed by the model

Having validated the model, we can exploit it to design an Infoshare system achieving a desired operational point. For example, Figure 22 reports the probability $\pi(0)$ that a given piece of information is not present in the system, for a large number of combinations of parameters μ and n. The plot allows the identification of the set of parameters that makes the probability $\pi(0)$ negligible, or smaller than a desired threshold. As a last remark, the results in the plot required a few seconds of computational times to be derived by the model, whereas they would require many days to be obtained from simulation.

8 Conclusions

We addressed the problem of information caching and delivery in intervehicular networks. An application, called Infoshare, based on a pull paradigm and application-level routing was outlined and its performance studied in a network scenario featuring a unidimensional, one-way road. Several parameters were investigated in order to gauge their impact on the system performance. Simulation results have shown that modeling the vehicle arrival process according to different distributions has non-negligible repercussions on the system performance. In the case of a multilane road with vehicles moving at different speeds, a prominent result is the impact of vehicle relative speeds with two effects offsetting each other: too quick topology changes leading to volatile connectivity and too slow topology changes resulting in reduced information exchange due to isolated groups of vehicles. Furthermore, simulation results have suggested that the ratio of the information requesting rate by vehicles to the cache dropping rate plays a fundamental role in promoting the efficient exchange of information among vehicles. We therefore developed an analytical model and use it to study the relationship between the system parameters and the sustainability of information along the road.

Finally, we would like to highlight that we also run some simulations using realistic mobility models for vehicular traffic, and the obtained results do not change significantly, as long as the density of vehicles in the network is more or less the same, and connectivity is maintained.

References

[1] R.H. Frenkiel, B.R. Badrinath, J. Borras, R.D. Yates, The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data, IEEE Personal Communications Magazine, 7 (2000) 66–71.

[2] Y.-B. Ko, N.H. Vaidya, Location-aided Routing (LAR) in Mobile Ad Hoc Networks, Wireless Networks 6 (2000) 307–321.

[3] L. Yin, G. Cao, Supporting Cooperative Caching in Ad Hoc Networks, IEEE Transactions on Mobile Computing, 5 (2006) 77–89.

[4] P. Bergamo, M. Cesana, D. Maniezzo, G. Pau, K. Yao, D. Whiteman, M. Gerla, IEEE 802.11 Wireless Network under Aggressive Mobility Scenario, in; Proc. International Telemetry Conference ITC/USA2003, Las Vegas, 2003.

[5] D. Bottazzi, A. Corradi, R. Montanari, Context-Awareness for Impromptu Collaboration in MANETs, in; Proc. Third IEEE International Symposium on Network Computing and Applications (NCA'04), Cambridge, 2004, pp. 339–342.

[6] R. Castaneda, S.R. Das, M.K. Marina, Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks, in; Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, 1999, pp. 186–194.

[7] S. Das, A. Nandan, G. Pau, M.Y. Sanadidi, M. Gerla, SPAWN: A Swarming Protocol for vehicular Ad-hoc Wireless Networks, in; Proc. First ACM Workshop on Vehicular Ad Hoc Networks (VANETs '04), Philadelphia, 2004. Berkeley, 2004, pp. 93–94.

[8] A. Ebner, H. Rohling, L. Wischhof, R. Halfmann, M. Lott, Performance of UTRA TDD Ad-Hoc and IEEE 802.11b in Vehicular Environments, in; Proc. IEEE 57th Vehicular Technology Conference Spring, Jeju, 2003, pp. 960–964.

[9] M. Fiore, C. Casetti, C.-F. Chiasserini, On-demand Content Delivery in Vehicular Wireless Networks, in; Proc. IEEE/ACM MSWIM 2005, Montreal, 2005, pp. 87–94.

[10] S. Ghandeharizadeh, S. Kapadia, B. Krishnamachari, PAVAN: A Policy

Framework for Content Availability in Vehicular Ad-hoc Networks, in; Proc. First ACM Workshop on Vehicular Ad Hoc Networks (VANETs ’04), Philadelphia, 2004, pp. 57–65.
[11] S. Ghandeharizadeh, B. Krishnamachari, C2P2: Peer-to-peer Network for On-demand Automobile Information Services, in; Proc. 15th International Workshop on Database and Expert Systems Applications (DEXA’04), Zaragoza, 2004, pp. 538–542.
[12] D.B. Johnson, D.A. Maltz, J. Broch, DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, in: C. E. Perkins (Ed.), Ad Hoc Networking, Addison-Wesley, 2001, pp. 139–172.
[13] A. Nandan, S. Das, G. Pau, M.Y. Sanadidi, M. Gerla, Cooperative Downloading in Vehicular Ad Hoc Networks, in; Proc. Wireless On demand Networks, Systems and Services (WONS ’05), St. Moritz, 2005, pp. 32–41.
[14] J. P. Singh, N. Bambos, B. Srinivasan, D. Clawin, Wireless LAN Performance under Varied Stress Conditions in Vehicular Traffic Scenarios, in; Proc. IEEE 56th Vehicular Technology Conference Fall, Vancouver, 2002, pp. 743–747.
[15] J.P. Singh, N. Bambos, B. Srinivasan, D. Clawin, Y. Yan, Empirical Observations on Wireless LAN Performance in Vehicular Traffic Scenarios and Link Connectivity based Enhancements for Multihop Routing, in; Proc. IEEE Wireless Communications and Networking Conference (WCNC ’05), New Orleans, 2005, pp. 1676–1682.
[16] W.H. Yuen, R.D. Yates, C.W. Sung, Effect of Node Mobility on Highway Mobile Infostation Networks, in; Proc. ACM MSWIM 2003, San Diego, 2003, pp. 82–91.
[17] W.H. Yuen, R.D. Yates, S.C. Mau, Exploiting Data Diversity and Multiuser Diversity in Noncooperative Mobile Infostation Networks, in; Proc. IEEE INFOCOM 2003, San Francisco, 2003, pp. 2218–2228.
[18] Ethereal, `http://www.ethereal.com/`
[19] The network simulator ns2, `http://www.isi.edu/nsnam/ns/`