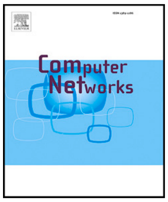
ARTICLE IN PRESS

Computer Networks xxx (xxxx) xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Diversity-improved caching of popular transient contents in Vehicular Named Data Networking

Marica Amadeo [a,b], Giuseppe Ruggeri [a,b], Claudia Campolo [a,b,*], Antonella Molinaro [a,b,c]

[a] University Mediterranea of Reggio Calabria, Italy
[b] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Italy
[c] Laboratoire des Signaux et Systémes (L2S), CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France

ARTICLE INFO

Keywords:
Vehicular Named Data Networking
Caching
Transient contents

ABSTRACT

In-network caching, natively enabled by Named Data Networking (NDN), is effective to speed up content retrieval in vehicular environments, where a large variety of contents are typically *transient*, i.e., they expire after a certain amount of time, and may exhibit different popularity profiles. Lifetime and popularity play a crucial role in the content caching decision: intuitively, caching a popular content with long lifetime can be more useful than caching an unpopular one that is ready to expire and then to be dropped from the content store. At the same time, making nearby nodes caching different contents and, therefore, improving the caching diversity, can be crucial to get better delivery performance over the broadcast wireless medium. In this paper, we devise a novel distributed caching strategy where vehicles autonomously decide which content is to be locally cached according to the content residual lifetime, its popularity and the perceived availability of the same content in the neighborhood. The target is to cache with higher probability more popular contents with a longer lifetime, which are not already cached by a nearby node, thus improving the caching diversity in the neighborhood. As a result, vehicles can find the majority of distinct fresh and popular contents nearby, without flooding the network with content requests that have to reach the original source. Performance evaluation shows that the conceived solution outperforms representative benchmark schemes, by guaranteeing, among others, the shortest content retrieval time and the lowest network traffic load.

1. Introduction

Characterized by high mobility, intermittent connectivity and unreliable wireless channel, vehicular environments require robust and smart content delivery solutions in order to distribute among vehicles relevant information, such as high-definition maps, parking lots availability, traffic jam notifications, etc. Thanks to the native in-network caching and routing-by-name mechanisms, the Information Centric Networking (ICN) [1] architecture called Named Data Networking (NDN) [2] well copes with situations in which the content provider and the consumer are not simultaneously connected, and candidates itself as a prominent solution for such challenging environments [3].

The idea of using NDN in a vehicular context, let us refer to it as Vehicular NDN (VNDN), was originally devised in the pioneering works in [4,5]. VNDN leverages broadcast transmissions of request packets called Interests to retrieve contents *by name*. Any node owning a copy of the content can broadcast a Data packet to answer the request, otherwise it can further re-transmit the Interest according to proper routines, defined at the forwarding plane, that cope with possible broadcast storm issues [6,7].

As a result, in VNDN, data delivery is aligned with the intrinsic broadcast nature of the wireless channel [8]: by receiving and processing the broadcast Data packets, carrying the NDN name, each vehicle may infer what contents are *available* in its neighborhood. Based on this information, the vehicle could *diversify* the selection of the contents to cache. For example, if a content is transmitted over the wireless medium by close cachers, there is in principle less need for further caching it locally, since a consumer could retrieve it from those cachers.

In-network caching plays a crucial role in improving the network performance. Unlike other wireless mobile terminals such as smartphones, tablets, or Internet of Things (IoT) devices like sensors and actuators, vehicles do not have stringent limitations in terms of storage and power and, therefore, caching operations can be more easily supported. Notwithstanding, the number of contents to be exchanged

---

* Corresponding author at: University Mediterranea of Reggio Calabria, Italy.
E-mail addresses: marica.amadeo@unirc.it (M. Amadeo), giuseppe.ruggeri@unirc.it (G. Ruggeri), claudia.campolo@unirc.it (C. Campolo), antonella.molinaro@unirc.it (A. Molinaro).

https://doi.org/10.1016/j.comnet.2020.107625
Received 16 July 2020; Received in revised form 14 October 2020; Accepted 16 October 2020

Please cite this article as: Marica Amadeo, *Computer Networks*, https://doi.org/10.1016/j.comnet.2020.107625

is expected to be much bigger than the storage capacity of vehicles and selective caching decisions must be devised.

In recent years, different strategies have been designed for VNDN that consider content-related attribute(s) (e.g., the popularity [9]) or node-related feature(s) (e.g., the topological centrality), or a combination of them, as metrics for taking the caching decision [10,11]. While these caching metrics have been widely studied, less attention has been paid to the content availability measure, which captures the *presence of the same content cached/owned in the vehicle's neighborhood*. If neighboring vehicles cache the same contents redundantly, the distributed storage resources are not properly used. In a broadcast wireless environment like VNDN, it is instead crucial to improve the caching *diversity*, i.e., storing different contents in the caches of neighboring vehicles. This would imply that consumers have a higher chance to find distinct contents nearby, with no need of flooding the network with content requests traversing multiple hops towards the content owners. Diversifying the caching decision among nearby vehicles and therefore, limiting the caching redundancy in the vicinity, translates into better network performance in terms of reduced traffic load and content retrieval delay.

Moreover, we focus on a specific category of vehicular contents, the *transient* ones, i.e., those having a limited temporal validity (lifetime), whose importance is expected to grow in the near future due to the rise of multiple Internet of Vehicles (IoV) applications. With the exception of very few works [12–14], existing strategies typically apply to static contents like multimedia files and do not capture the transient nature of many vehicular contents. Parking lots availability and intelligent driving assistance are just examples of services that, due to the variation of the driving environment, take as input contents that change with time [15–17]. Also, pollution and road congestion measurements expire after a short time period since they have been generated at the source. Such a transient feature may largely impact on the performance of the data retrieval by determining cache inconsistency issues and lower cache hit ratio. Long-lasting contents should have higher probability to be cached. Indeed, caching contents that are ready to expire could uselessly waste storage resources and cause a high number of content replacement operations.

In NDN, the data producers can include in the packets a *Freshness Period* parameter, which identifies, in seconds, the lifetime of the content. In our previous work [12], we deployed a simple caching strategy that computes the caching probability based on the *Freshness Period*, but wiser decisions could be taken if this information is used together with other notable content's parameters. Therefore, in this work, we define a composite caching decision metric based on *lifetime*, *popularity* and *availability* parameters, which are crucial in the considered VNDN environment.

This paper provides the following contributions:

- We conceive a novel caching strategy, referred to as *Diversity-improved cAchiNg of popular Transient contEnts* (DANTE), which is specifically designed for VNDN and aims to create content diversity in the neighborhood of vehicles. It also privileges caching popular contents with a longer lifetime, in order to better use the storage space and limit the number of packet replacements forced by the content lifetime expiration.
- We design a set of minor modifications in the NDN node architecture and packet fields to support DANTE operations. Caching decisions are taken by vehicles autonomously, i.e., without the exchange of additional signaling messages.
- We conduct a wide simulation campaign in ndnSIM [18], the official simulator deployed by the NDN community, to compare the performance of the proposed strategy against other representative benchmark schemes in the literature. We consider two distinct vehicular scenarios, urban and highway, with vehicles implementing NDN over the IEEE 802.11 access interface. In both scenarios, results show that our solution reduces the content retrieval latency while maintaining the traffic load very low under all the considered settings.

The remainder of the paper is organized as follows. Section 2 provides an overview of the main basics of the NDN paradigm, with special focus on its customization to the vehicular environments. The literature addressing caching strategies is scanned in Section 3, by dissecting the main open issues. The proposal is presented in Section 4 and evaluated in Section 5, before concluding the paper in Section 6.

2. Vehicular NDN

2.1. The access layer technology for VNDN

Vehicle-to-Everything (V2X) communications refer to the exchange of data between a vehicle and various elements in the surroundings, including other vehicles, pedestrians, Internet gateways, and transport infrastructure, such as traffic lights, signs, and Road-Side Units (RSUs). IEEE 802.11 is considered among the mainstream V2X connectivity solutions. In particular, IEEE 802.11p, conceived as amendment of the IEEE 802.11 standard, now superseded and part of the IEEE 802.11 standard [19], was initially considered, due to its operation simplicity and the native support for ad hoc communications. 802.11p supports communications among vehicles (Vehicle-to-Vehicle, V2V) and between vehicles and RSUs (Vehicle-to-Infrastructure, V2I). More recently, the interest for 802.11-based V2X connectivity revamped thanks to the creation of a new IEEE task group, aiming to investigate advanced physical-layer technologies that enhance the .11p throughput and coverage [20]. Therefore, in the following we refer to 802.11 as the access layer technology of VNDN, in agreement with the existing literature [3].

At the Medium Access Control (MAC) layer, 802.11p relies on a contention-based access leveraging the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. A node wishing to transmit senses the medium to detect its occupancy status. An immediate transmission is allowed, if it is idle, otherwise, a mechanism based on random backoff is performed to reduce the probability of collisions. Collisions and contention over the wireless channel, becoming particularly severe under heavy traffic load conditions, highly affect the performance of broadcast transmissions, that are crucial for content dissemination in many vehicular applications.

2.2. Forwarding fabric in VNDN

Communication in VNDN [4,5] is pull-based and consists of the broadcasting of two messages: the *Interest*, issued by *consumers* to request a content by name, and *Data*, which carries the requested contents and is sent back by any node owning the content, either the original producer or a cacher, in the following *both referred to as providers*.

Each VNDN node maintains three tables at the forwarding plane: the *Content Store* (CS), which caches incoming Data packets; the *Pending Interest Table* (PIT), which temporary stores the transmitted Interests that have not been consumed by the Data yet, and the *Forwarding Information Base* (FIB), which is used to forward the Interests. In wired NDN deployments, a routing protocol is implemented to fill the FIB entries. However, due to the dynamics of vehicular ad hoc environments, VNDN forwarding strategies in the literature, e.g., [4–6,21], usually work without a routing protocol and leverage *controlled-based flooding and overhearing* to limit packet collision and redundancy.

Without loss of generality, in this paper, we consider the pioneering work in [4] as the reference forwarding process in VNDN. More specifically, at the Interest reception, a VNDN node N looks for a matching Data in the CS and, if the content if found, it directly broadcasts the Data. Otherwise, N looks for a matching in the PIT. If this is the case, the Interest is discarded since there is an equal request pending. Otherwise, the node includes the Interest in the PIT and issues a new Interest broadcasting but delays the transmission of a defer time. If, during the defer time, N overhears the Data packet or the same Interest transmitted by another node, then it cancels its own transmission and the related PIT entry.

To speed up the Interest dissemination, in [4], the computation of the defer time for Interest transmissions depends on the distance from the Interest sender. When broadcasting the Interest, each sender S includes its Global Positioning System (GPS) coordinates. The receiving node calculates its distance from S and sets a random defer time that is inversely proportional to such distance. By doing so, the farthest node from the sender has higher transmission priority.

Interest rebroadcasting is used also as implicit acknowledgment for the sender S. In case no rebroadcasting is overheard, then S will retransmit the packet up to 7 times before giving up.

Data packets follow the PIT entries back to the consumer(s) and they can be potentially cached by any node in the content delivery path to serve future requests.

3. Literature overview

In NDN, the caching system is composed of two main parts: the *caching decision algorithm*, which identifies the contents to be cached, and the *replacement policy*, which selects the contents to be replaced when the CS is full.

The default caching algorithm in VNDN is Cache Everything Everywhere (CEE), where every incoming Data is indiscriminately cached. Packets are replaced according to the Least Recently Used (LRU) policy, which organizes the items in order of use and removes the one that has not been used for the longest amount of time.

The work in [22] compares, in a urban vehicular scenario, the CEE policy against No Caching, Random Caching and some notable coordinated caching schemes like Leave Copy Down and Move Copy Down. Surprisingly, results show that, due to the dynamics of the scenario, CEE performances are comparable to those obtained with the other schemes. Moreover, the authors show that, despite its simplicity, LRU replacement outperforms Least Frequently Used (LFU) and Random replacements.

To adapt the caching decision to the peculiarities of the broadcast mobile environment, multiple strategies have been proposed in the literature. For instance, mobility-aware caching approaches, such as the ones in [11,23], leverage vehicular mobility prediction algorithms to identify the cachers (RSUs or vehicles) and the contents to be stored. Other recent works, e.g., [24], apply deep learning models to formulate optimal caching strategies. These approaches, however, introduce a not negligible complexity in the system and may require the exchange of multiple signaling messages. This can become an issue in real VNDN implementations, where caching operations must be performed on-line, i.e., during the forwarding process, and network conditions rapidly change due to the high mobility of vehicles.

To simplify and speed up the caching process, autonomous probabilistic schemes have been proposed that consider, among other features, the content popularity as crucial decision metric [9,25]. However, these approaches are designed for static information and do not consider the fact that many vehicular contents can be characterized by an application-specific *lifetime*. For instance, the time validity of road traffic information (e.g., mean speed in a given road segment) or parking lots availability information span a few seconds or minutes [26,27].

To deal with transient contents, NDN defines in the Data packet header the so-called *Freshness Period* field, which can be set by the producer to indicate, in seconds, the lifetime of the generated content. A replacement policy that honors content freshness is also defined, where Data packets are erased from the CS when their *Freshness Period*, treated like a timeout, expires. Freshness-based replacement is supported in addition to a standard replacement policy like LRU. Therefore, in principle, transient packets can be removed before their lifetime expires.

In addition to popularity and lifetime, another crucial caching decision metric for VNDN is the content availability that, as previously introduced, allows to improve the caching diversity. Contents that are estimated to be available and closer to the vehicle should be cached with lower probability, since it is likely that other caching vehicles will reply with the requested contents to consumers. The three metrics have been already considered in the caching literature, but their combined effect has not been discussed and evaluated in vehicular scenarios. In the following, per each one of the considered metrics, we introduce some notable research works.

3.1. Popularity-based solutions

A large body of literature considered content popularity to drive the caching decision in wired and wireless information-centric networks [9]. Indeed, by preferably caching more popular contents, it has been shown that the cache hits at intermediate routers increase, thus reducing the content retrieval delay, the load of requests arriving to the original source and hence, reducing also the amount of traffic crossing the network to reach the source and to carry Data back to the consumers.

A pioneering NDN autonomous caching scheme based on popularity is Most Popular Contents (MPC) in [28]. It assumes that every NDN node maintains locally the number of received Interests for each content name, and stores the pair {Content Name; Popularity Count} into a Popularity Table. If the number of requests for a given content name reaches locally a fixed popularity threshold, the content is marked as popular and it is cached. To further improve the hitting rate and adapt the caching threshold dynamically, a new strategy called Dynamic Fine-Grained Popularity-based Caching (D-FGPC) has been defined in [29]. D-FGPC assumes that non-popular contents are stored if there is space in the CS and it periodically adjusts the popularity threshold according to the frequency of received Interests and the available cache capacity. The threshold decreases when the cache capacity increases to allow caching more data and quickly fill up the available memory. Vice versa, the threshold increases when the number of distinct requests increases to improve the filtering of the popular contents. The PopCache strategy in [30], instead, implements a popularity-based decision according to the position of the caching nodes in the network. The most popular contents are stored in the nodes closer to the consumers, thus guaranteeing that the majority of requests are served in very short time. The unpopular contents are cached in the rest of the nodes to reduce the load on the origin producer.

The work in [25] applies the popularity-based caching at the RSUs to pre-fetch the highly requested contents and reduce the retrieval delay for vehicles passing nearby. Vice versa, the work in [31] enables popularity-based caching at vehicles in a hybrid autonomous/coordinated way. When a vehicle becomes congested by the increasing requests for a certain popular content, it explicitly asks the neighbors for caching that content. This allows distributing the traffic load on distinct nodes. In [32], instead, the strategy called WAVE leverages a newly defined *caching bit* in the Data packet header to guide the distributed caching decision. When a node identifies a content as popular, it sets the bit to recommend the caching operation to the downstream neighbor.

3.2. Lifetime-based solutions

A pioneering work analyzing the feasibility of caching in presence of transient contents is [33]. The authors focus on a static NDN network environment and demonstrate that, even in presence of highly transient contents, caching can be extremely useful to improve the network performance. They also propose a caching strategy that takes decision according to a cost function based on two contradictory parameters, the freshness cost and the communication cost. The freshness cost is the lowest when the data is retrieved directly from the producer, while the communication cost is the highest in the same condition and reduces when the data is fetched from the cachers closer to the consumer. The impact of lifetime in the caching decision has been also studied in static NDN IoT environments [34–36], where sensor nodes define a caching probability that is proportional to the residual lifetime of the Data packet. The higher the residual lifetime the higher the caching probability. Conversely, in VNDN, the topic is still at its very infancy and only a very few works dealing with this topic can be found in the literature. A cache replacement strategy is defined in [13] that considers the *Freshness Period*, in addition to content popularity and distance between the content provider and the cacher, to select the packets to evict from the CS. A similar scheme, but for RSUs only, is deployed in [14].

These works, however, apply the CEE algorithm and do not consider the effects of lifetime in the caching decision process. In our previous work [12], we designed a simple autonomous scheme called Freshness-Driven Caching (FDC), which caches with higher probability the contents with longer residual lifetimes. Contents are classified into two categories, long- or short-lasting, and the distinction is done by setting a dynamic threshold value, obtained as the exponential weighted moving average (EWMA) of the lifetimes carried by the received Data packets. Long-lasting contents are cached with a probability equal to 1, while short-lasting contents are cached with a probability equal to the ratio between their lifetime and the threshold.

3.3. *Availability-based solutions*

A pioneering caching proposal centered about the availability metric is *Hamlet*, in [37]. It is designed for wireless ad hoc networks (not based on NDN) where *only consumer nodes* cache packets according to the so-called *content presence index*, a parameter in the range $[0, 1]$ that is inversely proportional to the *hop count distance* to the content providers. Value zero means that the content was not detected in an observation window; while value one means that a one-hop far away provider of that content has been sensed. In case multi-hop far away providers are available, the index is between 0 and 1. Basically, the lower the content presence index, the higher the caching probability. The hop count distance is inferred when receiving the packets and it is updated over time according to a weighted smoothing average. By doing so, Hamlet improves the content diversity and allows to resolve content requests by limiting the number of multi-hop transmissions, which is crucial in a wireless broadcast network to reduce the content retrieval latency and improve the overall network performance. Indeed, although broadcast storm mitigation techniques can be enabled, e.g., [6,38], packet redundancy and congestion cannot be completely suppressed, due to vehicles' mobility or hidden terminal phenomena.

Similarly to Hamlet, but focusing on a wired network environment, the ProbCache strategy in [39] aims at maximizing the caching diversity by defining a metric based on the hop count. ProbCache caches contents probabilistically in order to fairly distribute the overall storage space among the different data flows sharing (part of) the same path. It tracks the number of hops that the packets travel, from the server to the consumer, and uses this information to estimate the caching capability of the path, under the assumption that all the on-path nodes have the same amount of storage space. The caching probability is then computed autonomously by each node by jointly considering the number of hops traveled by the packets and the amount of traffic served per unit time. Basically, the conceived probability is inversely proportional to the distance from the consumer and the potential cacher: the higher the closeness to the consumer the higher the chance of caching the packet. Moreover, the probability reduces if the incoming traffic increases.

4. Our proposal

4.1. *DANTE in a nutshell*

In this paper, we propose DANTE, a probabilistic strategy implemented in each vehicle that autonomously decides whether caching contents according to a composite metric that estimates their popularity, lifetime, and availability in the vehicular neighborhood.

In our design, the *neighborhood* of a target vehicle v identifies the network nodes (vehicles or RSUs) that can be reached by the packets sent by v through single-hop or multi-hop transmissions. Of course, depending on the mobility of the nodes, the neighborhood of a vehicle can largely vary over time. According to [4], to avoid the excessive spreading of requests in VNDN, Interests have an hop count limit equal to 5, consequently limiting also the hops traversed by Data packets. Therefore, in this paper the neighborhood of v covers the nodes that are at most 5 hops far away.

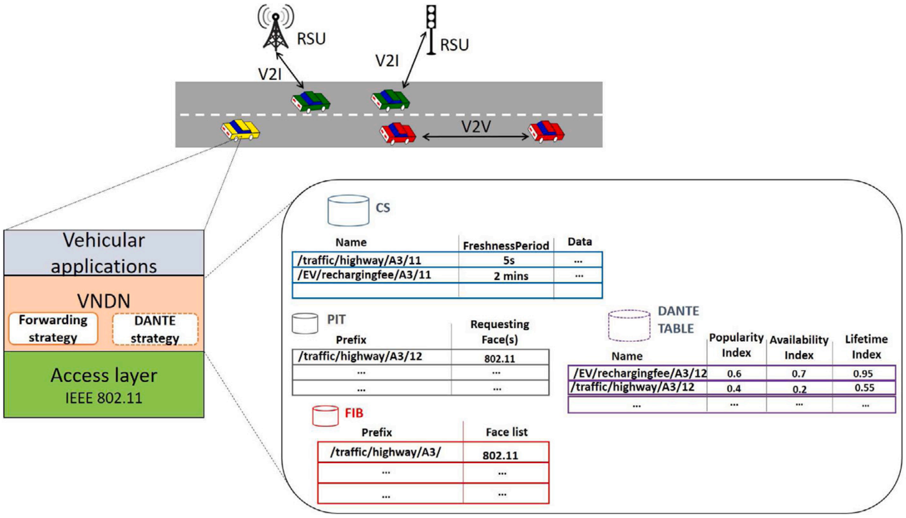
The idea of DANTE builds upon the Hamlet proposal [37] and advances it in the VNDN environment. In particular, we propose a caching strategy revolving around the following main design principles:

- **Caching with higher probability long-lasting popular contents**. DANTE uses the information about content lifetime and popularity, captured by nodes according to the received NDN packets, in order to privilege caching *popular long-lasting* contents, which can likely serve more requests during their lifetime, and do not waste precious caching resources. Jointly accounting for the two parameters is a novelty compared to the literature that instead either targeted popularity only, while being agnostic of the content lifetime, or limited the usage of the lifetime information in the replacement strategy.
- **Caching with higher probability the contents which are less available in the neighborhood.** In order to limit the data redundancy which could occur if all the nodes cache the same popular contents, in DANTE, the nodes estimate the content availability and the caching probability reduces as closer the content is. We integrate and extend the concept of content presence index in the caching scheme of the VNDN architecture where, unlike Hamlet, both consumers and forwarders can perform caching operations at the reception of Data packets.
- **Caching with higher probability contents which are leaving the neighborhood.** Due to the dynamics of the vehicular environment, which leads to varying network topology and volatile connectivity, the content presence index, expressed only in terms of number of hops as in [37], may not be sufficient to take the caching decision. Indeed, vehicles may travel on the same road, but in opposite directions, and have a very limited contact opportunity between each other. Hence, DANTE also considers the travel direction of the content providers to perform a wiser caching decision. Each candidate cacher checks whether the content provider is expected to travel along the same direction or not, and a higher caching probability is considered in the latter case.
- **Per-packet autonomous caching decisions.** Caching decisions, applied on a per-packet basis as required by VNDN, are autonomous and do not need any explicit coordination between vehicles, which is difficult to maintain in dynamic networks. Therefore, DANTE departures from the existing cooperative approaches, e.g., [40], where additional signaling is envisioned to coordinate caching operations.

4.2. *System model*

We consider the reference VNDN architecture as graphically sketched in Fig. 1.

Each VNDN node, either vehicle or RSU, runs the VNDN protocol stack. The implemented forwarding strategy is the one described in



Name	FreshnessPeriod	Data
/traffic/highway/A3/11	5s	...
/EV/rechargingfee/A3/11	2 mins	...

Prefix	Requesting Face(s)
/traffic/highway/A3/12	802.11
...	...
...	...

Prefix	Face list
/traffic/highway/A3/	802.11
...	...
...	...

Name	Popularity Index	Availability Index	Lifetime Index
/EV/rechargingfee/A3/12	0.6	0.7	0.95
/traffic/highway/A3/12	0.4	0.2	0.55
...	...	...	...

Fig. 1. Reference scenario (dashed modules are the ones specifically conceived by DANTE).

Section 2 and briefly summarized here. Vehicles, acting as consumers, broadcast Interest packets over an IEEE 802.11 access interface to retrieve contents by name. Vehicles or RSUs receiving the requests can act as providers or forwarders. To limit redundancy and collisions, packets broadcasting is deferred, with Data having higher transmission priority (i.e., lower defer times) than Interests [41], and overhearing with suppression is applied. The conceived caching strategy is, however, independent from the implemented forwarding scheme and it can work also with other approaches.

The same data structures foreseen in the legacy NDN, i.e., CS, PIT, FIB, are kept by each VNDN node. In addition, each VNDN node maintains the so-called DANTE Table that records the information needed to the caching decision, as it will be clarified in the following.

We consider a catalog of transient vehicular contents as $C = \{c_1, \dots, c_k\}$, with each content c_k composed of one or more Data packets $d_1^{c_k}, \dots, d_n^{c_k}$. Here, we focus on relatively high volatile contents whose lifetime may span from a few seconds to a few minutes, like parking lots availability [26] or real-time traffic information [27].

Each Data packet $d_n^{c_k}$ carries the *Freshness Period* (FP) set by the original producer, as in the legacy NDN, and the following additional fields that are used by the caching scheme (Fig. 2):

- *Timestamp*, $TS(d_n^{c_k})$, set by the original producer, which identifies the time the transient packet has been originated, and allows to calculate its residual lifetime. Time coordination is ensured by the fact that all vehicles maintain strict synchronization with the Coordinated Universal Time (UTC) that can be acquired from the Global Navigation Support System (GNSS) [42].

- *HopCount*, $H(d_n^{c_k})$, which is set equal to zero by the provider (i.e., original producer or cacher) and increased of one unit, hop by hop, at each receiver. We recall that, since the Interest hop count limit is set to 5 [4], Data packets can cover at most 5 hops.

- *KinematicsInfo*, $K(d_n^{c_k})$, which includes two kinematics information (position, in terms of latitude and longitude coordinates, and driving direction) of the provider carrying the content, as given by the on-board positioning system.

4.3. The caching decision

As in the legacy NDN implementation, when receiving a Data packet $d_n^{c_k}$, a vehicle, either a consumer or a forwarder, first checks its PIT. If a name matching is found, the vehicle decides whether caching the packet or not according to a caching probability, taking into account three content attributes, namely *lifetime index*, *availability index* and *popularity index*. The computation of the attributes is performed autonomously at each node, by monitoring the received Interest and Data packets. The resulting information is maintained in the DANTE Table, which records, per each requested content name, the three indexes. The DANTE Table can be reset on a trip basis.

If the PIT matching fails, the Data packet is unsolicited and it can be discarded. However, the packet is still useful to the caching policy to estimate the availability of that content in the neighborhood and its freshness. Therefore, before discarding the packet, the vehicle accesses the packet header and extracts the relevant information that is needed to fill the DANTE Table.

In the following, we describe how the three indexes are calculated and taken into account in the caching probability.

4.3.1. Lifetime index

This parameter reflects the transient nature of packet $d_n^{c_k}$ and allows each potential cacher to determine if $d_n^{c_k}$ is long-lasting or short-lasting with respect to the previously received packets.

Similarly to our proposal in [12], when a Data packet, $d_n^{c_k}$, traverses a node, the lifetime index, $L(d_n^{c_k})$, is computed as:

$$L(d_n^{c_k}) = min\left\{1, \quad \frac{RL(d_n^{c_k})}{Th_L}\right\}, \tag{1}$$

where:

- $RL(d_n^{c_k})$ is the residual lifetime of $d_n^{c_k}$, computed starting from the fields carried by the packet itself, as follows:

  $$RL(d_n^{c_k}) = FP(d_n^{c_k}) + TS(d_n^{c_k}) - t, \tag{2}$$

  where t is the time instant in which the vehicle receives the Data packet.
- Th_L is the current value of the so-called *lifetime threshold*, which is dynamically updated by each potential cacher at the reception of each Data packet, independently from the outcome of the caching decision, and tracks the average residual lifetime of the received packets. More specifically, Th_L is obtained as EWMA of the RL values of the previously received Data packets, as follows:

  $$Th_L = (1-\gamma)Th_{L,old} + \gamma \cdot RL(d_x^{c_y}), \tag{3}$$

  where: *(i)* parameter $\gamma \in (0,1)$ is set to 0.125 to avoid large fluctuations in the estimation and give more relevance to the historical values in front of the instantaneous ones [12], *(ii)* $Th_{L,old}$ is the previous value of the lifetime threshold, *(iii)* $RL(d_x^{c_y})$ is the residual lifetime of the previously received Data packet.[1]

---

[1] We use the subscripts x and y to indicate a generic previously received Data packet $d_x^{c_y}$, which may belong to a content, c_y, different from c_k.

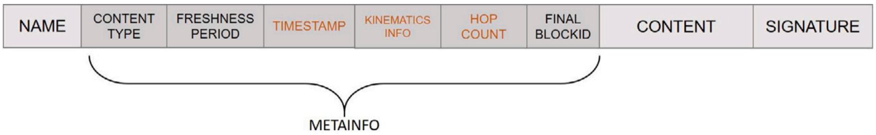


Fig. 2. Structure of the overhauled NDN Data packet (fields with text in orange are those added by DANTE). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Eq. (1) states that, if the residual lifetime of packet $d_n^{c_k}$ is larger than the current lifetime threshold, then the packet is labeled as long-lasting and the index is set to 1. Otherwise, the packet is labeled as short-lasting and $L(d_n^{c_k})$ is set to a value lower than 1.

After the caching decision is taken, regardless of its outcome, the node updates the lifetime threshold with the new value $RL(d_n^{c_k})$. At the reception of a subsequent Data packet, the novel value of the lifetime threshold will be used for the caching decision.

4.3.2. *Availability index*

The content availability index, A^{c_k}, accounts for the presence of content c_k in the neighborhood of a vehicle. Being each content generally composed of several Data packets, sometimes distributed and replicated in different nodes, this attribute is a collection of the contributions from the single packets, expressed in terms of hop count. More specifically, at the reception of Data packet $d_n^{c_k}$, the node checks the hop count $H(d_n^{c_k})$ carried in the packet and computes the packet availability index as:

$$A(d_n^{c_k}) = \delta \frac{1}{H(d_n^{c_k})}. \tag{4}$$

According to Eq. (4), the higher the hop count the lower the availability of the packet. In addition, each potential cacher, N, weights the hop count with the smoothing factor δ (with $\delta \in [0,1]$), set according to its driving direction compared to the one of the content provider, P (as conveyed in the Data packet).

It should be set differently in the two following cases: *(a)* when P and N move in the same direction and *(b)* when P and N move in divergent directions. In particular, we have:

$$\delta = \begin{cases} 1 & if \quad same \quad direction \\ 0.75 & \qquad\qquad otherwise \end{cases} \tag{5}$$

More details about the specific setting of the parameter δ are reported in Section 4.4.1.

Then, the node updates the content availability index A^{c_k} according to the following EWMA formula:

$$A^{c_k} = (1-\beta)A^{c_k}_{old} + \beta \cdot A(d_n^{c_k}), \tag{6}$$

with the smoothing factor $\beta \in (0,1)$, set to 0.8 to better reflect the actual content availability in the time-varying neighborhood and smooth out past observations of the content availability index, which get quickly outdated due to the volatility of vehicular topology. It is worth to observe that the availability index is not used to infer network topology or per-neighbor state information, as instead is done by several caching strategies deployed in more static network environments, e.g., [39]. It only serves to estimate if contents are available in the neighborhood and how close to the vehicle.

4.3.3. *Popularity index*

Each node observes all the Interests passing through itself to estimate the content popularity.

Given a content c_k, the popularity index for that content, F^{c_k}, is defined as the ratio between the current number of received requests for c_k, R^{c_k}, and the maximum number of requests received for a content of the catalog:

$$F^{c_k} = \frac{R^{c_k}}{max_{\forall c_i \in C}(R^{c_i})}. \tag{7}$$

Eq. (7) sorts contents according to their popularity as locally perceived by each node: $F^{c_k} = 1$ if c_k is the most requested content, whereas it is lower than 1 for contents receiving a lower number of requests.

4.3.4. *Caching probability and replacement policy*

A node (either a consumer or a potential forwarder) receiving a Data packet $d_n^{c_k}$, caches it with a probability $p_c(d_n^{c_k})$, which is calculated as follows:

$$p_c(d_n^{c_k}) = \left[1 - (1 - F^{c_k}) \cdot A^{c_k}\right] \cdot L(d_n^{c_k}). \tag{8}$$

Eq. (8) ensures that the caching probability gets smaller as the lifetime index decreases. If the content is long-lasting, i.e., $L(d_n^{c_k}) = 1$, the caching probability depends only on the popularity and availability indexes. In particular, popular contents that are also located (or moving) far away from the potential cacher, i.e., when $F^{c_k} \to 1$ and $A^{c_k} \to 0$, are cached with higher probability. Vice versa, the probability reduces when the content popularity decreases and the availability index increases. Only in presence of the most popular and long-last content(s), i.e., $F^{c_k} = 1$ and $L(d_n^{c_k}) = 1$, the caching probability is equal to 1.

The conceived probability *is simple on purpose* because of the objectives of *(i)* ensuring a quick and low-overhead computation, *(ii)* supporting a distributed and autonomous decision, without the need of explicit coordination (and hence, signaling) between vehicles or per-neighbor state information, *(iii)* enucleating how the metrics interact and their specific individual and joint effects.

In summary, the caching probability lets vehicles cache with a lower probability those contents which are estimated to be easily reachable in their neighborhood, while the probability increases in the opposite case. A direct consequence of this behavior is the improvement of the *caching diversity* in the vehicular neighborhood of the VNDN nodes. Contents are not cached redundantly and distinct ones can be stored by distinct vehicles, implying that a content request can be more likely satisfied nearby with no need to resort to the original source. At the same time, DANTE well trades off between the popularity and availability indexes, by guaranteeing that the availability of popular contents in the network is not compromised.

When a node caches a Data packet $d_n^{c_k}$ in the CS, it sets a *validity timeout* equal to the residual lifetime $RL(d_n^{c_k})$. When the timeout expires, $d_n^{c_k}$ is removed from the CS. We call this policy *RL-based eviction.*

However, if a new packet is selected to be cached and the CS is full, then an existing one must be replaced before it expires. In DANTE, the RL-based eviction is integrated with the traditional LRU replacement which, despite its simplicity, is highly effective even in vehicular environments [22].

4.4. *Early results*

Before analyzing the performance of the proposal against selected benchmark schemes under realistic vehicular settings, in the following we report preliminary results aiming at *(i)* tuning the smoothing factor δ and *(ii)* disclosing how the combination of the three devised indexes affects the caching probability.

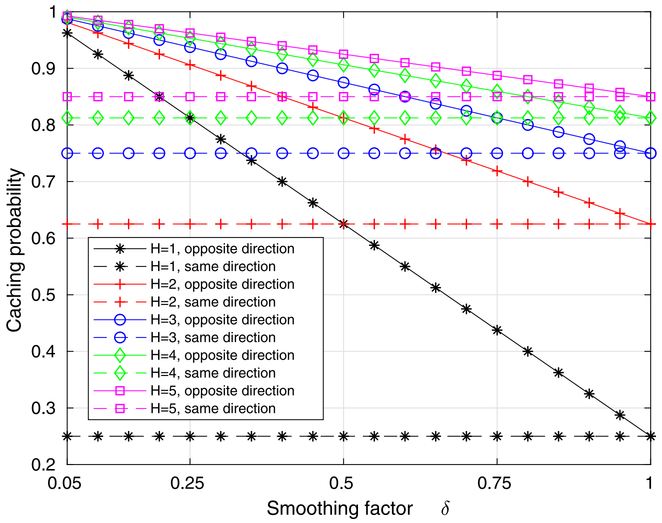


Fig. 3. Caching probability when varying the δ parameter, for different hop count (H) values (popularity index set to 0.25, long lasting content).

4.4.1. *Tuning the δ parameter*

The rationale behind the usage of the smoothing factor δ in Eq. (6) is to specifically account for the fact that the number of hops information must be complemented with further information in order to estimate more precisely the availability of a given content in a time-varying neighborhood. In particular, δ has the role of differently weighting the availability index by considering the direction of movement of a provider, P, of a content packet with regard to a receiving/forwarding node N. Specifically, the δ setting should ensure that the following conditions are met that take into account both the hop distance between P and N and their directions of movement:

- If P and N follow divergent directions, it is reasonable to expect that the provider carrying the content is leaving the neighborhood of N, hence the related availability index should be scaled down, so that the caching probability gets higher. We can summarize such a requirement as follows:

$$\delta: \quad p_c(P \leftarrow, N \rightarrow) > p_c(P \rightarrow, N \rightarrow), \tag{9}$$

  being the arrows indication of the moving directions of P and N. This requirement translates in δ different than 1 if P and N move in divergent directions.
- The availability index computed for the content carried by a provider P_a at a hop distance from node N equal to h_a and moving in divergent direction compared to N should be *always higher* than the availability index computed for the content carried by a provider P_b at a hop distance $h_b = h_a + 1$ and moving in the same direction. Correspondingly, the caching probability for the content carried by provider P_a should be *always lower* than the caching probability for the content carried by provider P_b. This constraint can be formulated as follows, where H is the hop distance between the providers and N:

$$\delta: \quad p_c(P_a \leftarrow, N \rightarrow, H = h_a) < p_c(P_b \rightarrow, N \rightarrow, H = h_a + 1). \tag{10}$$

To justify the specific setting of δ, we report in Fig. 3 the caching probability computed as in Eq. (8), as a function of δ for hop distance values, H, from 1 to 5, as typically considered in VNDN environments. Solid lines refer to the case when P and N move in divergent directions, dashed lines to the case of nodes moving in the same direction. Results show that setting δ in the range [0.75, 1) ensures that the aforementioned second condition is met for all values of the hop distance H.

For instance, let us consider the blue solid line corresponding to the caching probability for the case in which the provider P_a and the receiving node N move in opposite directions and the hop count H is equal to 3. For values of δ lower than 0.75, this curve is always above the green dashed curve corresponding to a provider P_b with an hop count H equal to 4 and moving in the same direction as N. The same holds for all the other hop count values. This is not the desired behavior. Values of δ in the range [0.75,1) are only acceptable to satisfy the requirement in Eq. (10) above. Therefore, among values in the restricted range [0.75,1) we opt for the lower bound of the range, i.e., 0.75 and used this value in the conducted simulations.

4.4.2. *Understanding the caching probability*

To figure out how the three conceived indexes affect the caching probability, we performed a preliminary simulation campaign in Matlab, when considering contents requested according to a Zipf-based popularity distribution [43].

Fig. 4 reports the caching probability when vehicles move in the same direction and the availability index is set to different values aimed to reflect an hop count ranging from 1 to 5 (as shown in the plots). The lifetime index is varied to account for a content lifetime in the range [0–150] s, while the popularity index is in the range [0–1]. Two different settings are considered for the Zipf's skewness parameter α, i.e., 0.4 and 1.

It can be clearly observed that the farther the content (the greater the hop count of a received packet), the higher the caching probability. This is even more true as the residual lifetime of the content increases. For contents which are, instead, available in the one-hop neighborhood the caching probability is close to zero, regardless of the content lifetime.

Since Zipf-distributed content requests are assumed, for α equal to 1, the popularity index results very low with the exception of a few contents, which are the most requested ones. In particular, there are a few very highly requested contents (a few caching probability points are shown corresponding to high popularity index values), whereas other contents receive less requests (caching probability points are concentrated in correspondence with low popularity index values). With α passing from 1 to 0.4, the caching probability points are reported spanning different popularity index values. In other words, the number of requests is more uniformly distributed. Moreover, the number of points increases because a higher number of distinct contents from the catalog is requested.

For contents with a residual lifetime longer than the average, the number of hops from the content provider, as expressed by the content availability index, is the parameter mainly affecting the caching probability.

The previous statements hold when the content provider and the candidate cacher move in the same direction. If it is not the case, the caching probability gets higher as shown in Fig. 5.

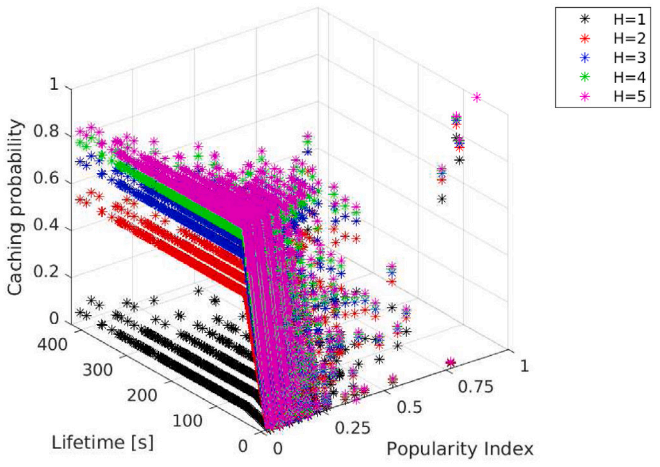
Fig. 6, instead, reports the caching probability for different settings of availability and popularity indexes, when considering only long-lasting contents. It can be clearly observed that the caching probability is equal to one in presence of the most popular content(s) and reduces when the popularity index decreases or the availability index increases.

Hence, the achieved trends confirm the capability of the conceived caching probability to target the objectives of the designed caching policy.

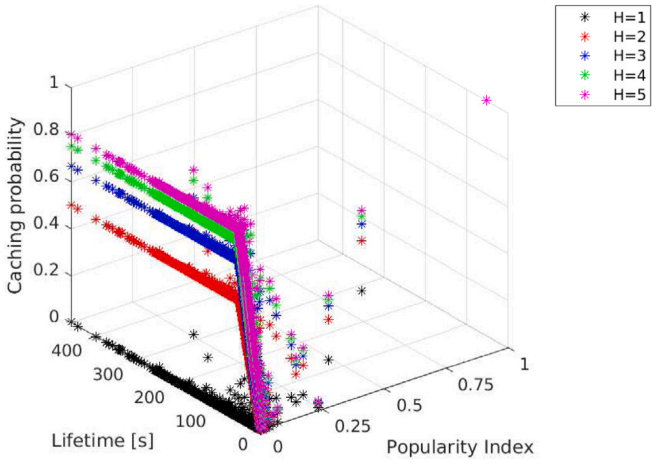
5. Performance evaluation

5.1. *Simulation settings and tools*

Extensive simulations have been conducted in ndnSIM, the official NDN simulator [18]. We properly overhauled the simulator to implement the proposal and the considered benchmark schemes. We consider two distinct vehicular scenarios, where vehicles move according to the Simulation of Urban MObility (SUMO) model [44].

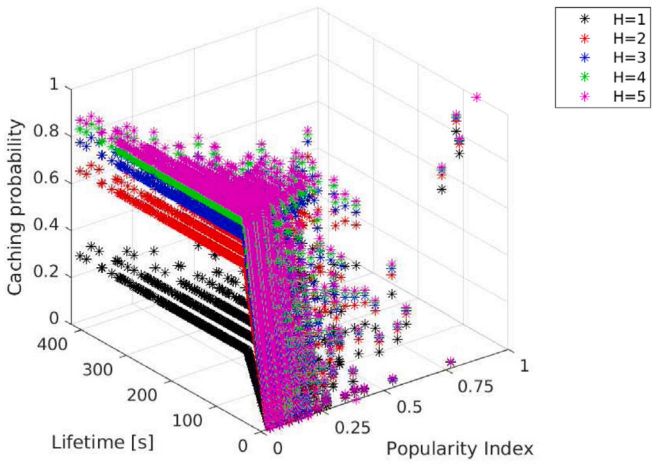


(a) α=0.4.

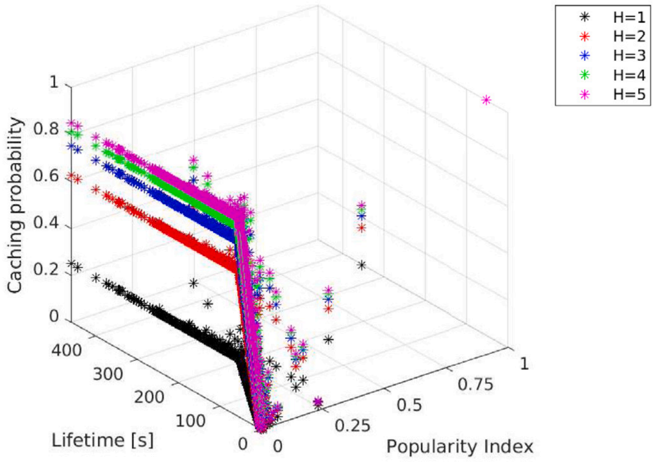


(b) α=1.

Fig. 4. Caching probability for different values of lifetime, popularity and availability indexes (Zipf distribution, $|C| = 10^6$, 20 000 content requests, content lifetime exponentially distributed with mean $\mu = 50$ s), when the content is provided by a node following the same direction as the receiving node for different α settings. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) α=0.4.



(b) α=1.

Fig. 5. Caching probability for different values of lifetime, popularity and availability indexes (Zipf distribution, $|C| = 10^6$, 20 000 content requests, content lifetime exponentially distributed with mean $\mu = 50$ s), when the content is provided by a node following a divergent direction compared to the receiving node, for different α settings. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- The first one is a Manhattan grid of size 1 km^2 with 2 lanes per direction, where 100 vehicles move at speeds ranging between 10 and 20 km/h. An RSU deployed in the middle of the topology is the only original producer of contents.
- As a second scenario we consider a 4 km-long highway road segment, where 100 vehicles move at a maximum speed of 90 km/h. One RSU deployed along the highway, in the middle of the topology, is the only original content producer.

Vehicles and RSUs implement IEEE 802.11p as access layer technology and interact through the broadcast transmissions of Interest/Data packets according to the VNDN [4] forwarding strategy, described in Section 2.

We consider the *Freshness Period* of contents uniformly varying in the range [10 – 200] s to match a vehicular data traffic pattern with heterogeneous transient contents. These latter exhibit different lifetimes to reflect the temporal scope of different application types, ranging from traffic assistance to infotainment [15,26,27].

A variable number of vehicles, ranging from 20 to 80, request contents according to the Zipf popularity distribution [43], with skewness parameter $\alpha = 1$. At the beginning of the simulation, the CSs of vehicles are empty but, after receiving and caching Data packets, they can act as providers, in their turn.

The main simulation settings are reported in Table 1.

5.2. Benchmark schemes and metrics

We compare the proposed scheme against three distinct autonomous benchmark schemes: *(i)* CEE, which represents the legacy caching strategy implemented in NDN, *(ii)* the Freshness-Driven Caching (FDC) [12], which implements freshness-aware decisions but it is oblivious of the content popularity and availability, *(iii)* a baseline popularity-based caching scheme [9], labeled as POP in the following, which uses the popularity as decision metric but it is not aware of the content freshness and availability, *(iv)* the topology-based ProbCache scheme [39], which is oblivious of popularity and lifetime information but, similarly to our

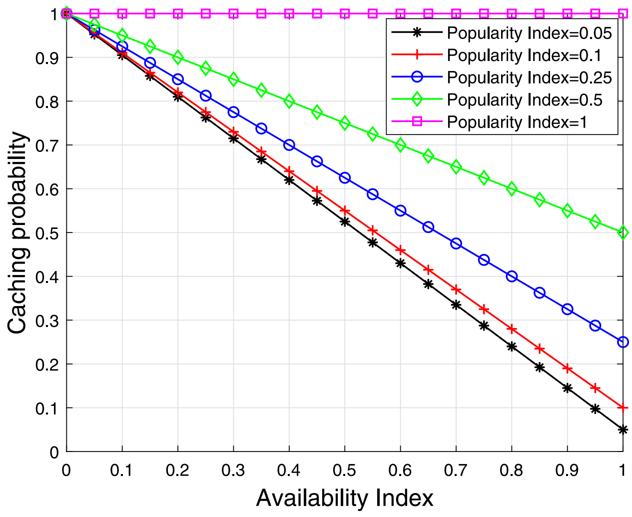


Fig. 6. Caching probability for different values of the content popularity index and availability index (Zipf distribution, $\alpha = 1$, $|C| = 10^4$, 200 content requests, long-lasting content).

work, it improves the caching diversity by considering the number of hops that the packets travel from the data source to the consumer.

While FDC aims at caching the long-lasting contents, POP aims at caching the most popular ones and ProbCache aims at diversifying the contents cached by on-path nodes.

For the sake of evaluation, the following metrics have been derived:

- Cache hits: it is computed as the ratio, in percentage, between the received Interests satisfied by the local CS and the total number of received Interests.
- Content retrieval delay: it is computed as the average time for retrieving a content.
- Number of hops: it is computed as the average number of hops traveled by the Interest packets for retrieving the requested content since the first Interest is issued.

Table 1
Main simulation settings.

Parameter	Value
Content catalog size	10 000 contents
Content size	100 Data packets
Data packet size	1000 bytes
Content popularity	Zipf-distributed with $\alpha = 1$
Content lifetime	Uniformly distributed in $[10 - 200]$ s
Propagation	Nakagami fading
Scenario	Urban topology (1 km^2 grid) Highway topology (4 km-long road segment)
Number of vehicles	100
Number of consumers	20–80
Number of producers	1 RSU
Caching capacity	5% of the content catalog

- Number of Interest packets: it captures the total number of Interest packets broadcasted by vehicles in the simulation.
- Interest retransmissions ratio: it is the average number of Interests retransmitted by consumers normalized over the expected number of Interests needed to retrieve the overall content.
- Number of CS replacements: it indicates the number of replacement events in the CS of the simulated vehicles.

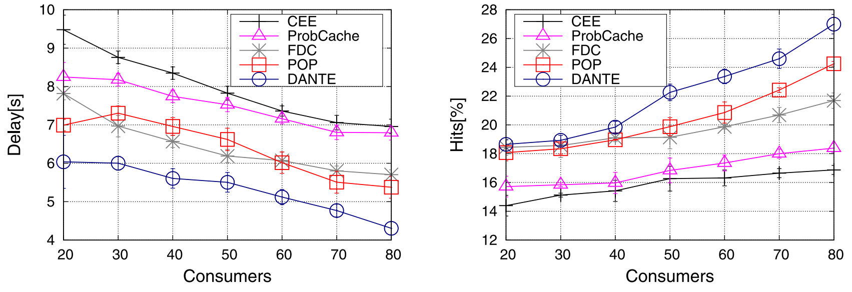
The first three metrics measure the effectiveness of the compared protocols, whereas the others allow to figure out their efficiency. In addition, it is worth to observe that higher cache hits translate into a reduction of the requests arriving at the original producer (the RSU in our scenarios). Moreover, a reduction in the hop number and retrieval delay implies that more distinct contents are available closer to the consumers. Therefore, these metrics also reflect the ability of the caching strategy to improve the caching diversity and, in turn, to reduce the caching redundancy.

Results are averaged over ten independent runs and reported with 95% confidence intervals.

5.3. Results

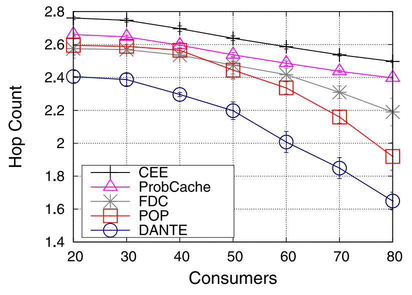
5.3.1. Urban scenario

Results in Figs. 7 and 8 refer to the urban scenario.



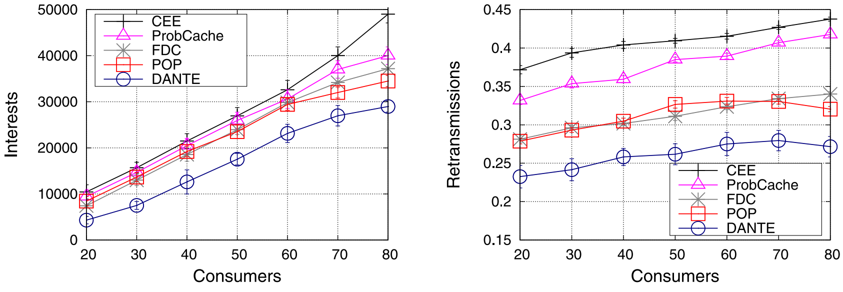
(a) Content retrieval delay.

(b) Cache hits.

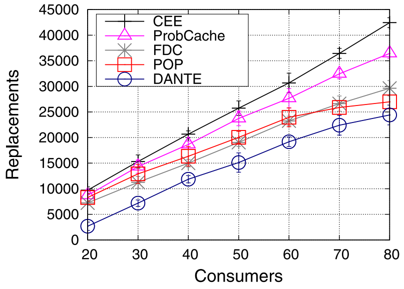


(c) Number of hops.

Fig. 7. Effectiveness metrics in the Manhattan scenario, when varying the number of consumers.



(a) Number of Interest packets. (b) Interest retransmissions ratio.



(c) Number of CS replacements.

Fig. 8. Efficiency metrics in the Manhattan scenario, when varying the number of consumers.

It can be clearly observed that, as expected, the CEE scheme exhibits the poorest performance for all the considered metrics. By allowing all packets to be cached indiscriminately, the strategy tends to host homogeneous contents in the CSs of neighbor vehicles. This translates in the highest content retrieval delay (see Fig. 7(a)), the lowest cache hit ratio (see Fig. 7(b)) and the highest number of hops (see Fig. 7(c)). CEE is also the least efficient scheme in terms of network resources usage. It generates the highest load of Interest packets. This is especially true as the number of consumers increases, confirming its poor scalability, Fig. 8(a). Moreover, it triggers a significantly higher number of retransmissions compared to the other schemes, under all settings, Fig. 8(b). The simplicity of the strategy is also paid in terms of the highest number of cached content replacements (see Fig. 8(c)), since it always happens that, when a new Data packet arrives and the CS of the vehicle is full, a replacement operation must be performed.

Similarly to CEE, also ProbCache performs poorly compared to the other schemes. Indeed, ProbCache implements topology-based decisions that depend on the forwarding path and are oblivious of lifetime and popularity information. The mechanism works well in static wired topologies with non-transient contents, where the routers can properly multiplex their storage space between competitive flows. Conversely, in VNDN, forwarding paths are highly dynamic and Data packets may travel across different vehicles depending on their ephemeral ad hoc connections. Therefore, it is difficult for a vehicle to take the right decisions with the ProbCache policy.

For lower number of consumers, the other two probabilistic-based benchmark schemes exhibit similar performance, although for different reasons. Indeed, FDC caches long-lasting contents with higher probability, while POP privileges the most popular ones. In both cases, however, the selective decision improves the performance compared to CEE and ProbCache. As the number of consumers increases, the POP strategy outperforms FDC, by experiencing a shorter delay, a higher cache hit ratio and a lower number of hops (see, respectively, Figs. 7(a)–7(c)). Such trends are due to the fact that under the simulated settings, due to the Zipf distribution, a higher number of requests, from different consumers, concentrate on the same few contents, the most popular ones. Hence, such a strategy achieves better performance by prioritizing the caching of popular contents.

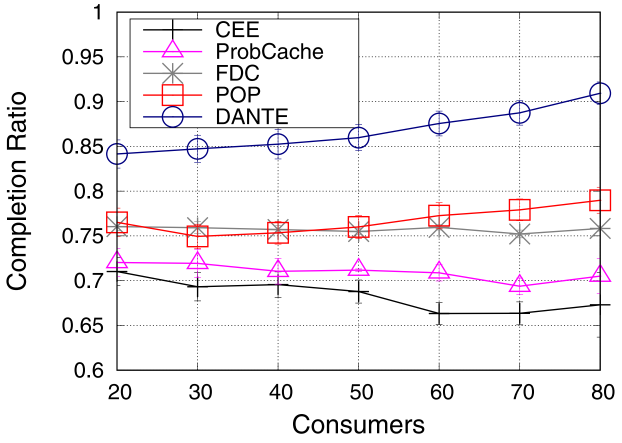
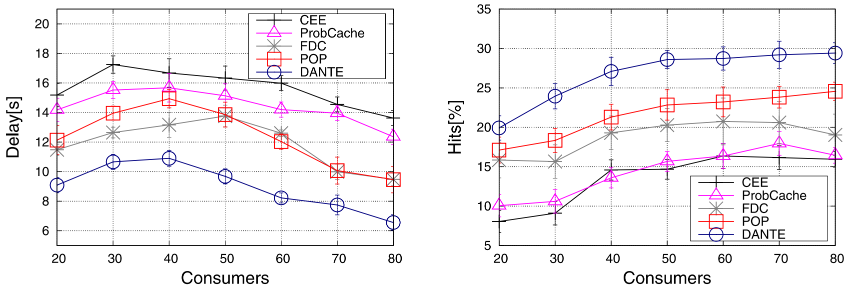


Fig. 9. Content completion ratio in the highway scenario, when varying the number of consumers.

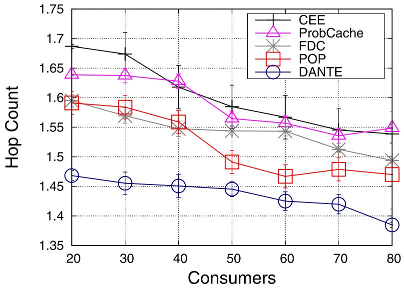
The proposed DANTE outperforms the benchmark schemes under all the considered settings. This achievement confirms that accounting for the content popularity, availability and lifetime allows to take more judicious content caching decisions. By improving the caching diversity, with DANTE, it is more likely to find a requested content in the CS of a neighbor vehicle and, hence, to retrieve it with a shorter latency, with no need to contact the original producer. Such improvements also translate in less wastage of network resources for transmitting Interest packets.

5.3.2. Highway scenario

The highway scenario encompasses a wider topology and more dynamic connectivity conditions than the urban one: vehicles can experience longer disconnection periods or not stable contacts and some of them could not be able to complete the content retrieval within the simulation time (400s). Therefore, in addition to the previous metrics, we compute the Content Completion Ratio, which identifies the average portion of vehicles that are able to complete the content retrieval

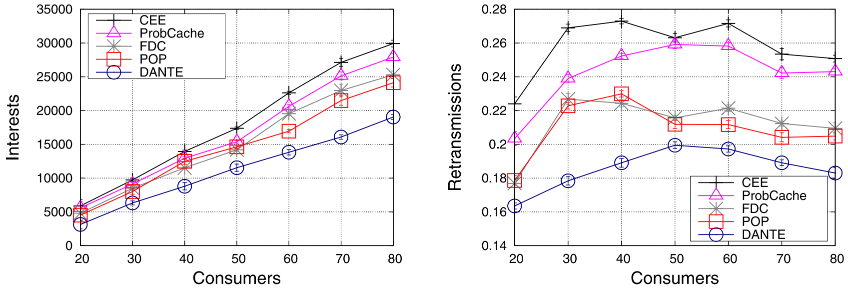


(a) Content retrieval delay. (b) Cache hits.

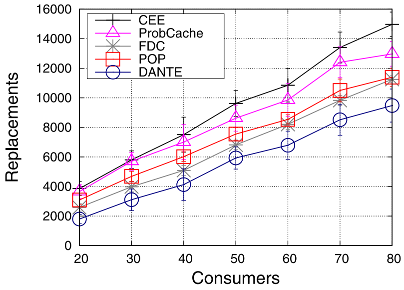


(c) Number of hops.

Fig. 10. Effectiveness metrics in the highway scenario, when varying the number of consumers.



(a) Number of Interest packets. (b) Interest retransmissions ratio.



(c) Number of CS replacements.

Fig. 11. Efficiency metrics in the highway scenario, when varying the number of consumers.

within the simulation time. Fig. 9 shows that DANTE, compared to the benchmark schemes, allows to a larger set of vehicles (between 85%–90%) to complete the content retrieval, thus demonstrating how crucial can be the role of a wiser caching strategy in highly dynamic vehicular environments. Moreover, performance also better scales with the number of consumers compared to the benchmark schemes.

Figs. 10 and 11 show, respectively, the effectiveness and efficiency metrics for the considered schemes, when varying the number of consumers. The same trends observed for the urban scenario hold here, with DANTE outperforming the benchmark schemes under all the considered settings.

Compared to the urban scenario, a higher delay for all the considered schemes can be observed in Fig. 10(a). It is mainly due to the fact that, when vehicles start to retrieve a content, there are no guarantees that a stable path towards a provider exists. Similarly, the hop count is lower than the one observed in the previous case: it is around $1.6-2.8$ in the urban scenario, while it is in the range $1.4-1.7$ in Fig. 10(c). This is due to the fact that it is difficult to maintain multi-hop connectivity in a highway scenario and the content retrieval occurs when the vehicle is closer to a provider.

Moreover, vehicles are scattered in a wider topology and there are less opportunities for packet collisions. This is why the number of

Table 2
Overhead analysis: DANTE Vs. benchmark schemes.

Strategy	Additional fields (size)	Overhead per Data packet	Additional Tables (footprint)
CEE	–	–	–
ProbCache	HopCount (2 bytes)	0.18%	–
FDC	Timestamp (4 bytes)	0.35%	–
POP	–	–	Popularity Table (480 kB)
DANTE	Timestamp (4 bytes) HopCount (2 bytes) KinematicsInfo (24 bytes)	2.6%	DANTE Table (640 kB)

Interest packets, retransmissions and cache replacements in Fig. 11 are generally lower than the ones observed in the urban scenario.

5.4. Overhead assessment

Table 2 summarizes the main differences between the compared schemes in terms of incurred overhead. DANTE entails three additional fields to be carried in Data packets. Such fields incur an additional overhead per Data packet. In particular, by assuming a legacy VNDN header with an average content name length of 40 bytes and a 4 bytes-long signature [45,46], the additional overhead is around 2.6%. It is worth observing that nonetheless such small additional overhead per Data packet, in DANTE the number of exchanged Interest packets is significantly reduced compared to the CEE scheme and also smaller than the one experienced by other lighter benchmark solutions. Hence, overall, DANTE is more efficient than the compared schemes in terms of exchanged bytes in the network.

Moreover, DANTE leverages a table tracing, per each requested content, the envisioned indexes, which represent, in a cumulative fashion, its popularity, availability and transient nature.

A rough estimation of the DANTE Table storage footprint is provided, for a content catalog size $|C| = 10^4$ and a content name length set to 40 bytes [45,46], when considering three additional 8 bytes-long fields for the three envisioned indexes. The resulting storage footprint of 640 kB is quite negligible, especially for vehicular devices. Moreover, such an estimation is an upper bound, since it considers the whole catalog, whereas only requested contents are tracked in practice. Vice versa, CEE and FDC do not maintain any per-content information, while ProbCache leverages only the hop count information. POP only traces the content popularity in a popularity table [28] and the estimated overhead is around 480 kbytes for the same settings as before.

6. Conclusions

In this paper we presented DANTE, a novel caching strategy specifically designed for VNDN that privileges caching popular contents with a longer lifetime, not available in the neighborhood. Such a choice is meant to better use the storage space and limit the number of packet replacements forced by the content lifetime expiration, while improving content diversity.

The proposal is shown to outperform three representative benchmark solutions in two different vehicular scenarios. DANTE succeeds in reducing the content retrieval delay, thanks to the higher cache hit, as well as the number of exchanged Interests. Such improvements are achieved at the expenses of a few additional bytes per packet and of keeping an additional light data structure.

As a future work, we plan to consider a more sophisticated estimation of the availability index, by letting a potential cacher dynamically set δ as a function of the residual contact time with the provider from which the content is received.

Moreover, the investigation of the benefits of DANTE in a vehicular cloud scenario supporting both caching and computing services will be a further research direction.

CRediT authorship contribution statement

Marica Amadeo: Conceptualization, Methodology, Software, Validation, Investigation, Writing - review & editing. **Giuseppe Ruggeri:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - review & editing. **Claudia Campolo:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - review & editing. **Antonella Molinaro:** Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, F.R. Yu, A survey of mobile information-centric networking: Research issues and challenges, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2353–2371.
[2] L. Zhang, et al., Named data networking, ACM SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73.
[3] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, A. Ksentini, Named data networking in vehicular ad hoc networks: State-of-the-art and challenges, IEEE Commun. Surv. Tutor. 22 (1) (2019) 320–351.
[4] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, VANET via Named Data networking, in: IEEE INFOCOM WKSHPS, 2014, pp. 410–415.
[5] M. Amadeo, C. Campolo, A. Molinaro, CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks, IEEE Commun. Lett. 16 (9) (2012) 1380–1383.
[6] S.H. Ahmed, S.H. Bouk, M.A. Yaqub, D. Kim, M. Gerla, CONET: Controlled data packets propagation in vehicular named data networks, in: IEEE CCNC, 2016, pp. 620–625.
[7] X. Liu, Z. Li, P. Yang, Y. Dong, Information-centric Mobile Ad Hoc Networks and content Routing: a survey, Ad Hoc Netw. 58 (2017) 255–268.
[8] M. Meisel, V. Pappas, L. Zhang, Ad Hoc Networking via Named Data, in: Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture, 2010, pp. 3–8.
[9] M.A. Naeem, M.A.U. Rehman, R. Ullah, B.-S. Kim, A comparative performance analysis of popularity-based caching strategies in Named Data Networking, IEEE Access 8 (2020) 50057–50077.
[10] G. Deng, L. Wang, F. Li, R. Li, Distributed probabilistic caching strategy in VANETs through named data networking, in: IEEE INFOCOM WKSHPS, 2016, pp. 314–319.
[11] N. Abani, T. Braun, M. Gerla, Proactive caching with mobility prediction under uncertainty in information-centric networks, in: Proceedings of the 4th ACM Conference on Information-Centric Networking, 2017, pp. 88–97.
[12] M. Amadeo, C. Campolo, G. Ruggeri, G. Lia, A. Molinaro, Caching transient contents in Vehicular Named Data Networking: a performance analysis, MDPI Sensors 20 (7) (2020) 1985.
[13] S. Ostrovskaya, O. Surnin, R. Hussain, S.H. Bouk, J. Lee, N. Mehran, S.H. Ahmed, A. Benslimane, Towards multi-metric cache replacement policies in vehicular Named Data Networks, in: IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2018, pp. 1–7.
[14] Z. Wei, J. Pan, K. Wang, L. Shi, Z. Lyu, L. Feng, Data forwarding and caching strategy for RSU Aided V-NDN, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2019, pp. 605–612.
[15] E. Lee, E.-K. Lee, M. Gerla, S.Y. Oh, Vehicular cloud networking: architecture and design principles, IEEE Commun. Mag. 52 (2) (2014) 148–155.
[16] S. Zhang, H. Luo, J. Li, W. Shi, X. Shen, Hierarchical soft slicing to meet multi-dimensional QoS demand in cache-enabled vehicular networks, IEEE Trans. Wireless Commun. 19 (3) (2020) 2150–2162.
[17] J. Zhang, K.B. Letaief, Mobile Edge Intelligence and computing for the Internet of Vehicles, Proc. IEEE 108 (2) (2019) 246–261.
[18] S. Mastorakis, A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM 2.0: A new version of the NDN simulator for NS-3, NDN, Technical Report NDN-0028.
[19] IEEE Std. 802.11-2012: IEEE Standard for Information Technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.
[20] G. Naik, B. Choudhury, J.-M. Park, IEEE 802.11 bd & 5g NR V2X: Evolution of radio access technologies for V2X communications, IEEE Access 7 (2019) 70169–70184.
[21] G. Grassi, D. Pesavento, G. Pau, L. Zhang, S. Fdida, Navigo: Interest forwarding by geolocations in Vehicular Named Data Networking, in: IEEE WoWMoM, 2015, pp. 1–10.
[22] F.M. Modesto, A. Boukerche, An analysis of caching in information-centric Vehicular Networks, in: IEEE International Conference on Communications (ICC), 2017, pp. 1–6.

[23] L. Yao, A. Chen, J. Deng, J. Wang, G. Wu, A cooperative caching scheme based on mobility prediction in vehicular content centric networks, IEEE Trans. Veh. Technol. 67 (6) (2017) 5435–5444.

[24] Y. Guo, Z. Ning, R. Kwok, Mobility prediction based vehicular edge caching: A deep reinforcement learning based approach, in: IEEE ICCT, 2019, pp. 1120–1125.

[25] R. Ding, T. Wang, L. Song, Z. Han, J. Wu, Roadside-unit caching in Vehicular Ad Hoc Networks for efficient popular content delivery, in: IEEE Wireless Communications and Networking Conference (WCNC), 2015, pp. 1207–1212.

[26] Y. Atif, S. Kharrazi, D. Jianguo, S.F. Andler, Internet of Things Data Analytics for parking availability prediction and guidance, Trans. Emerg. Telecommun. Technol. 31 (5) (2020) e3862.

[27] Y. Yu, B. Li, Real-time Traffic Data management for dynamic vehicle navigation system, in: 17th International Conference on Geoinformatics, 2009, pp. 1–5.

[28] C. Bernardini, T. Silverston, O. Festor, MPC: Popularity-based caching strategy for content centric networks, in: IEEE ICC, 2013, pp. 3619–3623.

[29] M.D. Ong, M. Chen, T. Taleb, X. Wang, V. Leung, FGPC: Fine-Grained popularity-based caching design for content centric networking, in: ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2014, pp. 295–302.

[30] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urushidani, S. Yamada, PopCache: cache more or less based on content popularity for Information-Centric Networking, in: IEEE Conference on Local Computer Networks, 2013, pp. 236–243.

[31] W. Quan, Y. Liu, X. Jiang, J. Guan, Intelligent popularity-aware content caching and retrieving in highway vehicular networks, EURASIP J. Wireless Commun. Networking 2016 (1) (2016) 1–10.

[32] K. Cho, M. Lee, K. Park, T.T. Kwon, Y. Choi, S. Pack, WAVE: popularity-based and collaborative in-network caching for content-oriented networks, in: IEEE INFOCOM Workshops, 2012, pp. 316–321.

[33] S. Vural, N. Wang, P. Navaratnam, R. Tafazolli, Caching Transient Data in Internet content routers, IEEE/ACM Trans. Netw. 25 (2) (2016) 1048–1061.

[34] M.A. Hail, M. Amadeo, A. Molinaro, S. Fischer, Caching in named data networking for the Wireless Internet of Things, in: IEEE RIoT, 2015, pp. 1–6.

[35] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, How to cache in ICN-based IoT environments?, in: IEEE/ACS International Conference on Computer Systems and Applications, 2017, pp. 1117–1124.

[36] O. Hahm, E. Baccelli, T.C. Schmidt, M. Wählisch, C. Adjih, L. Massoulié, Low-power Internet of Things with NDN & cooperative caching, in: ACM Conference on Information-Centric Networking, 2017, pp. 98–108.

[37] M. Fiore, F. Mininni, C. Casetti, C.-F. Chiasserini, To Cache or not to Cache?, in: IEEE INFOCOM, 2009, pp. 235–243.

[38] S.H. Ahmed, S.H. Bouk, M.A. Yaqub, D. Kim, H. Song, J. Lloret, CODIE: controlled Data and Interest evaluation in Vehicular Named Data Networks, IEEE Trans. Veh. Technol. 65 (6) (2016) 3954–3963.

[39] I. Psaras, W.K. Chai, G. Pavlou, Probabilistic in-network caching for information-centric networks, in: Proceedings of the second edition of the ICN workshop on Information-centric networking, 2012, pp. 55–60.

[40] W. Huang, T. Song, Y. Yang, Y. Zhang, Cluster-based cooperative caching with mobility prediction in Vehicular Named Data Networking, IEEE Access 7 (2019) 23442–23458.

[41] M. Amadeo, C. Campolo, A. Molinaro, Enhancing content-centric networking for vehicular environments, Comput. Netw. 57 (16) (2013) 3222–3234.

[42] K.F. Hasan, C. Wang, Y. Feng, Y.-C. Tian, Time synchronization in vehicular ad-hoc networks: A survey on theory and practice, Veh. Commun. 14 (2018) 39–51.

[43] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, et al., Web Caching and Zipf-like Distributions: Evidence and implications, in: IEEE INFOCOM, 1999, pp. 126–134.

[44] D. Krajzewicz, Traffic Simulation with SUMO–Simulation of Urban MObility, in: Fundamentals of Traffic Simulation, Springer, 2010, pp. 269–293.

[45] A.V. Ventrella, G. Piro, L.A. Grieco, Publish–subscribe in mobile information centric networks: Modeling and performance evaluation, Comput. Netw. 127 (2017) 317–339.

[46] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, B. Liu, Fast name lookup for named data networking, in: IEEE 22nd International Symposium of Quality of Service (IWQoS), 2014, pp. 198–207.

Marica Amadeo is an assistant professor at University Mediterranea of Reggio Calabria. She received a master degree (2008) in telecommunications engineering from the University Mediterranea of Reggio Calabria, and a Ph.D. degree in 2013 from the same University. Before her current appointment, she was a Post-Doc Researcher at the University Mediterranea of Reggio Calabria (2013-2019). Her major research interests are in the field of information-centric networking and edge computing.

Giuseppe Ruggeri received the master degree in electronics engineering in 1998 from the University of Catania, Italy. In 2002 he received the Ph.D. in electronics, computer science and telecommunications engineering from the University of Palermo, Italy. He is currently assistant professor at the University Mediterranea of Reggio Calabria. His current interests include self organizing networks, Internet of Things, Social Internet of Things.

Claudia Campolo is an associate professor of telecommunications at University Mediterranea of Reggio Calabria, Italy. She received a Laurea degree in Telecommunications Engineering (2007) and a Ph.D. degree (2011) from the University Mediterranea of Reggio Calabria, Italy. Before her current appointment, she was an Assistant Professor at the University Mediterranea of Reggio Calabria (2012–2020). Her main research interests are in the field of vehicular networking and 5G systems.

Antonella Molinaro is an associate professor of telecommunications at the University Mediterranea of Reggio Calabria. Previously, she was an assistant professor with the University of Messina (1998–2001) and the University of Calabria (2001–2004), and a research fellow at the Politecnico di Milano (1997–1998). She was with Telesoft, Rome (1992–1993) and Siemens, Munich (1994–1995). Her current research focuses on 5G, vehicular networking and future Internet architectures.