Optimal Content Prefetching in NDN Vehicle-to-Infrastructure Scenario

Giulia Mauri, Mario Gerla, *Fellow, IEEE,* Federico Bruno, Matteo Cesana, *Member, IEEE,* and Giacomo Verticale, *Member, IEEE*

***Abstract*—Data replication and in-network storage are two basic principles of the Information Centric Networking (ICN) framework in which caches spread out in the network can be used to store the most popular contents. This work shows how one of the ICN architectures, the Named Data Networking (NDN), with content pre-fetching can maximize the probability that a user retrieves the desired content in a Vehicle-to-Infrastructure scenario. We give an ILP formulation of the problem of optimally distributing content in the network nodes while accounting for the available storage capacity and the available link capacity. The optimization framework is then leveraged to evaluate the impact on content retrievability of topology- and network-related parameters as the number and mobility models of moving users, the size of the content catalog and the location of the available caches. Moreover, we show how the proposed model can be modified to find the minimum storage occupancy to achieve a given content retrievability level. The results obtained from the optimization model are finally validated against a Name Data Networking architecture through simulations in ndnSIM.**

***Index Terms*—Information Centric Networking, Named Data Networking, Vehicular Network, Integer Linear Programming, Content Store, ndnSIM**

I. Introduction

Today's Internet is focused on content retrieval instead of point to point communication. In addition, users are mostly mobile, continuously changing their location. One of the main problem of user mobility is the intermittent connectivity that causes loss of packets. An Information Centric Networking (ICN) framework is the solution to the previous problems. Indeed, the communication is based on a request/response model where the focus is on the content. This is the basis of all the ICN frameworks, where the communication paradigm shifts from retrieving the content from a given location to retrieving the content from the network.

The user does not care any longer where the packets come from, but rather wants them as soon as possible. Thus, an end node that wants a given content issues an `INTEREST` message with the name of the desired content. Then, the network is responsible for choosing the best node to fetch the content from by means of a `DATA` packet. In particular, in the NDN paradigm, every node can be potentially the best source of a given content for a given user (content consumer). Indeed, the widespread use of caching into the network nodes is one of the foundations of the NDN protocol.

This paradigm blends well with specific issues of user mobility, such as intermittent connectivity and changing topologies. Let us consider what happens during a handover and a user is downloading a video from You Tube in a moving car. As the car moves, the Access Point (AP), to which the user is connected, is no longer available and a new connection should be established seamlessly. Thus, handover procedures are used to keep track where the user is, that is, which mobile base station or access point she is currently attached, in order to re-route the video chunks to the proper access network device. Differently, in an NDN scenario, when a consumer moves from one location to another, a seamless handover is easily achieved by reissuing the interest messages from the new location for the subsequent video chunks, and the network can then deliver the content from the best source for the new location. This means that the `DATA` packet can be sent from whichever network node that has the content in its local cache.

We focus here on exploiting the NDN paradigm in the context of vehicular networks. Namely, we analyze the beneficial effects of pre-fetching contents at static network nodes on the performance of Vehicular-to-Infrastructure (V2I) communication paradigms. In our setting, vehicular users move along pre-defined paths and get in touch with several Access Points (APs) along their travel. The users entering the area served by the NDN network ask for a content from a *content catalog*. Each content is segmented in chunks, so the vehicular user can retrieve different parts of the content from different Access Points. A content is successfully retrieved if the user moves out of the area with all the chunks of a content.

We define the problem of optimally pre-fetching the content chunks in the network nodes in order to maximize the average content retrieval probability. We show that the problem can be formulated as an Integer Linear Programming (ILP) problem which can be extended/modified to encompass the case where the minimum retrieval proabability (out of all the mobile users) is maximized. The proposed formulations are then leveraged to study the impact of several parameters on the retrieval probability, such as the number of users in the system, the AP available bandwidth, the propagation latency, and the size of the available caches which are called throughout the paper *Content Stores*. We also introduce another model that returns the optimal storage size of the whole network in order to

A preliminary version of this paper appears in F. Bruno, M. Cesana, M. Gerla, G. Mauri and G.Verticale, "Optimal Content Placement in ICN Vehicular Networks", in the *5th International Conference on Network of the Future*, Sept 2014.

Matteo Cesana has been partially supported in this work by the Italian Ministry for Education, University and Research (MIUR) through the national cluster project MIE-Mobilità Intelligente Ecosostenibile (grant number: CTN01-00034-594122)

reach a desired retrieval probability. Then, we also compare our results with theoretical results and we show the efficiency of our solution. Finally, we compare the performance of the optimal pre-fetching solution against the popular Least Recency Used (LRU) reactive caching policy, by means of simulations with ndnSIM.

Thus, the main contributions of this work are:

- We demonstrate the advantages in using a pre-fetching policy in a V2I scenario with respect to a dynamic policy.
- We evaluate various network topologies and we study the impact of several network parameters.
- We consider also the optimal storage size of the whole network.
- We validate our model assumptions and complexity by means of real vehicular traces.

The paper is structured as follows: Section II provides an overall view of the related work. The Section III reviews some of the basic concepts of the Name Data Networking (NDN) architecture which is taken as reference in this work. Section IV explains the scenario considered in this paper and the behavior of the NDN-enabled nodes. The optimization models for content placement are presented in Section V. The numerical results are shown in Section VI. Conclusions are left for the last Section VIII.

II. Related Work

Mobile ICNs have recently attracted much attention within the research community. The works [1] and [2] provide comprehensive surveys on ongoing research in mobile ICN. Both papers highlight the benefits that different ICN designs (e.g. DONA [3], CCN [4], NetInf [5], NDN [6]) provide to content and producer mobility. In details, the first work mainly focuses on open challenges that should be addressed, whereas the second explains how to exploit the named-data paradigm for making the mobility management easier. In particular, [1] addresses the issues of producer mobility and dynamic routing, further considering the benefits in the management of user multihoming and handover. Differently, [2] evaluates the performance improvement due to in-network caching and the adaptability to various scenarios. Whilst the aforementioned work is not scenario-specific, the ICN paradigm is applied to specific VANET scenarios in [7], and [8]. The latter proposes a geo-based forwarding strategy and studies caching to avoid redundancy problem. Three heuristic strategies have been suggested and evaluated showing an improvement in network performance. While, the paper [7] shows how the adoption of NDN could bring benefits to vehicular communications by implementing a prototype of Vanet NDN.

In the field of Vehicular Networks, also the paper [9] raises the question whether CCN could be the solution for vehicular networks. The authors provide evidence that CCVN (Content Centric Vehicular Networking) performs better than the legacy TCP/IP-based architecture. Also this paper considers the advantages of using ICN for mobility support. The paper [10] proposes a mobility management scheme for CCN that confirms the previous conjectures. It evaluates the routing update latency and the delivery latency depending on the number of nodes. The results show that the mobile CCN scheme achieves better performance than the basic CCN protocol. Nonetheless, the work does not evaluate how to optimize content distribution in order to lower the perceived latency and to guarantee content retrievability.

Several papers present solutions to optimize content distribution by exploiting content prefetching for both ICN and non-ICN scenarios. A proactive caching algorithm for NDN is proposed in [11], where the main idea is to proactively ask and cache contents before the user moves from one access point to another. The simulations show that the proposed approach has better performance than the original NDN: lower handover cost, higher delivery ratio, and shorter handover latency. The solution is implemented modifying the `INTEREST` packet and the basic communication protocol. Differently than our work, no optimization model is given for content prefetching, but rather the proposed scheme is evaluated mainly through simulations.

The problem of optimizing content prefetching in “ classical” network scenarios has been largely debated in the past years. As an example, the paper [12] aims at reducing the content retrieval in the World Wide Web by predicting and prefetching those files that are most likely to be requested. The prefetching is done based on server advices to the client, who can choose to prefetch or not the suggested files; the model makes probabilistic predictions for prefetching based on received requests.

A predictive prefetching method is also presented in [13] for Wi-Fi networks. The paper develops new handoffs and data transfer strategies in order to reduce the connection setup latency and the download latency. The results show that the performance of the vehicular WiFi network could be improved using the proposed strategies. However, the authors leave some open challenges, for example how to implement the estimation mechanism for prefetching. Our work does not consider the handoff strategy because the ICN context does not need to establish end-to-end connections. Both the previous papers do not present an optimization model for content prefetching, but show that prefetching can help in performance improvement for fixed and mobile scenarios, while our paper considers a new optimization strategy for content prefetching in ICN.

Then, the paper [14] models the vehicular mobility using a fog-of-war probabilistic representation. The model is used to build a graph to formulate a non-integer linear programming optimization problem. The problem is to decide the amount of traffic to offload to the vehicular network. The output of the problem is used to take the content pre-fetching decisions. A speculation-based pre-fetching scheme is proposed in [15] to ensure the users’ quality of experience. Moreover, it guarantees, together with a new grouping-based storage strategy, a good data delivery efficiency and high lookup success rate. While, an interesting mobility similarity model is presented in [16] to enhance the data delivery efficiency. Users are grouped in a mobile community based on similar playback and movement. These three works provide an overview of the current research trend for content pre-fetching in V2I network scenario.

Finally, the authors of [17] analyze the optimal caching policy for file servers co-located with the APs in a WiFi-based content distribution community infrastructure. They formulate the content management problem as mixed integer programming with the aim of maximizing the file retrieval probability within a time interval. The model considers 100 content files whose popularity follows the Zipf's law with different skew parameters. Moreover, it uses 50 access points with a coverage of 250 meters. Our paper shares the same objective function of the aforementioned work but on the other hand, our optimization framework considers additional constraints which are specific to a multi-hop Named Data Networking (NDN) scenario.

III. Background on Named Data Networking

This section briefly reviews the core components of Named Data Networking [6] which constitute the reference networking paradigm for the present work.

NDN Packet Model

The request is contained in a INTEREST packet, while the response is in a DATA packet. An INTEREST packet carries the *Content Name*, the *Selector* and a *Nonce*. The DATA packet is composed of the *Content Name*, the *Signature*, the *Signed Info* and the *Data*. A DATA packet is the corresponding response of an INTEREST only if the *Content Name* in the INTEREST matches the prefix of the *Content Name* in the DATA packet. The names are typically hierarchical, so matching a prefix means that the *Content Name* in the DATA is in the subtree specified by the *Content Name* in the INTEREST.

NDN Node Model

Each node has three main data sets: the Content Store (CS), the Pending Interest Table (PIT) and the Forwarding Information Base (FIB). The Content Store is an associative container of data. Which data is stored in a node at a given time is decided by means of a CS management policy. The PIT stores the name prefixes that correspond to the INTEREST that the node could not satisfy and that it has sent to some other nodes; moreover, the node keeps track of the requesting nodes asking for the content, to send downstream the returned data. Finally, the FIB registers the prefixes and the corresponding list of neighbors to forward INTEREST packets.

NDN Communication Model

A node asks for a content by sending an INTEREST packet to all neighbors listed in the FIB for the matching prefix. After hearing the request, any node with the content responds with a DATA packet. When a node receives an INTEREST, it checks if there is a correspondence into its tables, i.e. CS, PIT and FIB. If the Content Store caches the requested DATA packet, the node sends out the content and drops the satisfied INTEREST. Otherwise, if the match is in the PIT, the corresponding entry is updated adding the requesting node and the INTEREST is discarded. If the match is in the FIB, the INTEREST is sent out to the next hop(s) and it is created a new entry in the PIT. Finally, if there is no match, the INTEREST is discarded because the node does not know how to find any matching DATA.

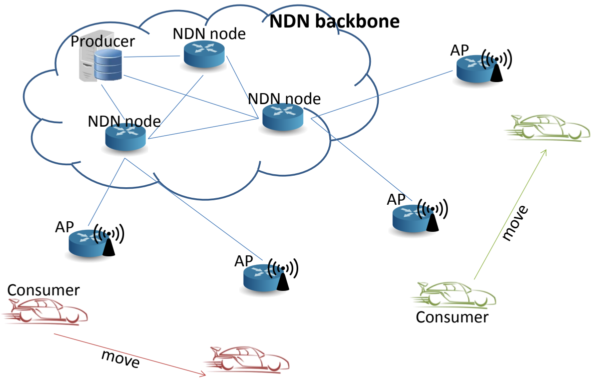


Fig. 1. The reference network scenario.

The DATA packet processing is quite similar, the node does a longest match lookup of the DATA packet *Content Name*, if there is a match in the Content Store, the node throws it away because it is a copy. Otherwise, the node looks in the PIT and if there is a match, it sends the data to the requesting nodes. In case of a reactive CS policy, the node adds the packet to the CS, possibly dropping some other content. A FIB match means an unrequested DATA, so the node drops the packet.

IV. A Vehicle-to-Infrastructure Scenario for NDN

This paper considers a Vehicle to Infrastructure (V2I) scenario composed of a set of Access Points (APs) to provide connectivity to moving vehicles, and a backbone network which interconnects the APs. See Figure 1.

The set of network nodes is identified by N where $\mathcal{I} \subset N$ is the set of Access Points. Each node in the network (vehicles, APs, backbone nodes) runs a simple version of the NDN protocol [6]. Namely, when a vehicle enters the area and connects to the first AP along its path, it issues an INTEREST message for the first chunk of content j from the content catalog, which is composed of S_j chunks, each of size $D^{(\mathrm{ck})}$. Each AP has a Content Store, capable of holding CS chunks. If the AP receiving the INTEREST message has the requested chunk of the object j, then it delivers it to the vehicle, otherwise it issues an INTEREST message upstream in order to retrieve the requested chunks. The same procedure is repeated by any node in the backbone. If they have the requested chunks, they send them downstream, otherwise they propagate the INTEREST upstream. The *Producer* node is a special node in the backbone which is assumed to have all the chunks of all the contents and always satisfies the requests. A vehicle keeps sending INTEREST messages as long as it receives chunks or it moves outside of the AP coverage area. When the vehicle connects to a new AP, it starts reissuing requests for the missing chunks until the content is fully received.

The reference scenario includes U_{tot} users moving at average speed s which may also change along the user's path. The overall mobility pattern is modeled as follows: each user

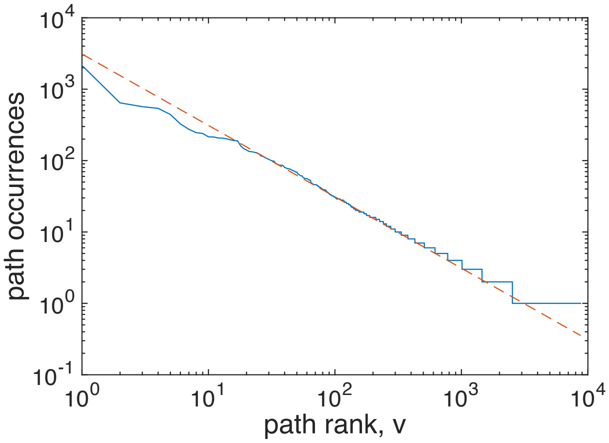


Fig. 2. Path frequency (log-log graph) in the considered area using real vehicular traces. The dashed line represents the Zipf's distribution with $\sim$ Zipf(1).

is randomly assigned to one moving path out of V possible paths. The v-th path has a probability β_v of being chosen that follows a Zipf's law with exponent α_p:

$$\beta_v \sim \text{Zipf}(\alpha_p)$$

To evaluate the feasibility of such path-assignment model and to verify the previous assumption in a practical case, we have analyzed real-life mobility traces of a medium/large urban environment available described in [18]. Namely, we have measured the path occurrence as a function of the path rank for the reference data set. As shown in Figure 2, the probability to choose a path (path occurrence) obey to the Zipf's law, since the log-log plot of the path occurrence is a function in the path rank.

Each moving path then determines the set and sequence of APs visited by the moving user along its way. The APs are assumed to have a transmission diameter d; consequently, the connection time available to one user at a given AP is defined as $T_{con} = d/s$. For the sake of presentation, the time needed to discover and associate to the APs is not considered; however, the model/scenario can be trivially extended by scaling down the T_{con} parameter by a factor which depends on the discovery/association time.

The binary parameter m_{iv} represents if a user is connected with the AP i along the path v. By letting m_{iv} be equal to 1 if a user along path v connects to AP i, and 0 otherwise, the number of users on each path can be written as $U_v = U_{tot} \cdot \beta_v$, while the average number of users connected to the ith AP per path v is:

$$U_i = \textstyle\sum_{v=1}^{V} U_v m_{iv} / \sum_{v=1}^{V} m_{iv}$$

The content catalog is composed of C content objects. The j-th content has a probability of being requested that follows the Zipf's distribution with exponent α_r, as mostly assumed by the literature [19], [20]. The probability that the jth content is requested is:

$$\sigma_j \sim \text{Zipf}(\alpha_r)$$

Let us define the set of links in the reference network $\mathcal{E}$; the available link bandwidth is c_e with $e \in \mathcal{E}$. While the radio channel has a bandwidth c^R. We further assume a generic but known routing pattern in the reference scenario, that is, for every node in the network it is known the set of edges (and ordered nodes) which constitute the shortest path to/from the reference node. Namely, it is defined the set SP_{ij} which includes the sequence of edges which belong to the shortest path from node i to node j.

The maximum number of content chunks which can be transferred over link e downstream towards the ith AP for the users in the vth path is called the Maximum Downloadable Burst (MDB), B_{eiv}, and depends on the available bandwidth, which must be shared among all the requests from the downstream nodes. Assuming that the bandwidth of the link is equally shared by all the users connected to the same AP i along the path v, we have:

$$B_{eiv} = \frac{c_e \cdot T_{iv}}{D^{(\text{ck})}} \Big(\sum_{i \in \mathcal{I} : e \in SP_{t(e)i}} U_i \Big)^{-1} \tag{1}$$

where $T_{iv} = T_{con} \cdot m_{iv}$ is the duration of the connection between AP i and the user along the path v, T_{con} is the total duration of the connection between the user and the APs and $SP_{t(e)i} \subset \mathcal{E}$ is the subset of links which belong to the shortest path towards access point i. The function $t(e)$ returns the tail of the edge e. Thus, given an edge e that is the directed link from node a to b, the function $t(e)$ returns the node a. While the Maximum Downloadable Burst associated to the radio channel, B^R_{iv}, towards the ith AP for users in the vth path is computed as follows:

$$B^R_{iv} = \frac{c^R \cdot T_{iv}}{D^{(\text{ck})} \cdot U_i} \tag{2}$$

It is also worth noting that retrieving chunks from a backbone node "closer" to the *Producer* node incurs in a transmission and processing overhead. The parameter $\eta_{eiv} \geq 1$ represents the ratio between the time needed to retrieve at AP i a content chunk for a user along path v through a backbone link e and the time to retrieve the same chunk if the Content Store is available at AP i. This factor can be computed as:

$$\eta_{eiv} = \frac{\sum_{f \in SP_{it(e)}} T^{(\text{ck})}_{ifv} + 2\tau \cdot |SP_{it(e)}|}{T^{(ck)R}_{iv} + 2\tau} + 1.$$

where $T^{(\text{ck})}_{iev} = D^{(ck)}/B_{eiv}$ is the transmission time for a single chunk using the available Maximum Downloadable Burst through link e and $T^{(ck)R}_{iv} = D^{(ck)}/B^R_{iv}$ is the corresponding Maximum Downloadable Burst through radio channel. A transmission latency τ accounts for the processing cost of the messages.

V. Optimal Content Placement

This section provides the Integer Linear Programming (ILP) formulation for the problem of content placement. Differently than the reference NDN protocol, we make the following assumptions:

1) The chunks in the Content Stores of the network nodes do not change over time according to a caching policy,

but are pre-fetched according to the decisions of an off-line management platform;

2) The content objects are protected with a Forward Error Correction code, which encodes a file of size S_j chunks in H chunks. The jth content can be fully reconstructed if the consumer obtains any S_j chunks. Therefore, the Consumer does not issue `INTEREST` messages for specific chunks, but issues general `INTEREST`s for additional chunks.

A. Maximizing the Content Retrievability

The optimization objective is to distribute content chunks into the Content Stores in order to maximize their availability for the retrieval to a user that moves around the network. Let us define the following sets:

- Access Points: $i \in \mathcal{I} = \{1, \ldots, I\}$
- Contents: $j \in \mathcal{J} = \{1, \ldots, C\}$
- Content Stores: $k \in \mathcal{K} = \{1, \ldots, K\}$
- Paths: $v \in \mathcal{V} = \{1, \ldots, V\}$
- Links: $e \in \mathcal{E} = \{1, \ldots, E\}$
- Shortest Paths: $s \in SP_{ij} = \{1, \ldots, N \times N\}$

For the sake of presentation, we assume in the following that $N = \mathcal{K}$, that is, all the nodes in the network (access points and backbone nodes) can cache contents. Moreover, we further assume that the content *Producer* is assigned index 0.

The proposed formulation leverages the decision variables and parameters resumed in the following Table I:

TABLE I
MODEL VARIABLES AND PARAMETERS

Variables	
$x_{jk} \in \mathbb{Z}$	number of chunks of content j cached into the content store k
$A_{jv} \in \{0, 1\}$	boolean variable that is 1 if the content j is retrievable along the path v, 0 otherwise
$y_{eivj} \in \mathbb{Z}$	number of chunks of content j retrieved through link e and meant for users associated to AP i along the path v
$z_{ivj} \in \mathbb{Z}$	number of chunks of content j retrieved through radio channel and meant for users associated to AP i along the path v
Parameters	
$\sigma_j \in [0, 1]$	probability of requesting content j
$\beta_v \in [0, 1]$	probability of choosing path v
$S_j \in \mathbb{Z}$	size of content j
$B_{eiv} \in \mathbb{Z}$	available MDB over link e towards the AP i for the users in the path v
$B^R_{iv} \in \mathbb{Z}$	available MDB over the radio channel toward the AP i for the users in the path v
$\eta_{eiv} \in \mathbb{Z}$	the highest cost in retrieving chunks from the furthest content store in the backbone network
$CS_k \in \mathbb{Z}$	capacity of content store k, measured in number of chunks

The problem of pre-fetching content at network content stores such that the probability of retrieving contents is maximized can be formalized as follows:

$$\max: \sum_{j \in \mathcal{J}} \sigma_j \sum_{v \in \mathcal{V}} \beta_v A_{jv} \tag{3}$$

s.t.

$$\sum_{i \in \mathcal{I}} z_{ivj} + \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{E}} y_{eivj} m_{iv} \geq A_{jv} S_j \qquad \forall j \in \mathcal{J}, \forall v \in \mathcal{V} \tag{4}$$

$$z_{ivj} + \sum_{f \in SP_{i0}} (\eta_{fiv} \cdot y_{fivj}) \leq B^R_{iv} \qquad \forall i \in \mathcal{I}, \forall v \in \mathcal{V}, \forall j \in \mathcal{J} \tag{5}$$

$$\sum_{f \in SP_{t(e)0}} y_{fivj} \leq B_{eiv} \qquad \forall i \in \mathcal{I}, \forall v \in \mathcal{V}, \forall e \in \mathcal{E}, \forall j \in \mathcal{J} \tag{6}$$

$$y_{eivj} \leq x_{jt(e)} \qquad \forall i \in \mathcal{I}, \forall v \in \mathcal{V}, \forall e \in \mathcal{E}, \forall j \in \mathcal{J} \tag{7}$$

$$z_{ivj} \leq x_{ji} \qquad \forall i \in \mathcal{I}, \forall v \in \mathcal{V}, \forall j \in \mathcal{J} \tag{8}$$

$$\sum_{j} x_{jk} \leq CS_k \qquad \forall k \in \mathcal{K} \tag{9}$$

$$y_{eivj} = 0 \qquad \forall i \in \mathcal{I}, \forall v \in \mathcal{V}, \forall j \in \mathcal{J}, \forall e \in \mathcal{E} : e \notin SP_{i0} \tag{10}$$

$$x_{jk} \geq 0, \qquad y_{eivj} \geq 0, \qquad z_{ivj} \geq 0, \qquad A_{jv} \in \{0, 1\} \tag{11}$$

The objective is to maximize the retrievability of the content j, A_{jv}, that is the probability of satisfying the request of the vehicular user. The objective function (3) depends on the probability of requesting the content j, σ_j and on the probability of choosing the path v, β_v. The first constraints (4) define the retrievability of a content. A content is retrieved if the overall number of chunks which are retrieved from any network node (AP and backbone node) is above the required threshold. Constraints (5) and (6) impose a limit on retrievability that depends on the maximum downloadable burst (MDB). In short, the overall number of chunks retrieved from network nodes closer to the *Producer* cannot exceed the capacity of the network links used to deliver them downwards. The parameter η_{eiv} emphasizes the fact that the further the chunks are in the network, the higher is the cost to retrieve them. The number of chunks cached in each content store constrains the maximum number of retrievable chunks, as enforced by constraints (7) and (8). Equations (9) introduce budget-type constraints on the maximum number of contents/chunks stored at any network node. Then, equations (10) impose to not retrieve chunks from links that are not comprised in the set of the shortest path between the considered access point and the producer node. Finally, the equations (11) define the decision variables of the formulation.

B. Maximizing the Worst Content Retrievability

The model introduced in the previous section maximizes the average content retrievability, thus fairness among different paths/users may be low. In this context, it is also worth enforcing a more fair solution by maximizing the "worst", i.e. the minimum, content retrievability out of all the mobile users. We introduce a new variable, ϵ, that we would like to maximize in our objective function. The value of this variable is determined from the first set of constraints (13), where the parameter β_v has been removed with respect to (3) in the original formulation in order to guarantee a fair comparison among the paths. This allow us to study the impact of caching on paths with different popularity.

Objective function:

$$\max\colon \epsilon \tag{12}$$

Constraints:

$$\sum_j \sigma_j A_{jv} \geq \epsilon \quad \forall v \in \mathcal{V} \tag{13}$$

$$x_{jk} \geq 0,\ y_{eivj} \geq 0,\ z_{ivj} \geq 0,\ A_{jv} \in \{0,1\}, \epsilon \in [0,1] \tag{14}$$

The constraints from (4) to (10) are unchanged. The variables and the indexes remain the same, as in Section V-A.

The objective function (12) maximizes ϵ that represents a probability; indeed, ϵ is the smallest success probability within the success probability associated to the different paths v in the network, deduced from the constraints (13). Equation (14) adds the information that ϵ should be within 0 and 1.

C. Minimizing the Total Size of Content Stores

The previous two formulations target the maximization of the content retrievability metric under a given budget in terms of available Content Store size. Here, our aim is to minimize the total size of content store in order to guarantee a desired success probability. We introduce a new parameter that is the minimum content retrievability P_{succ}.

Objective function:

$$\min\colon \sum_{j,k} x_{jk} \tag{15}$$

Constraints:

$$\sum_{j,v} \sigma_j \beta_v A_{jv} \geq P_{\text{succ}} \tag{16}$$

The constraints from (4) to (11) are unchanged, except for constraint (9) that is deleted. The variables and the indices remain the same, as in Section V-A.

The objective function (15) minimizes the sum of all the x_{jk} that represents the number of chunks stored in the network content stores. The new constraint (16) imposes a lower bound on the content retrievability.

D. On the Problem Size

For the sake of completeness, we provide the asymptotic complexity of the problem formulation expressed as a function of the number of variables and constraints. The complexity is similar for all the three models. The number of variables of the first model V-A is:

$$O(C \cdot I \cdot V \cdot E + C \cdot K)$$

While the number of constraints is:

$$O(K + I \cdot V \cdot C \cdot E)$$

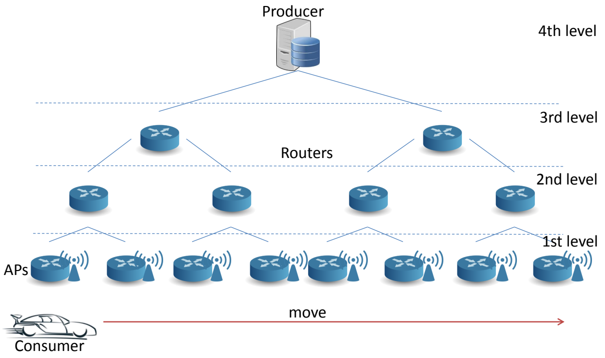


Fig. 3. The reference V2I scenario: the tree topology.

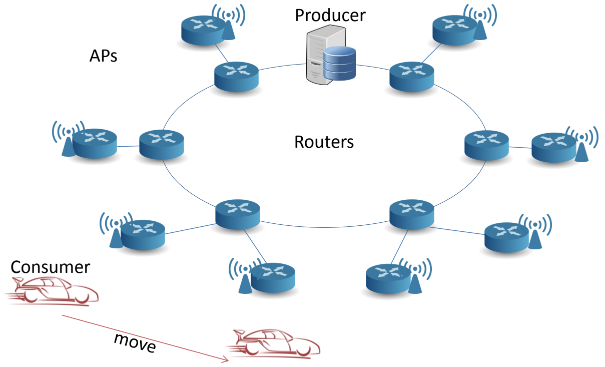


Fig. 4. The reference V2I scenario: the circle topology.

The second model V-B has an additional variable and $O(V)$ additional constraints. Finally, the third model V-C has the same number of variables and $O(K)$ fewer constraints.

In practical network scenarios, the largest term in the above expression is likely the number of possible paths followed by the vehicular users. To this extent, we discuss in Section VI-D how the number of paths actually used does not grow fast with the size of the reference mobility arena, thus extending the usability of the proposed model formulation in real-life environments.

VI. Performance Evaluation

In this section, we show the results obtained by solving the ILP model by means of AMPL (A Mathematical Programming Language) with the solver IBM ILOG CPLEX Optimization Studio. We start off by focusing on a tree-structured network topology , as this is one of the most widespread solutions for the backbone implementation (Figure 3). Routing to/from the APs happens along a shortest-path tree as represented in the figure. Later on in this section we further apply our optimization problem to more general network topologies, namely ring topologies and meshed ones. Table II further summarizes the scenario parameters and the values used in the optimization, except if stated otherwise. It is worth noting that the IEEE standard 802.11p for vehicular communications provides a peak capacity of 27 Mbit/s. However, the authors of [21] show that lower data rates maximize the data volume

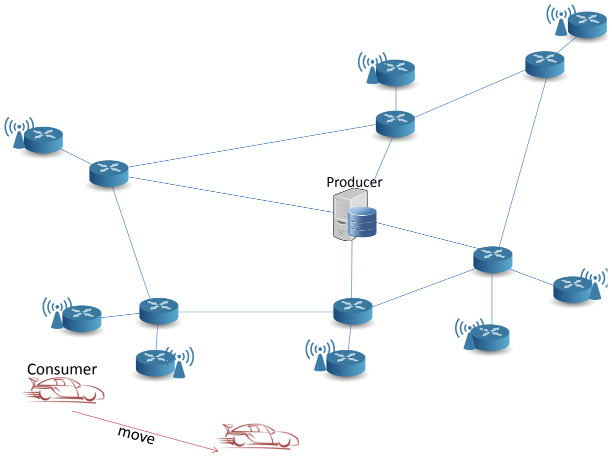


Fig. 5. The reference V2I scenario: the mesh topology.

that can be delivered, therefore we set the link capacity to 9 Mbit/s.

TABLE II
SCENARIO PARAMETERS

Parameter	Description	Value
τ	link latency	0.25 ms
c^R	radio channel capacity	9 Mbit/s
c_e	backbone link capacity	1 Gbit/s
L	tree depth	4
E	number of links	28/30/36
I	number of access points	8
K	number of content stores	14
C	size of the content catalog	50
α_r	exponent of content popularity	1
V	number of paths	5
α_p	exponent of path popularity	1
U_{tot}	total number of users	700
S_j	content size	1000 chunk
$D^{(\mathrm{ck})}$	chunk size	1000 byte
CS_k	content store size	2000 chunk
d	diameter of AP coverage	250 m
s	vehicle speed	90 km/h
T_{con}	total duration of connection	10 s
m_{iv}	connection indicator	1

Throughout this section, we are mainly interested in evaluating how network-related parameters impact the content retrievability and the cost for caching. We define the success probability P_{succ} as:

$$P_{\mathrm{succ}} = \sum_{j\in\mathcal{J}} \sigma_j \sum_{v\in\mathcal{V}} \beta_v A_{jv}.$$

The network costs depend on the content store size, the available bandwidth and the number of access points per path. In order to study how these design choices impact on the success probability under different network and mobility conditions, we use three slightly different models so as to expose different points of view. A model striking a balance among the different objectives according to some policy can be easily obtained.

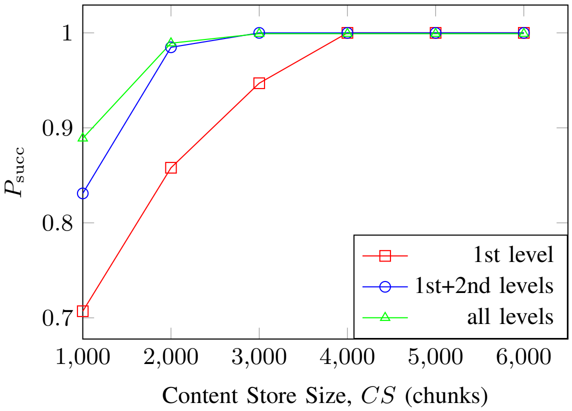


Fig. 6. Success probability, P_{succ}, in content retrieval depending on Content Store size, CS, assuming storage at different levels of the network.

A. Maximizing the Content Retrievability

Then, we evaluate the importance that storage has at the different levels of the tree. Figure 6 shows the success probability in content retrieval depending on Content Store size assuming that storage is available only at the APs, at the APs plus the second level of nodes, or at all the levels. As the Content Store size increases, the success probability also increases. Having storage only at the APs however does not allow the network to achieve its full potential. Adding storage to the nodes of the second level increases the success probability. However, as stated in [22], the performance improvement is at most 17% relative to the first level. Adding storage also to the third level nodes slightly increases the success probability, but only to a limited extent, which becomes null as the storage in the outer nodes grows.

Figure 7 shows the success probability P_{succ} in content retrieval depending on the contact time ratio. We define the Contact Time Ratio, CTR, as the ratio between the longest and the shortest contact between the user and each AP. We assume that the user moves with different speeds along its path but the total time the user is connected with the APs in the network is 80 s. We notice that as the ratio grows, the success probability lowers. This happens because the connection with the APs is more intermittent: the user alternates very long and short connections. Moreover, also in this case, we show that the gain provided by the additional level of content stores is at most 17% relative to the first level.

The results reported in Figures 6 and 7 lead us to the same conclusions of [22]. Indeed, by using a sensitivity analysis, the paper [22] shows that an NDN architecture with pervasive caching and nearest-replica routing can provide at most 17% of best-case improvement in network performance over a simple edge-based caching architecture. Moreover, it proves that if the edge caches are doubled, an edge-caching architecture performs even better than NDN. Thus, as concluded in [22], we can state that making pervasive use of content stores does not provide substantial advantages that could justify to add additional complexity to the network. While, we can think that the gain given from use of caching can be achieved in an incrementally deployable fashion.

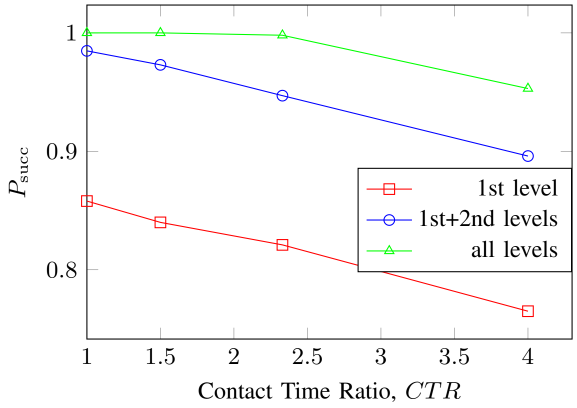


Fig. 7. Success probability, $P_{\rm succ}$, in content retrieval depending on contact time ratio, CTR, assuming storage at different levels of the network.

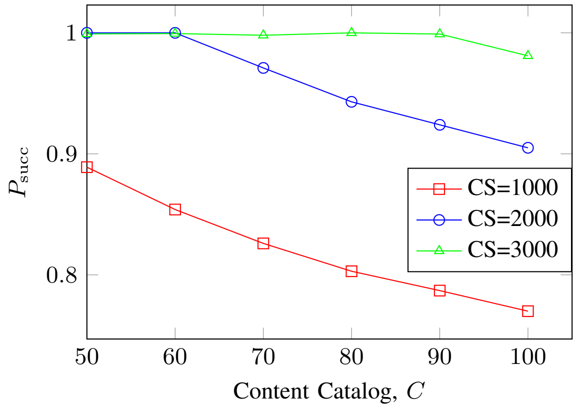


Fig. 8. Success probability, $P_{\rm succ}$, in content retrieval depending on content catalog size, C, assuming different storage sizes.

Figure 8 depicts the success probability $P_{\rm succ}$ in content retrieval depending on content catalog size. We assume different sizes for the content stores. It can be noticed that the success probability is almost 1 for a catalog size ranging from 50 to 100 contents with CSs of 3000 chunks. Then, the success probability is more than 90% if the content stores can store 2000 chunks. The success probability is smaller as smaller is the content store size and as bigger is the catalog.

The results reported so far show that large Content Stores, high link capacity, non intermittent connections, and small content catalog, all increase the success probability, but the effect is different depending on the other system parameters. In particular, increasing the wireless capacity and avoiding intermittent connections have a beneficial impact, but at the same time are unlikely to be controlled and changed. The size of the Content Store, on the other hand, is likely to be effortless to improve and can partly compensate a less dense network or a network with less capacity. However, we should take into account that spreading the content stores over a big network is far from being an easy task. Finally, we should consider that the bigger is the variety of content, the more important becomes the size of the content stores.

Figure 9 shows the number of users in the network versus the user speed. The continuous line represents the upper bound for the tradeoff between the number of users in the network and their speed to get a success probability equal to 1, assuming that all the contents chunks are stored in all the content stores of the first level.

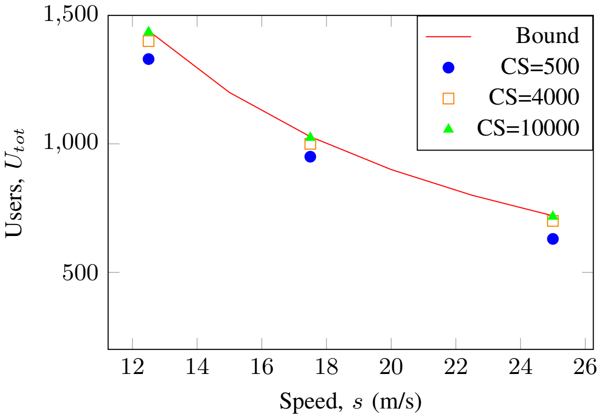


Fig. 9. Success probability, $P_{\rm succ} = 1$, in content retrieval depending on user's speed and number of users in the network varying the content store size CS.

The values are computed using the following equation:

$$U_{tot} \leq \frac{c^R \cdot d \cdot I \cdot \sum_{i=1}^{I} m_{i1}}{S_j \cdot D^{(ck)} \cdot s}$$

where s and U_{tot} are the changing variables to get the chosen success probability. Note that m_{i1} represents the number of contacts with the access points along the first path. The value is the same for all the paths.

The dots in the Figure represents the number of users versus their speed to achieve 100% success probability, if it is exploited the solution given by our model. We depict values varying the content store sizes CS_k, that is the same in all the tree levels. The bigger are the content stores, the nearest are the values obtained from our model to the bound value.

Thus, this Figure highlights that the results provided by our model are comparable to the optimal bound. Moreover, also small content stores can reach 100% of success probability paying only a small price in terms of decreased speed and diminished number of users.

Then, we also consider the two more general topologies depicted in Figures 4 and 5 with the same number of APs, backbone nodes and one content Producer. Routing to/from APs along the shortest path is pre-computed with respect to the topology.

Figure 10 compares the success probability values under different backbone network topologies. Namely, the chart reports the success probability, $P_{\rm succ}$ as a function of the content store size, CS, and parametrized with respect to link latency τ (and, consequently, to η_{eiv}). To make the comparison fair among the different topologies, the ring and meshed topologies share the very same number of APs as the tree topology, and the very same routing paradigm, that is, routes to/from APs adhere to the shortest path paradigm. The figure highlights the influence of the overall network latency, showing that the success probability is larger with smaller τ. Therefore, if the links have a larger latency, the storage size must be increased

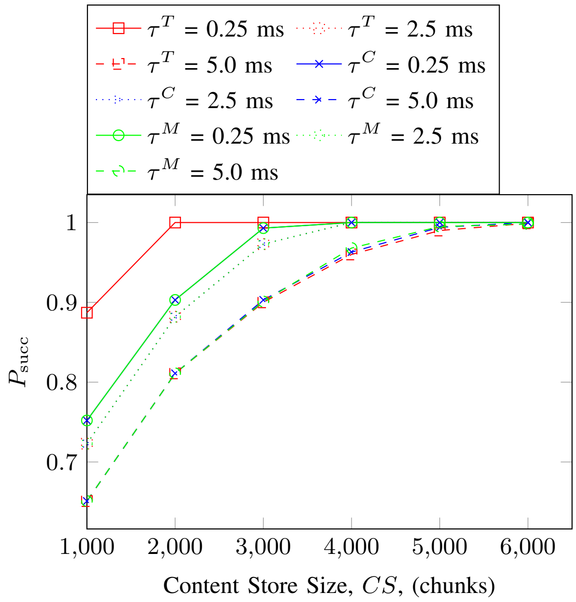


Fig. 10. Success probability, P_{succ}, in content retrieval depending on Content Store Size, CS, with different values of link latency, τ, in tree topology, T, circle topology, C, and mesh topology, M.

in order to achieve the same success probability. Moreover, as can be expected, the storage size has a big effect: the bigger are the content stores, the higher is the success probability. It can be noticed that the tree topology provides an higher success probability than the circle and mesh topologies with small τ. While, with bigger values of τ, the backbone topology does not influence the success probability because the contents are only retrieved from the access points.

B. Maximizing the Worst Content Retrievability

The second model tries to maximize the worst success probability among the possible paths. Thus, ϵ represents this success probability in content retrieval relative to the path with the smaller probability of being chosen. We assume that there are 480 users that connect with only six over the eight possible APs. The Figure 11 represents the values of ϵ as a function of the Contact Time Ratio assuming storage at different levels of the network. It can be noticed that the success probability lowers if the users move faster. Moreover, differently from Figure 7, the values of content retrievability are very close for all the three considered situations, this is due to the new *minmax* model that tries to balance the success probability of all the paths. Thus, we can say that adding levels of caching does not provide advantages in content retrievability among different paths. However, we have to pay the price of decreased performance.

C. Minimizing the Total Size of Content Stores

The third proposed model allows us to evaluate the size of the content stores to install in the network in order to guarantee a chosen success probability, that we fixed at $P_{\text{succ}} = 0.95$. The total content store size in the network is represented as $CS_{tot} = \sum_{j=1}^{C} \sum_{k=1}^{K} x_{jk}$. We study the number of users

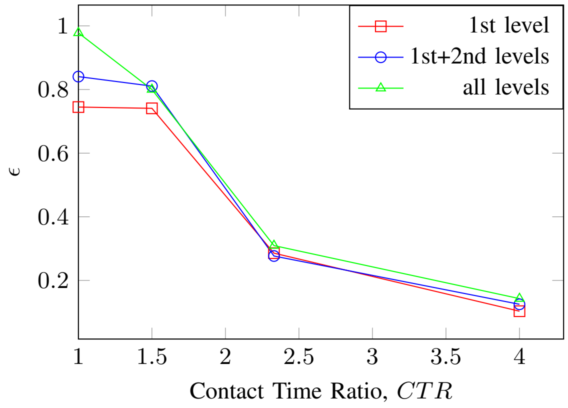


Fig. 11. Worst Content Retrievability, ϵ, versus the contact time ratio, CTR, assuming storage at different levels of the network. Number of contacts with the APs: 6; number of users in the network: 480.

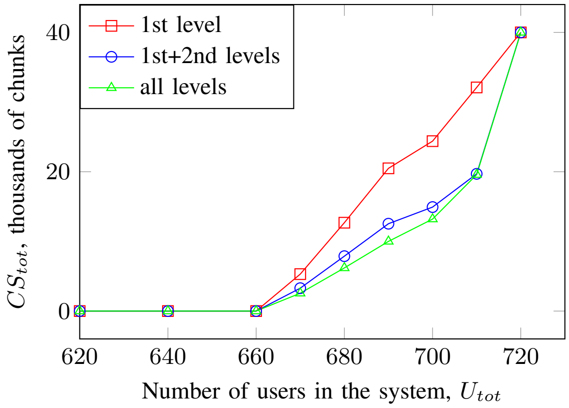


Fig. 12. Total content store size to achieve a success probability of 95% in content retrieval depending on number of users in the network, U_{tot}, assuming storage at different levels of the network.

that our network with different content store sizes can serve in Figure 12. Moreover, we assume that content stores are available at different levels of the tree. We note that the content stores are useless with less then 660 users because the network can afford the load and retrieve the contents from the repository at the root of the tree. While, ranging from 660 to 720 users, it is possible to raise the content store sizes in order to achieve the fixed success probability. Then, if the users are more then 720, the network can not supply the requests due to lack of bandwidth. Thus, content stores cannot overcome the limit imposed by the link access availability. Considering the case with content stores in all the tree levels, we noticed that, as the number of users grows, the content stores are pushed down towards the access network. On the contrary, the content stores in the upper levels are more effective when there are fewer users.

Figure 13 shows the total content store size as a function of the exponent of the path popularity to achieve a success probability of 95% in content retrieval. We assume that there are 480 users that connect with only six out of the eight possible APs. It can be noticed that when the users distribution across the available paths is more skewed, that is, when the congestion level of the more congested path increases, the

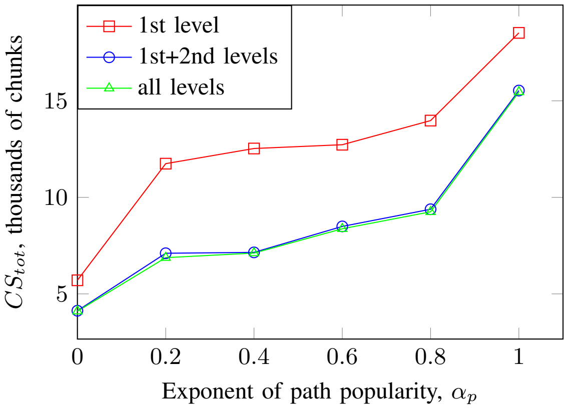


Fig. 13. Total content store size to achieve a success probability of 95% in content retrieval depending on the exponent of path popularity, α_p, assuming storage at different levels of the network. Number of contacts with the APs: 6; number of users in the network: 480.

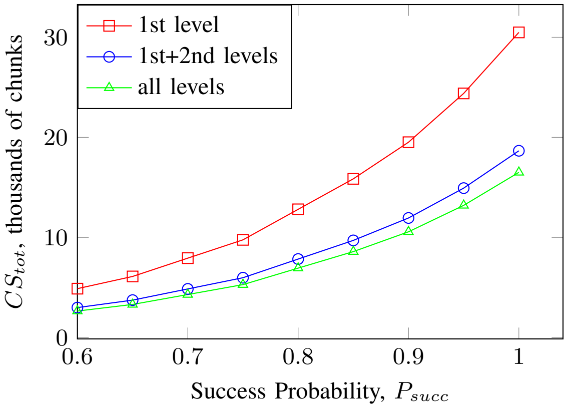


Fig. 14. Total content store size to achieve a chosen success probability in content retrieval, assuming storage at different levels of the network.

size of the deployed content stores increases. Moreover, as it could be expected, if only the first level in the topology is available, it is necessary to install bigger content stores than the cases with also the second and the third level. This Figure also confirms that the third level of content stores is less useful for the scenarios considered.

Figure 14 represents the total content store size as a function of the chosen success probability in content retrieval. As it can be expected, the higher is the desired success probability, the bigger the content stores size. Furthermore, moving from 80% of success probability to 100%, we should more than double the total content store capacity. Thus, if we would like to guarantee an higher probability in content retrieval, we should evaluate whether to invest more in content store is worth the costs. Moreover, it can be noticed that by installing content stores not only in the Access Points, it is possible to reduce the total content store size and so to save in costs.

Finally, Figure 15 depicts the total content store size needed to achieve a chosen success probability, considering variable users' speed. The succession of long and short connections with the APs brings to install bigger content stores in the network. However, the investment for the storage to guarantee the same chosen success probability in case of different CTRs is reasonable and provides an infrastructure that is more suitable for heterogeneous scenarios.

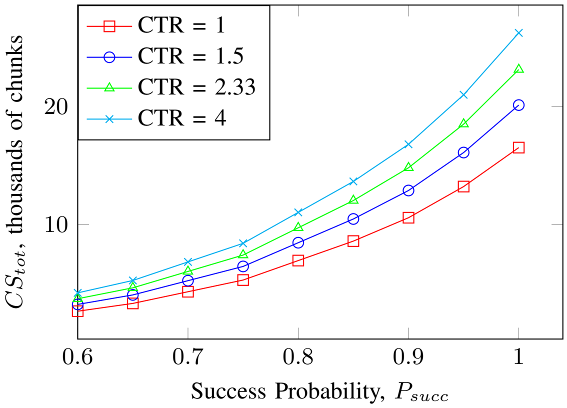


Fig. 15. Total content stores size to achieve a chosen success probability in content retrieval, assuming variable users' speed.

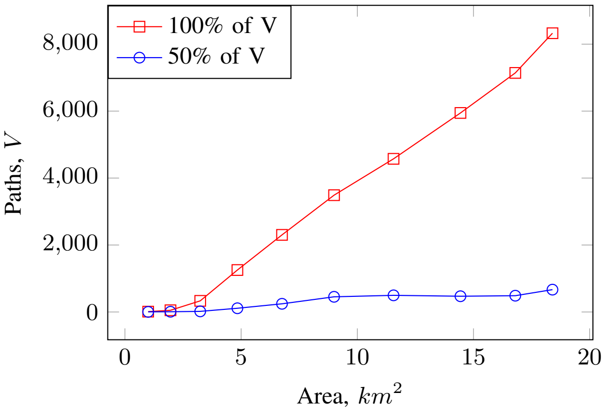


Fig. 16. Number of paths, V, depending on the area, considering all the paths, 100%, or only the most probable 50% of them.

D. Discussion on the usability of the proposed approach

As already observed in Section V-D, the cardinality of the set of possible paths V may increase exponentially in real-life mobility environments, which would make the solution of the optimization problem impractical, or, in the best case, too slow. To assess the scalability of the proposed approach, we have considered real-life vehicular traces of the city of Zurich [18] which provide detailed information on users mobility in a area of 18,4 square kilometers; leveraging these traces, we have evaluated the total number of distinct paths which can be observed in such vehicular realization when varying the size of the observation area. Namely, Figure 16 reports the total number of paths and the number of the most popular paths as a function of the size of the geographical region of observation (up to the complete Zurich area). Interestingly, the total number of paths and the number of most popular 50% paths are not exponential in the size of the area, but rather feature a linear growth.

Figure 17 shows the success probability when applying the optimization model considering a growing number of paths in the model parameter space starting from the most popular

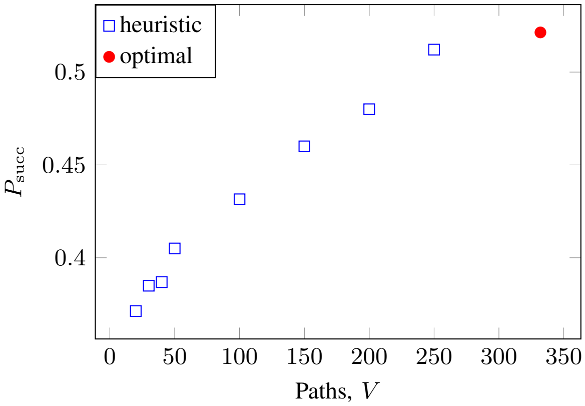


Fig. 17. Success probability, P_{succ}, in content retrieval obtained by running the model with a growing number of paths, V, in a 3,24 square kilometer area. The optimal solution is also plotted.

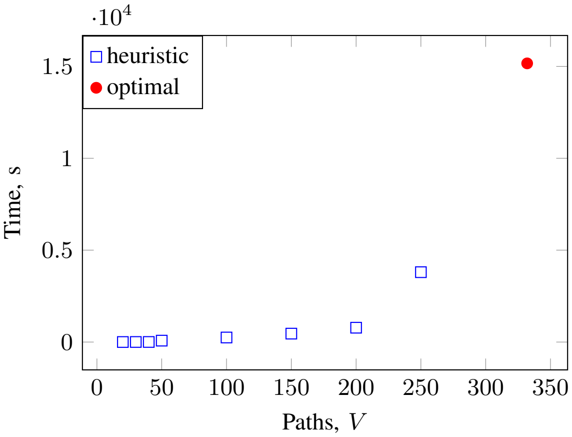


Fig. 18. Time, in seconds, to find the success probability in content retrieval obtained by running the model with a growing number of paths, V, in a 3,24 square kilometer area.

ones till all the paths are considered in the model. It is worth noting that by considering only the number of paths that cover 50% of probability, i.e. about 20 paths, instead of the total of paths, i.e. 330, the quality of the achieved solution is within 15% with respect to the optimum. Figure 18 depicts the time to solve the problems with a growing number of paths. We can notice that the time required to solve the problem with 20 paths is in the order of seconds, while it is in the order of hours if all the paths are considered. This type of analysis suggests that the proposed optimization-based approach can be of practical use to plan the content placement in realistic network scenarios.

VII. Optimal Content Pre-fetching vs Dynamic Cache Management

In this section, we compare the results given by our mathematical model which returns the optimal content pre-fetching strategy against dynamic caching policies. To this extent, we leverage the ndnSIM simulator [23] which implements a Least Recently Used (LRU) caching policy. In short, whilst the optimal solution statically design the caching strategy given a priori information on the reference network/scenario, in the dynamic case the caches are filled in and managed in reactive ways depending on the the observed NDN traffic pattern.

In order to have a fair comparison, we compute the number of active users in the network using the Little's law, i.e. $U = \lambda \cdot W$, where U is the mean number of users in the system, λ is the mean number of entering users per unit time and, W is the mean time spent in the system by a user. Then, we evaluate the following:

$$U_{tot} = f_{in}[P_{succ}T_m + (1 - P_{succ})T_p]$$

where f_{in} is the frequency of user entering the system, T_m is the mean time to retrieve a content and, T_p is the time spent in the system. Finally, P_{succ} is the success probability in content retrieval, that we have previously defined

Figure 19 displays the probability of content retrieval versus the number of users in the system, U_{tot}. This Figure compares the results obtained from the solution of our model and the same scenario implemented by means of ndnSIM. The results of the dynamic cache management (labeled ”LRU”) have been obtained for each number of users by averaging across 10 instances each one representing a different realization of the random processes representing the users arrival/departure and content popularity. The output results have been tested according to the t-student statistical test resulting in a confidence interval below 5%, given a confidence level of 95%.

It can be noticed that the success probability P_{succ} is 1 with less than 200 users and is 0 with more than 800 users in both cases. While, the gain in using the optimization model instead of the classical LRU policy ranges from 50% with about 400 users to a really big gain when the users are 700. Thus, we can say that the proactive placement of content into the content stores can provide big advantages over a reactive caching policy.

It is worth pointing out that proactive placement requires to have full knowledge on the content popularity and the users' mobility patterns. To this extent, the observed gain of the proactive content placement with respect to dynamic cache management can be interpreted as an upper bound of the actual gain; on the other hand, there is evidence in the literature that content popularity is fairly stable over time. As an example, the authors of [24] show that near-future popularity can be predicted with relatively high accuracy (close to 90%), while popularity after 90 days can be predicted with an accuracy around 85%; this results support the fact that the gap between proactive content placement and dynamic cache management observed in our simulation is indeed representative of realistic network conditions.

VIII. Conclusion

This paper exploits the in-network memory, foundation of the NDN framework, to simplify the delivery of the contents in a vehicular network. Two are the critical aspects for the success of NDN: content placement and caching policy. Our work provides a modeling framework that is a starting point for solving these issues. We define a model that aims to maximize the probability that a vehicular node can obtain the requested content during its stay in the network. We

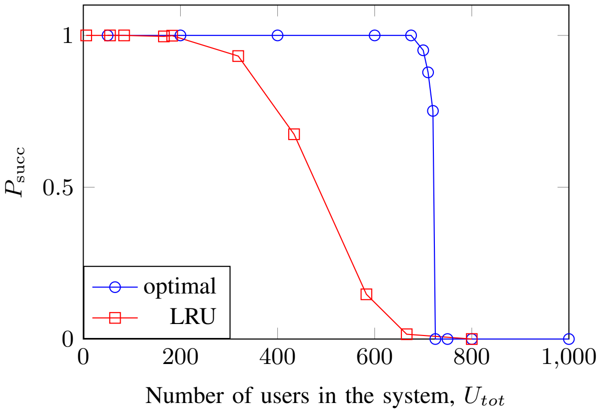


Fig. 19. Success probability, P_{succ}, in content retrieval evaluated using the proposed optimal content placement and ndnSIM with the LRU caching policy varying the number of users in the system.

evaluate the system in terms of success probability in content retrieval and investment on storage capacity in the network. By assuming that the Content Stores of the network nodes can be populated in advance, we provide an ILP formulation of the problem of optimally placing the content chunks in the network. The proposed optimization framework is then applied to realistic NDN scenarios to assess the impact of several network parameters onto the content retrievability. Such analysis provides insightful views on where to place content stores and which size of content store to place in ICNs. The following general guidelines/outcomes are sample results of the analysis carried out in this work:

- increasing the storage capacity at the nodes can improve the success probability especially when the APs are sparse, when the content catalog is bigger or when there are a lot of users in the network;
- the maximum gain is achieved by investing in storage capacity in the APs, that is, in the access network closer to the end-users;
- investing in storage capacity in the second level of network nodes also improves the performance of about 17%, especially if the link latency is low;
- the access link capacity is often the bottleneck on content retrievability, even if infinite content store size is available;
- the mobility pattern of the users has an impact on the content store allocation; namely, when users move with heterogeneous speed values, in general, bigger content stores are needed and adding levels of storage helps in performance improvement.

References

[1] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: Challenges and research directions," in *Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications*, ser. NoM '12. New York, NY, USA: ACM, 2012, pp. 1–6.

[2] Z. Zhu, A. Afanasyev, and L. Zhang, "A new perspective on mobility support," *Named-Data Networking Project, Tech. Rep*, 2013.

[3] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, pp. 181–192.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[5] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf) - an information-centric networking architecture," *Comput. Commun.*, vol. 36, no. 7, pp. 721–735, Apr. 2013.

[6] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. T. D. K, S. B. Zhang, G. T. K. C. Dmitri, K. D. Massey, C. Papadopoulos, T. A. Lan, W. P. Crowley, L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (ndn) project ndn-0001," 2010.

[7] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Vanet via named data networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, April 2014, pp. 410–415.

[8] C. Bian, T. Zhao, X. Li, and W. Yan, "Boosting named data networking for data dissemination in urban VANET scenarios," *Vehicular Communications*, pp. –, 2015.

[9] M. Amadeo, C. Campolo, and A. Molinaro, "Content-centric networking: Is that a solution for upcoming vehicular networks?" in *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, ser. VANET '12. New York, NY, USA: ACM, 2012, pp. 99–102.

[10] J. Lee, S. Cho, and D. Kim, "Device mobility management in content-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 28–34, 2012.

[11] Y. Rao, H. Zhou, D. Gao, H. Luo, and Y. Liu, "Proactive caching for enhancing user-side mobility support in named data networking," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, 2013, pp. 37–42.

[12] V. N. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve world wide web latency," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 3, pp. 22–36, Jul. 1996.

[13] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular wifi access," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '09. New York, NY, USA: ACM, 2009, pp. 263–276.

[14] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Content download in vehicular networks in presence of noisy mobility prediction," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 5, pp. 1007–1021, May 2014.

[15] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "Qoe-driven user-centric vod services in urban multihomed p2p-based vehicular networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 5, pp. 2273–2289, Jun 2013.

[16] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based vod streaming over vehicular ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 64, no. 3, pp. 1201–1217, March 2015.

[17] Y. Huang, Y. Gao, K. Nahrstedt, and W. He, "Optimizing file retrieval in delay-tolerant content distribution community," in *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems*, ser. ICDCS '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 308–316.

[18] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 108–119.

[19] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Teletraffic Congress (ITC), 2011 23rd International*, Sept 2011, pp. 111–118.

[20] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking." in *INFOCOM*, A. G. Greenberg and K. Sohraby, Eds. IEEE, 2012, pp. 2426–2434.

[21] A. Paier, R. Tresch, A. Alonso, D. Smely, P. Meckel, Y. Zhou, and N. Czink, "Average downstream performance of measured ieee 802.11

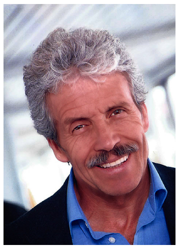
p infrastructure-to-vehicle links," in *Communications Workshops (ICC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–5.

[22] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable icn," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013, pp. 147–158.

[23] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, October 2012. [Online]. Available: http://named-data.net/techreports.html

[24] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.

Giulia Mauri is currently a Network Architect in the Network Services area of SIA S.p.A. Giulia received the degree of High Qualified Research Doctor (PhD) in Information and Communication Technology from Scuola Interpolitecnica di Dottorato (a jointly project among Politecnico di Milano, Torino and Bari) in April 2016. Her thesis, entitled Analysis of Security Issues in Information Centric Networking, covers problems related to trust management and privacy. Giulia has been a Visiting Scholar at University of California, Los Angeles (UCLA), in 2014. Giulia received the Master of Science Degree cum laude in Telecommunication Engineering in October 2012, her thesis was about preserving privacy in a Smart Grid scenario, and she received the 2nd price from CLUSIT for the thesis. Giulia received the Bachelor Degree in Telecommunication Engineering from Politecnico di Milano in July 2010 with a thesis about analysis of SAS data.



Mario Gerla (F'02) is a Professor in Computer Science at UCLA. During his Ph.D. program at UCLA he worked on early ARPANET protocols under Prof. Leonard Kleinrock. Since joining the UCLA Faculty in 1976, he has contributed to several network designs including wireless (ODMRP and CODECast) and Internet transport (TCP Westwood). He has lead the ONR MINUTEMAN project (2001-2006), designing the next generation scalable airborne Internet. He has developed at UCLA a Vehicular Testbed for safe navigation and intelligent transport. He was elevated to IEEE Fellow in 2002 and was recently recognized with the MILCOM Award (2011), the IEEE Ad Hoc and Sensor Network Award (2011) and the Sigmobile Life Achievement Award (2015)

Federico Bruno received the M.Sc. degree in telecommunication engineering from the Politecnico di Milano, Milan, Italy, in 2014. Since then, he has been with ICT Consulting S.p.A., Milan, Italy, where he is currently telecommunication consultant.

Matteo Cesana (M'00) is currently an Associate Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano, Italy. He received his MS degree in Telecommunications Engineering and his Ph.D. degree in Information Engineering from Politecnico di Milano in July 2000 and in September 2004, respectively. From September 2002 to March 2003 he was a visiting researcher at the Computer Science Department of the University of California in Los Angeles (UCLA). His research activities are in the field of design, optimization and performance evaluation of wireless networks with a specific focus on communication technologies fo the Internet of Things and Future Generation Cellular Networks. Dr. Cesana is an Associate Editor of the Ad Hoc Networks Journal (Elsevier).

Giacomo Verticale (S'00–M'03) is a Researcher at Politecnico di Milano, Italy. He is co-head of the Broadband Optical Networks, Security and Advanced Internet (BONSAI) Laboratory in the Department of Elec- tronics, Information, and Bioengineering (DEIB). Before joining Politecnico di Milano, he was with the CEFRIEL research center. He graduated in 2003 at Politecnico di Milano defending a thesis on the performance of packet transmission in 3G mobile networks. His research interests are in network security and in performance evaluation of network protocols.