Optimal Content Placement in ICN Vehicular Networks

Federico Bruno*‡, Matteo Cesana*§, Mario Gerla†, Giulia Mauri*§, Giacomo Verticale*§

* Department of Electronics, Information, and Bioengineering, Politecnico di Milano
§{name.surname}@polimi.it, ‡federico1.bruno@mail.polimi.it
† Department of Computer Science, University of California, Los Angeles
gerla@cs.ucla.edu

Abstract—Information Centric Networking (ICN) is a networking framework for content distribution. The communication is based on a request/response model where the attention is centered on the content. The user sends interest messages naming the content it desires and the network chooses the best node from which delivers the content. This way for retrieving contents naturally fits a context where users continuously change their location. One of the main problems of user mobility is the intermittent connectivity that causes loss of packets. This work shows how in a Vehicle-to-Infrastructure scenario, the network can exploit the ICN architecture with content pre-distribution to maximize the probability that the user retrieves the desired content. We give an ILP formulation of the problem of optimally distributing the contents in the network nodes and discuss how the system assumptions impact the success probability. Moreover, we validate our model by means of simulations with ndnSIM.

Index Terms—Information Centric Networking, Vehicular Network, Integer Linear Programming, ndnSIM

I. Introduction

In an Information Centric Network (ICN), the communication paradigm shifts from retrieving the content from a given location to retrieving the content from the network. 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 from which delivers the content. In particular, in the ICN paradigm, every node can be potentially the best source of a given content for a given consumer. Indeed, the widespread use of caching into the network nodes is one of the foundation of the ICN protocol. This paradigm blends well with specific issues of user mobility, such as intermittent connectivity and changing topologies. Let consider, e.g., a user in a moving car making a phone call. As the car moves, the Access Point (AP), to which the user is connected, is no longer available and the handover procedure is exploited in order to not loose the connection. A different approach could be taken in a mobile-oriented ICN scenario. In practice, when a consumer moves from one location to another, a seamless handover is easily achieved by reissuing the interest messages from the new location. Thus, the network will deliver the content from the best source for the new location avoiding the failure of the connection.

The benefits of ICN for mobile networks have attracted several researchers. The papers [1] and [2] provide a survey of mobility research in ICN. Both papers highlight the benefits that different ICN designs (e.g. DONA, CCN, NetInf, NDN...) provide to content and producer mobility. Then, the first work considers the challenges that should be inspected; while the second explains how to exploit the named-data paradigm for making the mobility management easier.

This paper discusses the advantages of ICN in Vehicle-to-Infrasctructure (V2I) communications, in particular the ability of an ICN network to store content in every node. In our setting, the vehicular users move along a chosen path connecting to a predefined sequence of Access Points, one at a time. The users entering the area served by the ICN network ask for a content from the content catalog, that is fixed. Each content is segmented in chunks, so the vehicular user can retrieve different parts of the content across different Access Points. A content is successfully retrieved if the user moves out of the area with all the chunks of a content. We formulate the Integer Linear Programming (ILP) problem of optimally deploying the content chunks in the network nodes so as to maximize the probability that a vehicular user succeeds in retrieving a given content from the network. The model allows to study the influence on the performance of different network parameters, i.e. the number of users, the network latency, the available bandwidth and the Content Store capacity. Moreover, the results suggest how to size the network, especially the Content Stores, to save for the installation costs. Finally, we provide evidence of the advantages given by using our model instead of a classical caching policy, i.e. Least Recency Used (LRU), by means of simulations with ndnSIM.

The paper is structured as follows: Section II provides an overall view of related work. The Section III reviews some of the basic concept of CCN protocol. Section IV explains the scenario considered in this paper and the behavior of the ICN-enabled nodes. The optimization model for content placement is presented in Section V. The numerical results are shown in Section VI. Conclusions are left for the last Section VII.

II. Related Work

The problem of reducing the latency in content retrieval is well investigated by the research community in various contexts. The paper [3] tries to solve the problem, predicting and prefetching those files that are most likely to be requested soon. The prefetching is done based on server advices to the client, who can choose to prefetch or not the suggested files. However, the model makes probabilistic predictions for prefetching based on received requests. Our model is an optimization model for content prefetching based on a priori knowledge of request probability distribution.

The paper [4] proposes a mobility management scheme for Content Centric Networking (CCN). 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.

A proactive caching algorithm for NDN is proposed in [5]. The main idea is to proactively ask and cache contents before the user moves from one access point to another, modifying the INTEREST packet and the basic communication protocol. The simulation shows that the proposed approach has better performance: lower handover cost, higher delivery ratio and shorter handover latency. Our solution does not require to add new functionalities or to modify the packets.

Finally, [6] raises first 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. Our work would also like to give an incentive in using ICN for mobility support.

III. CCN COMMUNICATION MODEL

Using as reference the Content Centric Networking (CCN) model presented in [7], a node asks for a content by sending an INTEREST packet to all available neighbors. An INTEREST packet carries the *Content Name*. After hearing the request, any node with the content responds with a DATA packet, only if the *Content Name* in the INTEREST matches the prefix of the *Content Name* in the DATA packet. When a node receives an INTEREST, it does the longest-match lookup on its *Content Name* and checks if there is a correspondence into its tables, i.e. the Content Store (CS), the Pending Interest Table (PIT) and the Forwarding Information Base (FIB). If the CS caches the DATA packet, the node sends out the content on the same face and throws out the satisfied INTEREST. Else, if the match is in the PIT, where the "not-yet-satisfied" INTEREST are stored, the corresponding entry is updated adding the requesting face and the INTEREST is discarded. While, if the match is in the FIB, that stores all the places data might be, the INTEREST is sent out to the next hop faces 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. 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 CS, 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 face(s) and adds it to the CS. A FIB match means an unrequested DATA, so the node gets rid of the packet.

IV. A VEHICLE-TO-INFRASTRUCTURE SCENARIO FOR ICN

This paper considers a Vehicle to Infrastructure (V2I) scenario in which vehicular users move in a predefined area covered with Access Points (APs) and download contents from the network. The APs are interconnected with a tree topology as shown in Figure 1. Since we do not consider the problem of choosing the routing of contents, we focus on the tree topology, but the results can be extended to any network.
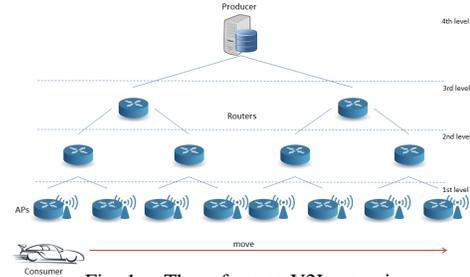
Fig. 1. The reference V2I scenario.

The reference scenario comprises a tree topology with L levels. Each node in the network follows a variation of the CCN protocol [7]. In particular, 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 $D^{(ck)}$. 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 the nodes at the higher levels. If they have the requested chunks, they send them downstream, otherwise they propagate the INTEREST upstream. The root of the tree, the Producer node, is a special case, because it 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 link between each pair of nodes has a capacity LC_l and a transmission latency τ, which accounts for the processing cost of the messages. Each of the I APs has a coverage of diameter d and a capacity LC_1. We neglect the transmission latency at the AP. There are U_{tot} users in the system and each user moves with a speed s. Thus, the total duration of connection between the user and the access points is $T_{con} = d/s$. Notice that we do not consider the time that the user needs to connect to the APs in order to simplify the model but we plan to consider it as a future work.

During its movement, the user chooses a path into the set of V possible paths, which determines the set of APs to which the user connects and it is different from path to path. The binary variable m_{iv} is equal to 1 if a user along path v connects to AP i, 0 otherwise. Each path v has a probability P_v of being chosen that follows the Zipf's law with exponent α_p: $P_v = v^{-\alpha_p} / \sum_{v=1}^{V} v^{-\alpha_p}$.

Therefore, the number of users on each path is $U_v = U_{tot} \cdot P_v$, while the average number of users connected to the ith AP per path v is:

$$U_i = \sum_{v=1}^{V} U_v m_{iv} / \sum_{v=1}^{V} m_{iv} \qquad (1)$$

The content catalog is composed of C content objects. Each content j has a probability of being requested that follows the Zipf's distribution with exponent α_r. The probability that the jth content is requested is: $Req_j = j^{-\alpha_r} / \sum_{j=1}^{C} j^{-\alpha_r}$.

The maximum amount of data that a node at level l can deliver downstream to the ith AP for users in the vth path is called the Maximum Downloadable Burst, MDB_{ilv}, and depends on the available bandwidth, which must be shared among all the requests from the downstream nodes. Assuming that the bandwidth is equally divided between the users connected on the same AP i along the path v, we have:

$$MDB_{ilv} = LC_l \cdot T_{iv} (D^{(ck)} \sum_{i \in \mathcal{I}(l,i)} U_i)^{-1} \qquad (2)$$

where $T_{iv} = T_{con} \cdot m_{iv}$ is the duration of the connection between AP i and the user along the path v and T_{con} is the total duration of the connection between the user and the APs. $\mathcal{I}(l,i)$ is the set of APs sharing the lth-level link with the ith AP:

$$\mathcal{I}(l,i) = \begin{cases} \{i\} & \text{if } l = 1 \text{ or } l = 2 \\ \{i, i+1\} & \text{if } l = 3 \text{ and } i \bmod 2 = 1 \\ \mathcal{I}(l, i-1) & \text{if } l = 3 \text{ and } i \bmod 2 = 0 \\ \{i, i+1, i+2, i+3\} & \text{if } l = 4 \text{ and } i \bmod 4 = 1 \\ \mathcal{I}(l, i+1-(i \bmod 4)) & \text{if } l = 4 \text{ and } i \bmod 4 \neq 1 \end{cases}$$

It is also worth noting that retrieving chunks from an upstream node incurs in a transmission and processing overhead. The variable $\gamma_{ilv} \geq 1$ represents the ratio between the time needed to retrieve a content chunk from a repository at level l for a user along path v and the time to retrieve the same chunk from the CS in the AP i. This factor can be computed as:

$$\gamma_{ilv} = \sum_{l=1}^{L} \frac{T_{ilv}^{(ck)} + 2\tau}{T_{i1v}^{(ck)} + 2\tau}. \qquad (3)$$

where $T_{ilv}^{(ck)} = T_{con}/MDB_{ilv}$ is the transmission time for a single chunk using the available MDB from a node at level l.

V. OPTIMAL CONTENT PLACEMENT

This section provides an Integer Linear Programming (ILP) formulation for the problem of content placement using an approach similar to [8]. We give the formulation for the case with $L = 4$ levels and $I = 8$ APs. The extension to other cases is straightforward. The optimization objective is to distribute content chunks into the CSs in order to maximize their availability for the retrieval to a user that moves around the network.

As opposed to the standard CCN protocol, we make the following assumptions:

1) The chunks in the CSs of the network nodes do not change over time according to a caching policy, but are predistributed according to the decisions of an offline management platform;

2) The content objects are protected with a Forward Error Correction code, which allows to obtain an arbitrarily large number of chunks for any given content. 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 INTERESTs for additional chunks.

Sets:
- Access Points: $i \in \mathbb{I} = \{1, \ldots, I\}$
- Contents: $j \in \mathbb{J} = \{1, \ldots, C\}$
- Paths: $v \in \mathbb{V} = \{1, \ldots, V\}$
- Tree levels: $l \in \mathbb{L} = \{1, \ldots, L\}$

Variables:
- $X_{ijl} \in \mathbb{Z}$: integer number of chunks of content j cached into a content store of level l linked to AP i;
- $A_{jv} \in \{0, 1\}$: boolean variable that is 1 if the content j is retrievable along the path v, 0 otherwise;
- $Rec_{ijlv} \in \mathbb{Z}$: integer number of chunks of content j retrieved from content store of level l linked to AP i along the path v.

Parameters:
- $Req_j \in [0, 1]$: probability of requesting content j;
- $P_v \in [0, 1]$: probability of choosing path v;
- $S_j \in \mathbb{Z}$: size of content j;
- $MDB_{ilv} \in \mathbb{Z}$: available MDB to the user along the path v connected to AP i from level l;
- $\gamma_{ilv} \in \mathbb{Z}$: the highest cost in retrieving chunks from the furthest content store in the distribution tree;
- $CS \in \mathbb{Z}$: content store capacity, in number of chunks.

Objective function:

$$\max: \sum_j Req_j \sum_v P_v A_{jv} \qquad (4)$$

Constraints:

$$\sum_{i,l} Rec_{ijlv} \cdot m_{iv} = A_{jv} \cdot S_j \quad \forall j, \forall v \qquad (5)$$

$$\sum_l (\gamma_{ilv} \cdot Rec_{ijlv}) \leq MDB_{i1v} \quad \forall i, \forall j, \forall v \qquad (6)$$

$$Rec_{ijlv} \leq X_{ijl} \quad \forall i, \forall v, \forall j, \forall l \qquad (7)$$

$$\sum_j X_{ij1} \leq CS \quad \forall i \qquad (8)$$

$$\sum_{i \in \mathcal{I}, j} X_{ij2} \leq CS \quad \mathcal{I} = \{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\} \qquad (9)$$

$$\sum_{i \in \mathcal{I}, j} X_{ij3} \leq CS \quad \mathcal{I} = \{1, 2, 3, 4\}, \{5, 6, 7, 8\} \qquad (10)$$

$$Rec_{ijlv} \leq MDB_{ilv} - \sum_{q=1}^{l-1} Rec_{ijqv} \quad \forall i, \forall v, \forall j, l = 2, 3, 4 \qquad (11)$$

$$X_{ijl} \geq 0, \ Rec_{ijlv} \geq 0, \ A_{jv} \in \{0, 1\} \qquad (12)$$

The objective is to maximize the retrievability of the content j, A_{jv}, that is the probability of satisfying the request of

TABLE I
SCENARIO PARAMETERS

Parameter	Description	Value
τ	link latency	0.25 ms
LC_1	access link capacity	9 Mbit/s
LC_l	link capacity ($l > 1$)	1 Gbit/s
L	tree depth	4
I	number of access points	8
C	size of the content repository	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
$D^{(ck)}$	chunk size	1000 byte
CS	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

the vehicular user. The objective function (4) depends on the probability of requesting the content j, Req_j and on the probability of choosing the path v, P_v.

The first constraint (5) represents the retrievability of a content. A content is considered *retrievable* if the summation of the chunks retrieved from the system is equal to the number of chunks the file is composed of. Equation (6) imposes a limit on retrievability that depends on the maximum downloadable burst (MDB). The parameter γ_{ilv} emphasizes the fact that the further the chunks are in the tree, 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 written in equation (7). Constraints (8), (9), and (10) limit the placement of the contents to the content store capacity. The sum of the chunks of all contents, stored in the content store of level $l = 1, 2, 3$ connected to AP_i cannot be larger than the maximum capacity CS. The link capacity is evaluated in equation (11). The number of chunks retrieved from CS cannot exceed the number of chunks the node can supply. Finally, the equations (12) guarantee that the decision variables are positive and A_{jv} is boolean.

VI. RESULTS

This section shows the results obtained by solving the ILP model by means of AMPL with CPLEX. We show the success probability in content retrieval, P_{succ}, that we define as the content retrievability weighted on the content popularity, Req_j, and depending on the choice of the path, P_v, so

$$P_{\text{succ}} = \sum_{j=1}^{C} Req_j \sum_{v=1}^{V} P_v A_{jv}.$$

Table I summarizes the scenario parameters and the values used in the optimization, except if stated otherwise. The results show how different network parameters influence the content retrievability. We compare different results in order to find the best tradeoff between success probability and network costs. The network costs depend on the content store size, the available bandwidth and the number of access points per path.

Figure 2 shows the success probability, P_{succ}, in content retrieval depending on the size of the content store, CS, and with different values of τ and, consequently, of γ_{ilv}. defined in Section IV. Thus, this chart 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 in order to achieve the same success probability. Moreover, as can be expected, the storage size has a big effect: the bigger are the caches, the higher is the success probability. We noticed that also the
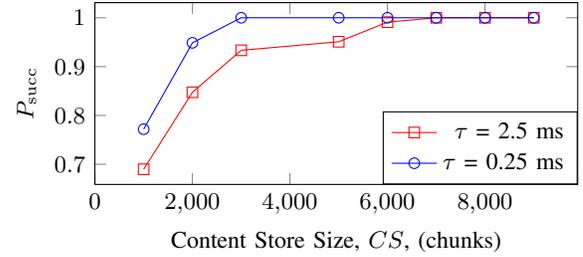
Fig. 2. Success probability, P_{succ}, in content retrieval depending on Content Store Size, CS, with different values of link latency, τ.

access capacity has an influence. If it is small, the vehicle must retrieve the content from multiple APs. Therefore, large gains are obtained by replicating the most popular contents on multiple APs. While, when the access capacity is large, the mobile users are likely to retrieve the entire content from a single AP and there is less margin for exploiting the content popularity.

Then, we evaluate the importance that storage has at the different levels of the tree. Figure 3 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 [9], the performance improvement is at most 12% 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 4 shows the success probability P_{succ} in
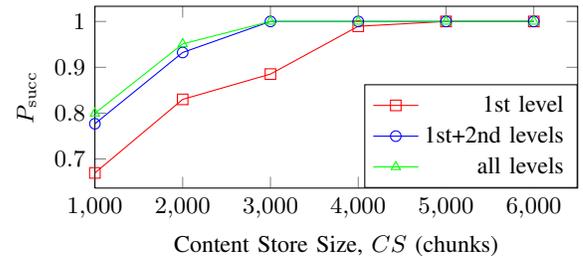
Fig. 3. Success probability, P_{succ}, in content retrieval depending on Content Store size, CS, assuming storage at different levels of the network.

content retrieval depending on 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 seconds. 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 caches is at most 12% relative to the first level. The results showed in Figures 3 and 4 lead us to consider if spreading caches over the network can provide advantages over the high cost of installation or incrementally deploying the Content Stores into the nodes, accordingly to the author of [9], could be a better and cheaper solution.
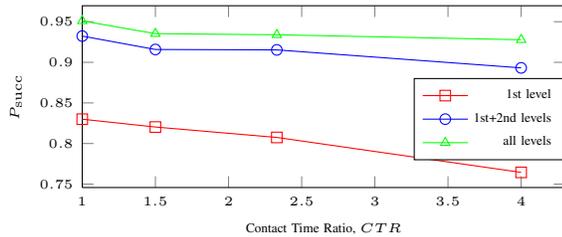
Fig. 4. Success probability, P_{succ}, in content retrieval depending on contact time ratio, CTR, assuming storage at different levels of the network.

Finally, Figure 5 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 [10] The main difference is that our model optimally places the contents into the CSs using a prefetching approach, while the model used in ndnSIM exploits a Least Recently Used (LRU) caching policy. 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 30% with about 400 users to 100% when the users are 700. Thus, we can say that the proactive placement of contents into the caches can provide big advantages over a reactive caching policy. However, using a prefetching policy requires to have a lot of information about contents and users.
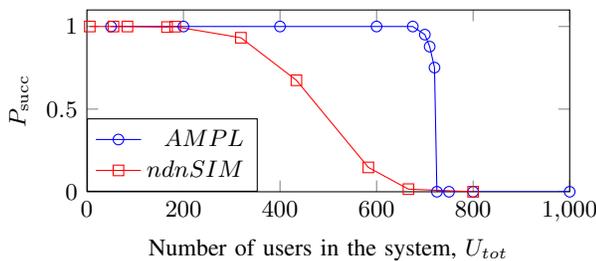
Fig. 5. Success probability, P_{succ}, in content retrieval evaluated using the proposed optimal content placement and ndnSIM with the LRU caching policy varying the number of users in the system.

The results reported so far show that large CSs, high link capacity, and non intermittent connections, all increase the success probability, but the effect is different depending on the other system parameters. In particular, increasing the wireless capacity and keeping the connection even have a beneficial

impact, but at the same time are likely to be most expensive improvements. While, the size of the CS is likely to be cheaper 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 caches over a big network is far from being an easy task.

VII. CONCLUSION

This paper discusses how a vehicular network can leverage on the ICN framework to simplify the delivery of the contents and maximize the probability that a vehicular node can obtain the requested content. The results confirm what one would expect. In particular, improving the storage capacity at the nodes can improve the success probability especially when the APs are sparse or when the capacity is low. The maximum effect is achieved by placing storage in the APs. Furthermore, it should be considered that storage in the second level of nodes also improves the performance of about 12%, especially if the link latency is low, but it has a bigger implementation cost. We also show that our model is robust and efficient against a common caching policy as LRU. As a future work, we would like to consider the possibility that the content catalog changes over the time in terms of number of contents and contents popularity. Moreover, we plan to evaluate different topologies and network scenarios in order to get close to a more realistic V2I network.

VIII. ACKNOWLEDGMENT

REFERENCES

[1] G. Tyson *et al.*, "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 *et al.*, "A new perspective on mobility support," *Named-Data Networking Project, Tech. Rep*, 2013.

[3] 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.

[4] J. Lee *et al.*, "Device mobility management in content-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 28–34, 2012.

[5] Y. Rao *et al.*, "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.

[6] M. Amadeo *et al.*, "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.

[7] V. Jacobson *et al.*, "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.

[8] Y. Huang *et al.*, "Optimizing file retrieval in delay-tolerant content distribution community," in *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*. IEEE, 2009, pp. 308–316.

[9] S. K. Fayazbakhsh *et al.*, "Less pain, most of the gain: Incrementally deployable icn," in *Proceedings of the ACM SIGCOMM 2013*, ser. SIGCOMM '13. New York, NY, USA: ACM, 2013.

[10] A. Afanasyev *et al.*, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, October 2012. [Online]. Available: http://named-data.net/techreports.html