Adaptive Content Seeding for Information-Centric Networking under High Topology Dynamics

Ion Turcanu*, Thomas Engel* and Christoph Sommer†‡
*Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg
†Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
‡Faculty of Computer Science, TU Dresden, Germany
https://secan-lab.uni.lu/ { ion-turcanu , thomas-engel }
https://www.cms-labs.org/people/sommer/

Abstract—**High-fidelity content distribution and other emerging applications of 5G and beyond-5G mobile broadband networking can put massive load on the core and Radio Access Network (RAN). To address this, direct Device to Device (D2D) communication has recently become a first-class citizen of these networks. While Information-Centric Vehicular Networking (ICVN) based on fog computing can indeed exploit such D2D links to alleviate the load on the RAN by proactively seeding content in the network, it has been shown that such seeding can cause even more load if performed where not needed. In addition, trying to determine where to seed content often causes additional load, negating the benefit of seeding. In this work, we therefore propose to adaptively seed fog nodes based on a purely virtual clustering approach. Here, vehicles are unaware of clustering decisions, thus no longer requiring an explicit exchange of control messages. We show that the benefit of such an adaptive approach goes beyond simply being able to flexibly trade off performance metrics versus each other: instead, it can consistently lower the load on the RAN link. We also show that this property even holds if node location information is only available as coarsely-grained as macro-scale grid cells.**

I. Introduction

Recent advances in wireless communication technologies have boosted the development of a wide range of new applications and services. These emerging applications pose increasing demands in terms of throughput, reliability, and latency, continuously stressing the Radio Access Network (RAN) and core network. To cope with these high demands, telecom companies are expected to invest enormous amounts of resources in next-generation cellular networks, new infrastructure and spectrum. At the same time, Device to Device (D2D) communication has become a first-class citizen of modern wireless networks, such as in 5G and beyond-5G designs. Despite this, content distribution in networks that present high topology dynamics, such as vehicular networks, remains particularly challenging.

The concept of an Information-Centric Networking (ICN) architecture has long been proposed as an alternative to traditional IP-based host-centric networking [1]. In ICN, users retrieve the content by name rather than looking for the specific host that holds the content. This means that any other user that currently holds a valid copy of the required content can share it immediately, allowing for more efficient and rapid content distribution. An ICN node can act as data producer, data requester, and/or forwarder of interest and data messages. To this end, every node in ICN keeps three main data structures: a Forwarding Information Base (FIB), a Pending Interest Table (PIT), and a Content Store (CS). When a node receives an interest message, it first checks if the requested content is available in the local CS. If it is not, the node registers a new entry in its PIT which matches the interest message with the sender and forwards the interest using the forwarding rules defined in the FIB. The main function of the FIB is to maintain updated records regarding which interfaces should be used to retrieve specific content. If the content is found in the CS of a node, a data message containing this content is created and propagated back to the requester following the PIT entries of the intermediate forwarders. This in-network caching and rapid content replication to neighboring nodes in a peer-to-peer mode are very attractive ICN features for vehicular networks.

The application of ICN architecture to vehicular networks is known as Information-Centric Vehicular Networking (ICVN). When the content is generated or maintained at a central location, the canonical means of getting it consists in sending an interest message to (and getting the data directly back from) the backend service via a mobile broadband RAN. In ICVN, vehicles can exploit D2D communication to alleviate the load on the RAN.

Several recent works propose bringing the content closer to potential requesting vehicles by proactively placing it on infrastructure nodes, such as Road Side Units (RSUs) or Multi-access Edge Computing (MEC) nodes [2], [3]. Vehicles can then retrieve this content directly from these infrastructure nodes via D2D communication channels. In our previous work [4] we propose seeding the content directly on a subset of strategically selected vehicles, identified based on network graph connectivity metrics, then analyze different fog seeding strategies. From this study we learned that proactive seeding can be extremely beneficial, but only if performed when and where it is actually needed.

In this work we build on these insights and propose Adaptive Content Seeding (ACS), an adaptive fog seeding system for ICVN. The proposed solution leverages network connectivity information, community detection algorithms, and node centrality metrics. While the system is based on the general concept of clustering (or community), it is important

IEEE Vehicular Technology Magazine

to note that these clusters exist as a concept on the content server only. Vehicles are not (nor do they need to be) aware of clustering decisions. ACS adaptively identifies the communities in which proactive seeding is required, needing only minimal (and, in many networks, incidental) information to do so. In brief, the key contributions of this paper are:

- we design an adaptive fog seeding system for ICVN;
- we compare our approach, ACS, with both a *never seed* (i.e., classical ICN content retrieval) approach and an *always seed* approach (i.e., a baseline approach based on global and always-up-to-date information about communities);
- we show that the performance of our adaptive ACS approach not only falls in between that of the *never seed* and *always seed* approaches, but does so with one notable (desired) exception: in terms of network load on the cellular network, it delivers best-of-both-worlds performance.

II. Related Work

Two major components that affect the performance of our proposed solution are the community detection mechanism and the seeding strategy. As a consequence, we first present general approaches towards community detection in vehicular networks and then describe the pre-eminent existing seeding algorithms in ICVN.

A. Community Detection in Vehicular Networks

Network community detection (or clustering) is the process of identifying groups of nodes with common characteristics and relationships. Clustering algorithms have been widely studied in the past in the context of Mobile Ad-hoc Networks (MANETs) [5] and, more recently, in Vehicular Ad-hoc Networks (VANETs) [6].

Cooper et al. [6] identify the importance of clustering algorithms in VANETs and propose a classification based on the application they target. Examples of such applications are routing (clustering to create virtual backbone networks), data offloading from the cellular infrastructure (clustering to select gateways between cellular infrastructure and vehicular network), and exploiting clustering to design a distributed Public Key Infrastructure (PKI). In this work, however, we use clustering to perform proactive seeding in heterogeneous ICVN.

More generally, the main motivation for clustering is to improve network performance by providing better local coordination and resource management. Yu and Chong [5] provide a general classification of different clustering techniques in MANETs. An important observation made by the authors is that the cost of building and maintaining clusters in MANETs is non-negligible, particularly due to the dynamic nature of the underlying network topology. This issue is even more challenging in VANETs, which operate in much more complex communication environments. Also here, the formation and maintenance of a cluster usually involves a series of steps [6]: neighborhood discovery, cluster head selection, cluster affiliation, and cluster maintenance. Each of these steps requires a number of control messages to be exchanged among vehicles, inducing additional overhead on the communication channel.

In contrast, in this work we use the concept of *virtual clustering*, in which vehicles do not need to be aware of being part of a cluster; hence there is no need for explicit exchange of control messages. Instead, the clustering operation is performed by a central controller, which we assume can derive an updated view of the current network connectivity graph from nodes' infrastructure connectivity.

B. Caching and Seeding in Information-Centric Vehicular Networking

One of the most important features introduced by the ICN paradigm is the nodes' ability to temporarily store received content objects and share them with other nodes interested in the same content in a peer-to-peer fashion. This concept is well described by Duarte et al. [7], who propose a distributed framework that addresses mobility-related issues in vehicular named-data networking. The mechanism that manages the decision of where and when the content should be stored is known as in-network caching [8]. In general, two main caching approaches can be distinguished in ICN: *reactive*, that is, the content is stored after it has been requested,[1] and *proactive*, in which the content is seeded in advance, anticipating potential future requests. The main goal of any caching strategy is to improve content retrieval by strategically placing it on well-connected nodes across the network so that the content is close to any other node that might be interested. To identify such well-connected nodes, many caching strategies in ICN rely on network connectivity metrics, such as network clustering and node centrality.

Caching is particularly important in networks with highly-dynamic topologies, such as vehicular or space-terrestrial integrated networks [9]. Modesto and Boukerche [10] highlight the importance of caching in ICVN and provide an overview of existing solutions. They evaluate a series of the most popular caching techniques and identify those that could be exploited in the context of ICVN. Mahmood et al. [2] propose a seeding mechanism on edge access points that is based on a priori knowledge of vehicular mobility information. In particular, by considering the trajectory of vehicles and their dwell time under each access point, the authors propose an analytical model that predicts the vehicles' request probability of a certain content object from a specific edge node. The mobility information is also exploited by Grewe et al. [3], who propose a distributed proactive seeding strategy for placing content objects on RSUs in order to improve network performance and decrease latency.

A distinguishing design feature of most existing solutions is seeding on fixed network nodes, from infrastructure RSUs to parked cars [11]. We instead exploit the idea of fog vehicular computing [12] and propose a strategy that seeds directly on carefully-selected vehicular nodes. In a previous work,

[1]The content can be stored not only by the node that has originally requested it, but also by any other node along the delivery path.

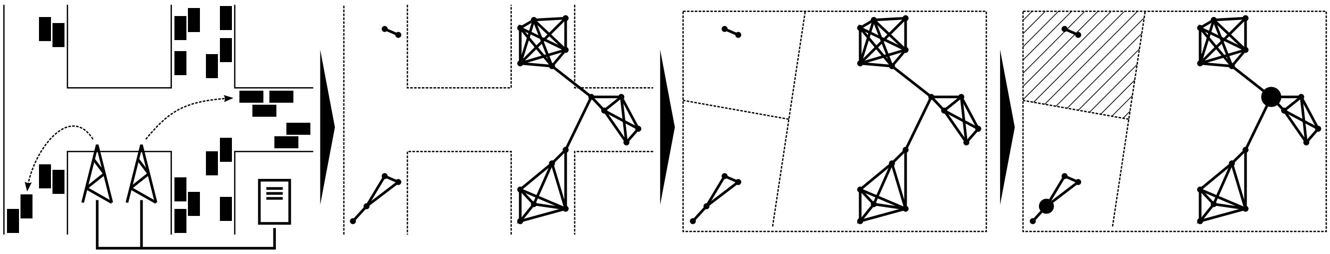

Figure 1. Network partitioning in virtual clusters, illustrated in four abstraction steps, from left to right. First step: physical topology of two intersections populated by vehicles. Second step: virtual topology considering connectivity only (with the street layout still overlaid for clarity). Third step: virtual topology considering connectivity and communities. Fourth step: virtual topology considering connectivity and communities with seed selection based on community and centrality (if/where needed).

Rondinone et al. [13] explore the possibility of injecting content via the cellular network to a subset of vehicles located at intersections in order to improve its dissemination in the vehicular network. However, the main objective of this work differs from ours: it aims at improving the Packet Delivery Ratio (PDR) while minimizing the number of cellular-based injections. In this work we focus on content retrieval while decreasing the impact on the RAN without unduly loading the D2D communication channels. In particular, we build on our previous work on static seeding strategies to propose a fog seeding mechanism that adapts to the current network situation and seeds only in areas where it is actually beneficial.

III. Graph-Based Multi-Layer Architecture

In our previous work [4] we proposed the following multi-layer architecture based on network connectivity graph and composed of three main layers: Connectivity, Community, and Centrality. The Connectivity layer is responsible for building and maintaining an updated view of the network connectivity graph. The Community layer partitions the graph into tightly-interconnected groups of nodes. Finally, the Centrality layer identifies in every community the most "central" node on which to seed the content.

A baseline seeding strategy we investigated in this work consisted of proactively seeding the newly created content in every community detected. A centralized network management entity combines different approaches to select one vehicle in every community on which to proactively seed the content of interest. Other vehicles in the network can then retrieve it via multi-hop D2D communication. The analysis revealed two important observations: the choices of both when and where to seed are crucial. However, we also observed that the optimal choice was not uniformly distributed within a larger network.

In this work, we will detail how a novel seeding strategy based on this graph-based multi-layer architecture against two baseline strategies.

IV. Adaptive Content Seeding (ACS)

The main idea behind our proposed seeding strategy is to only seed when and where it is actually needed. To do this, the centralized controller must be able to predict the distribution of potential requesters in different communities. The simplest way the controller can learn that a certain vehicle is a requester is if this requester downloads the content directly from the content provider, which only happens if the D2D-based content retrieval process has failed. It should be noted that this mechanism typically allows the controller to identify only some of the requesters, not all of them, unless all requesters failed to retrieve the content via D2D links.

We assume that if a vehicle becomes a requester at time t_{s-1}, it will continue to be a requester at time t_s, with s representing discrete time steps at which new content is generated. This models the case where many vehicles are interested in obtaining a larger data bundle which consists of multiple pieces of content (e.g., multiple packets of a stream or multiple fragments of a larger file) and obtaining the content is not time-sensitive.

Based on these assumptions, we can define our ACS algorithm to be executed by the controller as follows:

1) Identify all the requesters that have downloaded the content directly from the content provider at step t_{s-1};
2) At step t_s, create new communities based on the updated network connectivity graph and select the seeding nodes according to the chosen centrality measures (one seed in every community);
3) Seed the new content only in those communities that contain at least one of the requesters identified in step 1).

We note that, in our solution, the community detection process (or network clustering) is virtual rather than physical. Vehicles do not need to be aware of which virtual cluster they are in, nor do control messages need to be exchanged for joining, leaving, or maintaining virtual clusters.

Figure 1 illustrates this concept: the virtual clustering is simply an abstract view used by the controller to identify those parts of the network in which seeding is required. This means that there is no additional overhead introduced on the D2D channels.

V. Simulation Setup

We first define the general implementation details of our ACS approach, as well as two baseline approaches for performance comparison reasons. Then, we describe the simulation framework in which we implemented these solutions.

A. Implementation

The protocol implementation can be divided into two parts: (i) backend service operation and (ii) vehicular network operation.

1) Backend service operation: We assume the backend service generates new content every t seconds. We also assume the centralized controller maintains an updated view of the network connectivity graph. From a practical point of view, the information contained in every vehicle's local neighbor table can be exploited to build the connectivity graph.

Initially, we assume that this connectivity information is available at the centralized controller. This allows us to investigate a best-case scenario where such key information is readily available. Note, also, that in many 5G and beyond-5G designs, where most or all message exchange in the network is mediated by base stations, this information is readily known by the network. For comparison, we relax this assumption later in the paper (Figure 6) to investigate scenarios where much more coarse-grained information is known. We also assume that the backend service provides the generated content only on demand, e.g., only when a vehicle directly asks for it via the RAN. We name this approach *Never Seed*. Alternatively, the backend service can proactively seed the content in the network on a subset of strategically-selected vehicles, according to a predefined seeding algorithm. One such algorithm is our ACS strategy defined in Section IV. To validate our proposal, we compare ACS against the seeding strategy described in [4], in which the content is always seeded in every community. We refer to this approach as *Always Seed*. It should be noted that in all three strategies, the backend service continues to reply to direct content requests (via the RAN) coming from vehicles that were not able to retrieve the content from the D2D network.

2) Vehicular network operation: We assume a vehicle becomes requester with a certain probability p. Once a requester, the vehicle continues to request the content every t seconds until the content is no longer generated. From a simulation point of view, this means that a requester will continue to stay so for the entire duration of the simulation. The requester first tries to retrieve the content from the vehicular network via D2D multi-hop communication. If this attempt fails, the requester sends a direct request to the backend service via RAN, which then replies back with the requested content. Independent of how the content is retrieved, the requester locally caches the received content and becomes content provider for other potential requesters for the remaining lifetime of the content.

The D2D content retrieval process itself is based on the traditional ICVN approach. In particular, the requester issues an *interest* message on the D2D Control Channel (CCH) every time it needs new content. The interest is propagated via multi-hop communication up to H hops, where H is incremented from 1 up to $H_{\max}$ at every failed attempt (i.e., expanding ring search). It should be noted that FIBs are very difficult to maintain in an ICVN, given the highly dynamic nature of these networks. As a consequence, in our implementation, the multi-hop forwarding process of the interest message is based on the ETSI Contention-Based Forwarding (CBF) protocol as defined in ETSI EN 302 636-4-1 v1.3.1. If a vehicle that has a valid copy of the content in its CS receives an interest, it immediately sends an *acknowledgment* to notify the requester on the CCH, followed by the actual content on the D2D Service Channel (SCH). Both the acknowledgment and content messages are propagated to the original requester by following the PIT entries of the intermediate forwarding vehicles. In our implementation we simulate only the CCH communication and assume the content is successfully retrieved if the requester receives an acknowledgment.

B. Simulation Framework and Scenario

To validate the proposed approach, we use a simulation framework composed of the open source[2] vehicular network simulation framework Veins 5.0 [14], the discrete-event simulation engine OMNeT++ 5.4.1, the road traffic and mobility simulator SUMO 1.2.0, and the complex network analysis tool suite NetworKit 5.0. To derive statistically meaningful results of our stochastic simulations, we employ independent pseudorandom number generator streams and repeat every simulation using 100 different seeds.

We focus on the well-established highly realistic simulation scenario modeling mobility patterns and traffic flow in the city of Luxembourg over a 24 h time span, LuST [15].

We separate evaluations in this 24 h period by vehicular density, focusing on a density of 205 veh/km^2 in the evaluation, but pointing out differences where interesting.

Without loss of generality, we model D2D communication as IEEE 802.11p technology and furthermore assume an always-available channel of infinite capacity for the downloading of the content via RAN. We note that the proposed solutions are media-independent and the results obtained do not depend on a specific technology. In fact, other vehicular communication technologies could be used, such as 5G/C-V2X or IEEE 802.11bd, which will, in most cases, result in a different physical network topology graph. We also impose no limits on, e.g., amounts of data transferred and/or cached to focus on results for a single, identical piece of information and, in doing so, investigate effects that stem purely from the decision of seeding (or not seeding) content in isolation. The main simulation parameters are illustrated in Table I.

For the graph-based multi-layer architecture, we employ the best options for the connectivity and centrality layers determined in [4] (i.e., weak connectivity and closeness centrality) but simulate the full range of choices for the community layer: Connected Components, Parallel Louvain Method (PLM), Parallel Label Propagation (PLP), and Static Grid (baseline). We note that, for the study presented in this paper, the concrete choice of layers is secondary, so, in the performance evaluation, we focus on PLM community detection.

[2]https://veins.car2x.org/

Table I
SIMULATION PARAMETERS

Parameter	Value
D2D channel	IEEE 802.11p
Channel	5.89 GHz
Transmission power	20 mW
Bandwidth	10 MHz
Bitrate	6 Mbit/s
RAN channel	ideal
Simulated area	4 km^2
Simulation duration	100 s
Vehicular density (high, medium, low)	205, 130 and 70 veh/km^2
Request probability p	0.01, 0.2, 0.4, 0.6, 0.8 and 1
Beaconing interval	1 s
Content update interval t	10 s
$H_{\max}$	10
Interest/Acknowledgment size	43 Byte

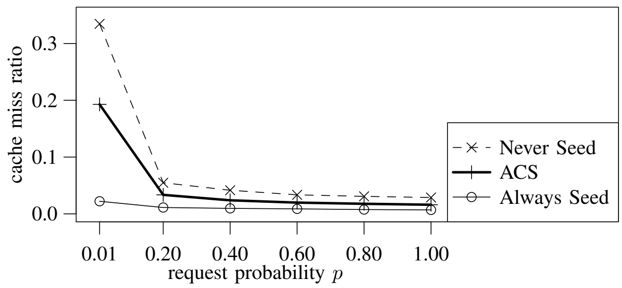

Figure 2. Cache miss ratio for different vehicular densities.

VI. PERFORMANCE EVALUATION

We start our discussion by investigating the performance of the three considered seeding strategies – our ACS approach, Never Seed, and Always Seed – according to two metrics, *Cache miss ratio* and *Content hop-count*. The cache miss ratio, computed by every requester, is calculated as the ratio between the number of non-satisfied content requests (i.e., interests sent via D2D links) and the total number of interest messages sent. The Content hop-count estimates the D2D SCH load and latency. It is computed by every requester and measures the number of hops to reach the content via D2D communication links. Note that this metric only captures cases where the content could actually be reached. These two metrics allow us to confirm that the behavior of all three investigated strategies is as expected.

Figure 2 shows the cache miss ratio for different request probability values. As expected, the *Never Seed* approach demonstrates the highest miss ratio, especially for low request probability values. The reason is that, without a proactive seeding strategy in place, vehicles are less likely to retrieve the content from the D2D network, especially when only a few nodes are interested in this content. The *Always Seed* strategy performs best because it employs the most "aggressive" proactive seeding approach, i.e., it seeds in every detected community. Finally, the cache miss ratio values of our ACS approach (effectively a "sometimes seed" strategy) are always between Never Seed and Always Seed. Similar considerations apply to the content hop count metric, illustrated in Figure 3. In particular, the more the content is proactively seeded (i.e., on more vehicles), the fewer hops are needed to retrieve it via D2D communication. Also, as the number of requesting vehicles increases (i.e., higher values of p), the content is better distributed across the vehicular network. This means that more vehicles have a valid copy of the content in their CS, which further decreases the content hop count metric.

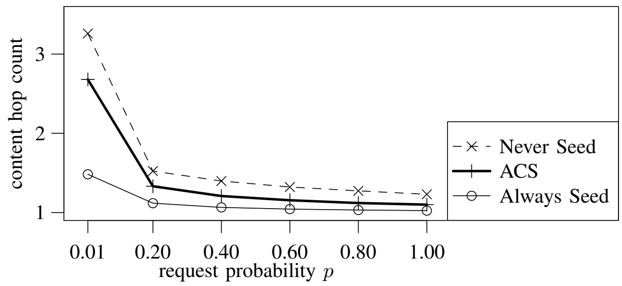

Figure 3. Mean number of hops needed to reach the content via D2D communication links.

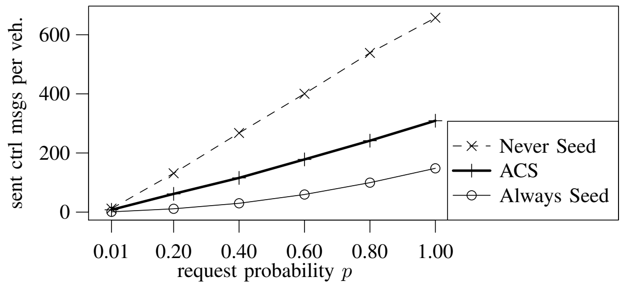

Figure 4. Mean total number of control messages per vehicle sent on the D2D control channel.

We now turn to a metric which allows us to investigate how this translates into load on the D2D channel, which we use to alleviate burden on the RAN. For this, we selected the number of *sent control messages* as a metric. It is computed by every vehicle by counting the total number of sent interest and acknowledgment messages.

Figure 4 illustrates the results obtained for this metric. Naturally, as the request probability increases and thus more vehicles are requesting the content, more control messages are being sent, causing a higher load on the CCH. Never Seed generates the largest number of control messages as a low fraction of vehicles will have content readily available – and, because of the expanding ring search approach, the most expensive operation is searching for content in a subgraph where there is none. In general, the total number of generated control messages by our ACS is always between Never Seed and Always Seed, which translates into a proportionate induced channel load on the CCH.

What remains to be investigated is if, indeed, our ACS approach is successful at reducing the load on the RAN – particularly, if it reduces load on the RAN more than simply

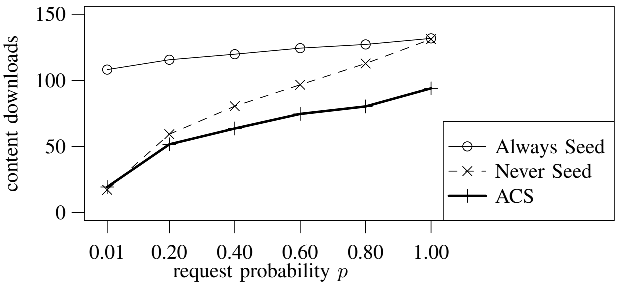

Figure 5. Mean number of content downloads from the RAN.

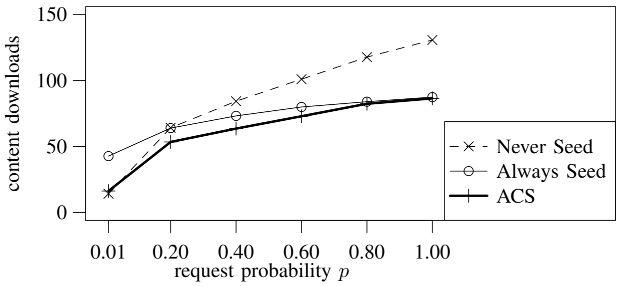

Figure 6. Mean number of content downloads from the RAN (static grid at low density).

trading it off for load on the D2D channel. For this, we select *Content downloads* as a metric, computed by the centralized controller and measures the total load on the RAN. It counts the number of content downloads from the backend service via RAN for all vehicles during one simulation run. The metric includes the downloads triggered by non-satisfied D2D content requests and downloads triggered by the proactive seeding strategy.

Figure 5 illustrates the results. We can see that, as expected, the Always Seed strategy triggers the highest amount of content downloads. These downloads are mainly generated by the proactive seeding mechanism, which indiscriminately seeds the content in every community, even when this is not necessary (e.g., even though two vehicles are in different communities they might still be part of the same connected subgraph, hence being able to obtain the content from one another via D2D). In fact, for low values of p, the Never Seed strategy performs better than Always Seed, meaning that always seeding in every community is counter-productive when the number of requesters is very low. However, the performance of Never Seed degrades rapidly as the number of requesters increases, since there are more and more vehicles that fail to retrieve the content via D2D communication and, as a consequence, download it via RAN.

Notably, the best performance is shown by our proposed ACS strategy, that is, for the target performance metric, it beats both other strategies. In more detail, since the content is seeded only in those communities that are known to have at least one requester, our strategy avoids unnecessary seeding in communities where it is not needed, being more efficient than Always Seed. At the same time, since ACS exploits proactive seeding, it is also more efficient than Never Seed. Overall, we can observe a decrease of up to 25 % of content downloads with respect to Never Seed ($p = 1$), and up to 80 % with respect to Always Seed ($p = 0.01$). Similar results have been observed when considering scenarios of low vehicular density (70 veh/km^2, data not shown).

Taken together with the results illustrated in Figures 2 and 3 this underlines that the key benefit of ACS lies not in making content *available* via D2D but in making it available via D2D *cheaply* (i.e., seeding content where it is most beneficial).

We now relax one more constraint, that of being able to employ PLM for robust community detection of cars. We switch the community detection algorithm to one that merely requires knowledge about which sector a vehicle is in (e.g., which base station it is connected to) and model this as a static grid partitioning of the scenario into a 2 × 2 grid (i.e., 4 communities down from the typically 12 communities created by PLM).

Results are shown in Figure 6. While, for very low values of p, Always Seed still performs worse than Never Seed, it significantly improves when the number of requesters increases, which is consistent with our previous results obtained in [4]. Interestingly, Always Seed generates fewer content downloads in the case of Static Grid with respect to PLM, which confirms the fact that most of the downloads are triggered by the proactive seeding mechanism. Most importantly, however, we can see that, even with only coarse-grained knowledge of communities, our ACS approach outperforms the baseline.

VII. Conclusion

We have proposed an approach for content distribution in 5G and beyond-5G network designs, which can make use of not just Radio Access Network (RAN) but also Device to Device (D2D) links. Similarly to related work, we proposed to proactively seed content on fog nodes and follow an Information-Centric Vehicular Networking (ICVN) paradigm for distribution. As a novel contribution, following insights that proactive fog seeding can be extremely beneficial, but only if performed when and where it is actually needed (and noting that obtaining fine-grained information about this can cause additional load on the network), we proposed to adaptively seed fog nodes based on a *virtual clustering* approach: Here, vehicles do not need to be aware of being part of a cluster, hence there is no need for explicit exchange of control messages. Instead, the clustering operation is performed by a central controller, which we assume can derive an updated view of the current network connectivity graph from nodes' infrastructure connectivity.

We showed that the performance of the proposed adaptive approach allows metrics like latency and load on D2D links to be flexibly traded off against each other, but – as desired – can consistently lower the load on RAN links and core network. We also showed that this property even holds if node location information is only available as coarsely-grained as macro-scale grid cells (e.g., only knowing the serving infrastructure node).

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A Survey of Information-Centric Networking,” *IEEE Communications Magazine (COMMAG)*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[2] A. Mahmood, C. Casetti, C.-F. Chiasserini, P. Giaccone, and J. Härri, “Mobility-aware edge caching for connected cars,” in *12th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2016)*, Cortina d’Ampezzo, Italy: IEEE, Jan. 2016, pp. 1–8.

[3] D. Grewe, S. Schildt, M. Wagner, and H. Frey, “ADePt: Adaptive Distributed Content Prefetching for Information-Centric Connected Vehicles,” in *87th IEEE Vehicular Technology Conference (VTC2018-Spring)*, Porto, Portugal: IEEE, Jun. 2018.

[4] I. Turcanu, T. Engel, and C. Sommer, “Fog Seeding Strategies for Information-Centric Heterogeneous Vehicular Networks,” in *11th IEEE Vehicular Networking Conference (VNC 2019)*, Los Angeles, CA: IEEE, Dec. 2019, pp. 282–289.

[5] J. Y. Yu and P. H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32–48, May 2005.

[6] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, “A Comparative Survey of VANET Clustering Techniques,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, Feb. 2016.

[7] J. M. Duarte, T. Braun, and L. A. Villas, “MobiVNDN: A distributed framework to support mobility in vehicular named-data networking,” *Elsevier Ad Hoc Networks*, vol. 82, pp. 77–90, 2019.

[8] 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*, Rome, Italy: ACM, 2009, pp. 1–12.

[9] F. Song, Y.-T. Zhou, L. Chang, and H.-K. Zhang, “Modeling space-terrestrial integrated networks with smart collaborative theory,” *IEEE Network*, vol. 33, no. 1, pp. 51–57, 2019.

[10] F. M. Modesto and A. Boukerche, “An analysis of caching in information-centric vehicular networks,” in *IEEE International Conference on Communications (ICC 2017)*, Paris, France: IEEE, May 2017, pp. 1–6.

[11] F. Malandrino, C. Casetti, C.-F. Chiasserini, C. Sommer, and F. Dressler, “The Role of Parked Cars in Content Downloading for Vehicular Networks,” *IEEE Transactions on Vehicular Technology (TVT)*, vol. 63, no. 9, pp. 4606–4617, Nov. 2014.

[12] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, “Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing,” *IEEE Vehicular Technology Magazine (VTMag)*, vol. 12, no. 3, pp. 55–64, Sep. 2017.

[13] M. Rondinone, J. Gozalvez, J. Leguay, and V. Conan, “Exploiting context information for V2X dissemination in vehicular networks,” in *14th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, IEEE, Jun. 2013, pp. 1–9.

[14] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[15] L. Codeca, R. Frank, and T. Engel, “Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research,” in *7th IEEE Vehicular Networking Conference (VNC 2015)*, Kyoto, Japan: IEEE, Dec. 2015.