Fog Seeding Strategies for Information-Centric Heterogeneous Vehicular Networks

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
{ion.turcanu,thomas.engel}@uni.lu
sommer@ccs-labs.org

***Abstract*—Content downloading of mobile broadband users today routinely causes network load to exceed what the Radio Access Network (RAN) can sustain without degrading user experience. Many works are therefore considering edge or fog computing paradigms and Device to Device (D2D) communication using 4G/5G C-V2X or IEEE 802.11p/bd links to obtain content. When content originates (or is maintained) at a central location, however, this requires expensive seeding of the content into the mobile network. We thus advocate basing the decision of when and where to perform fog seeding on a network graph that respects connectivity metrics while performing community detection and exploiting node centrality. We present different seeding strategies and investigate their respective benefits compared to traditional information-centric networking. Using simulations, we demonstrate that choosing a suitable strategy can yield substantial benefits – and vice versa.**

I. Introduction

As wireless communication technology matures, it is able to deliver higher data rates at lower latencies – but it is also *expected* to do so by its users [1]. Today, users of mobile broadband networks are routinely uploading, downloading, or streaming substantial content over wireless connections while on the move. This, however, comes at the cost of a proportional increase of load in the Radio Access Network (RAN), the core network, as well as the backend services. Today, this load routinely exceeds what, e.g., the RAN can sustain [2], [3] – with solutions made even more complex in networks characterized by high node mobility like the up-and-coming vehicular networks. Many researchers foresee massive investments in new infrastructure to enhance spectral efficiency and smaller cell sizes to improve spectrum reuse as the only ways to tackling this problem [4].

An alternative option has been gaining increasing interest recently: moving content closer to where it is consumed and using multi-hop Device to Device (D2D) communication to obtain content. Such D2D communication can either rely on the same access technology (like in 5G NR-V2X or 4G LTE-V2X) networks, or can be performed using alternative technologies (such as IEEE 802.11bd, IEEE 802.11p, or Wi-Fi) [5]. These developments had their beginnings in edge computing [6], [7] and have continued further with fog computing [8].

When content originates (or is maintained) at a central location, however, this requires seeding the content into the mobile network – and this process requires resources as well. This raises the question of when and where seeding can actually be beneficial to content retrieval in information-centric networks characterized by high topology dynamics. To fill this gap,

- we advocate basing the decision of when and where to perform fog seeding on a network graph that respects connectivity metrics while performing community detection and exploiting node centrality;
- we present different seeding strategies assuming a centralized instance (e.g., an operator or a content provider) for decision making;
- we investigate their relative benefits and their benefits with respect to traditional information-centric networking;
- we demonstrate that choosing sub-optimal metrics for connectivity, community detection, or node centrality can yield substantial performance loss, whereas following a suitable strategy can yield substantial benefits.

II. Related Work

In order to overcome the limitations of IP-based host-centric networks, Information-Centric Networking (ICN) has been considered as an alternative solution to improve content distribution [9]. One of the properties that makes ICN so attractive is the possibility of leveraging in-network storage for caching and rapid replication of content objects throughout the network in a peer-to-peer fashion [10]. In ICN, every cache-equipped network node can store and share with other nodes the content objects it receives, independent of whether the node itself is the destination of the content or not. One of the most important problems that can significantly affect content distribution performance is content placement, i.e., the strategy that governs the decision of where, what, and when to cache.

In general, the classical ICN caching approach provides for nodes to indiscriminately cache the content objects along the delivery paths. Chai et al. [11] have studied the effects of such a ubiquitous reactive caching strategy and found that it is sub-optimal. In particular, they demonstrate that allowing only a subset of nodes to cache the content can significantly increase the cache hit rates and improve network performance. They propose a centrality-based caching strategy that selects the nodes with the highest betweenness centrality as caching nodes and compare it with the classical ICN approach, as well as with a naïve random-based algorithm. The results show that even the random caching strategy outperforms the pervasive

978-1-7281-4571-6/19/$31.00 ©2019 IEEE

ICN approach, while the proposed centrality-based algorithm achieves the best results in terms of hop-count and cache hits.

In order to improve the network efficiency and reduce the backhaul traffic load, recent works have proposed bringing the content closer to end users by caching it at the edge of the network. Chang et al. [12] provide an overview of architectures and strategies for network edge caching and propose to exploit big data analytics schemes and machine-learning techniques in edge caching designs. They introduce and evaluate a hierarchical collaborative edge caching architecture, recognizing the performance gains that can be achieved. Said et al. [13] explore the concept of community detection in social networks when designing a proactive edge caching strategy. They propose a community detection algorithm based on the D2D cellular network and use an eigenvector centrality measure to select the most influential users in each community. Their idea is that influential users generate popular content which is more likely to be requested in the future by other users. The proposed approach is shown to satisfy up to 30 % more users when compared to a reactive caching scheme.

Being mobility friendly, the ICN approach has also gained significant interest in the vehicular networking domain: Modesto and Boukerche [14] provide a detailed overview of existing caching strategies in Information-Centric Vehicular Networks (ICVN). They identify the importance of strategically placing the content objects closer to end users to decrease average hop-count and increase cache hits. They evaluate the influence of different caching policies and parameters, such as content popularity, cache size, request frequency, replacement and coordination policies, and conclude that always caching the content does not automatically improve content delivery.

An initial study on how to optimally seed the content in an ICVN has been performed by Bruno et al. [15]. In particular, this work shows the benefits of pre-distributing the content to a predefined sequence of access points in order to maximize the probability of content retrieval. The content is assumed to be divided into chunks, which are split across different access points organized in a tree topology. Vehicles can download the content chunks from different access points while on the move. The authors formulate an Integer Linear Programming (ILP) problem of optimally distributing these chunks so as to maximize the probability that a vehicle can retrieve the requested content. A similar idea of distributing the content chunks across the infrastructure edge nodes has been considered by Mahmood et al. [16]. In this study the authors describe a mobility-aware probabilistic edge caching mechanism for vehicular networks, which considers information concerning the trajectory of vehicles and dwell time under each edge node. They propose an analytical model that predicts the probability of vehicles requiring the content at a specific edge node and develop a caching strategy based on this model. The proposed approach, evaluated using a realistic mobility scenario, outperforms existing popularity-based strategies.

The importance of selecting the right node to proactively cache the content in ICVN has also been acknowledged by Grewe et al. [17]. This study also leverages information regarding the future position of the vehicles in order to proactively cache the content chunks at the Road Side Units (RSUs). The proposed solution uses a centralized algorithm that requires the exact position of the vehicle, its velocity, the frequency of requiring the content, as well as the position and communication range of all the RSUs so as to place the content chunks in advance one hop away from the vehicle. In more recent work [18], the same authors propose a fully-distributed solution as well. Here, the edge nodes gather information regarding the content request frequency, number of hops to reach the content, and freshness time. This information is used to identify and prefetch popular data on the edge nodes in order to increase network performance and decrease latency.

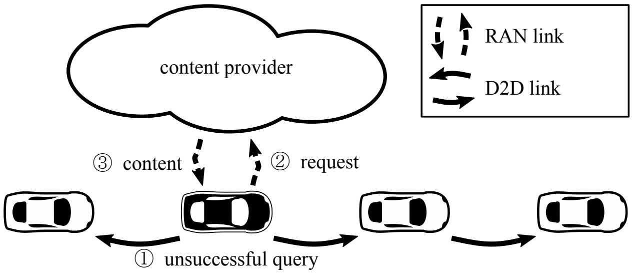


Figure 1. An example of unsuccessful content retrieval attempt in heterogeneous ICVN via D2D links.

Summing up related work, it can be seen that most of the existing solutions focus on proactively placing the content on fixed infrastructure nodes and leveraging mobility prediction mechanisms to identify where exactly to cache the content. Recent works have investigated the idea of fog vehicular computing [19], i.e., exploiting vehicles' computation and storage capabilities to provide services at the edge of the network [20]. We build on this idea and seed the content on strategically selected vehicles. In addition, based on recent studies that demonstrate the advantages of heterogeneous vehicular networking [21], [22], we propose a strategy that uses the D2D communication links to retrieve the content from the vehicular network, falling back on the RAN communication links only if the content is not found. While also exploiting network graph centrality measures, we provide a broader perspective on the effects of different graph connectivity metrics, and also focus on the impact of – and the trade-offs between – the RAN and D2D communication channels.

III. Proactive Fog Seeding Strategies

Figure 1 illustrates the typical process of retrieving content objects in a heterogeneous ICVN: a vehicle becomes a *requester* and queries the network every time it needs new content by issuing an *interest* message. This message is propagated in a multi-hop fashion via D2D links until it reaches another vehicle that has the content in its local storage, and which responds to the message by sending back the requested content. If the content is not found, the requester redirects its query (via a RAN link) to the content originator/maintainer running on a backend server. The problem with this approach is

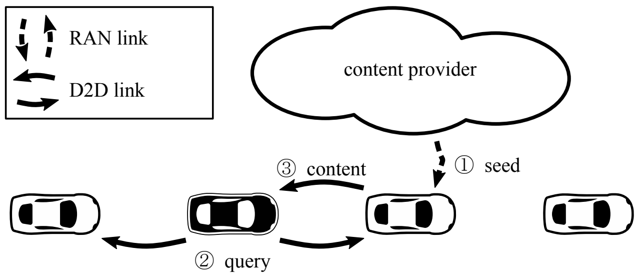
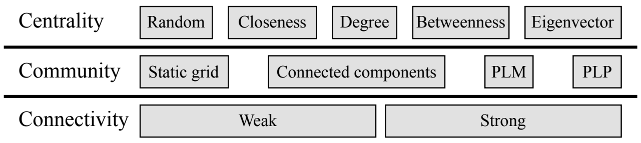


Figure 2. Proactive content seeding concept.



Centrality: Random | Closeness | Degree | Betweenness | Eigenvector

Community: Static grid | Connected components | PLM | PLP

Connectivity: Weak | Strong

Figure 3. Three-layer architecture.

that, if the content is not present or ill-distributed across the vehicular network, it puts pressure on both D2D and RAN communication channels – thus impacting network performance of other applications and increasing operational expenditure.

To address this problem, we adopt the idea of proactively seeding the content on strategically selected nodes, as shown in Figure 2. Our proposed seeding approach builds on a centralized decision mechanism based on graph theory that combines different connectivity metrics in order to select the seed nodes. In detail, we propose a three-layer architecture composed of

1) a network connectivity graph,
2) community detection algorithms, and
3) node centrality measures,

each of which offers multiple alternatives for achieving its respective goals. A high-level overview of this architecture is illustrated in Figure 3.

The proposed approach aims to select the seed nodes in such a way that the number of hops to reach the content from any other node in the network is reduced. The advantage is twofold: reducing the number of hops to the content, i.e., placing the content closer to the potential requesters, increases the probability of reaching the content via D2D links, which, in turn, reduces the number of requesters retrieving the content from the backend service via RAN links. In the following, we describe the mechanisms employed at each layer in detail.

A. Connectivity Layer

The connectivity layer is at the base of our architecture, laying the foundation for the other two layers. We assume a centralized network entity, such as a Software-Defined Networking (SDN) controller, that has an up-to-date view of the network connectivity graph. In practice, this can be obtained by letting vehicles periodically send updated local neighbour tables such as ETSI Local Dynamic Maps (LDMs) to the controller. These tables are typically obtained through a process of periodically exchanging one-hop beacon messages such as ETSI Cooperative Awareness Messages (CAMs) via D2D links. Alternative options are exploiting Bloom filter tables [23] or tapping into the central database available at no extra networking effort if Vehicle-to-Everything (V2X) messages are mediated via infrastructure as, for example, in the 3gpp 4G/5G or Japanese ARIB ecosystems. The choice of process for keeping these tables up to date on every vehicle and the upload procedure are out of scope for this article.

The neighbour tables are used to create the graph $G = (V, E)$, where V represents the set of vehicles and

$$E \subseteq \big\{(u,v) \,|\, (u,v) \in (V \times V) \wedge u \neq v\big\} \tag{1}$$

is the set of communication links between any two vehicles.

We consider two ways of building the set E, which results in two connectivity types: weak and strong.[1]

1) Weak Connectivity: Let N_i be the neighbour table of a generic vehicle i. We say that G is *weakly connected* if

$$E = \big\{(u,v) \,| u \in N_v \vee v \in N_u, \forall\,(u,v) \in (V \times V)\big\}. \tag{2}$$

2) Strong Connectivity: Considering the previous definition of N_i, we say that G is *strongly connected* if

$$E = \big\{(u,v) \,| u \in N_v \wedge v \in N_u, \forall\,(u,v) \in (V \times V)\big\}. \tag{3}$$

In other words, in case of weak connectivity, we add a link between two nodes if there is at least a unidirectional link between them (optimistic approach), while in case of strong connectivity, we add a link only if there is a bidirectional link between the two nodes (conservative approach).

B. Community Layer

Network community detection algorithms have been widely studied in the past [24]. These algorithms generally partition the network into groups of nodes that are more interconnected among themselves and less with nodes from other groups.

A core idea of our approach is to detect communities in G as a basis for determining where to seed the content. As a basis for discussion, we consider the four prototypical community detection mechanisms described below.

1) Static Grid: As a baseline method, we consider splitting the area of interest into static grids, similar to a Military Grid Reference System (MGRS), and letting vehicles belonging to the same grid be part of the same community. Notice that this approach does not need the connectivity graph G as an input, since the static grids are defined purely based on geographic coordinates.

2) Connected Components: A connected component of a graph G is defined as a subgraph H, in which a path (i.e., single or multi-hop communication link) exists between any two nodes of H, and there is no path between the nodes of H and any other node of G. If G is our network connectivity graph, then this method detects communities of vehicles which are completely disconnected from other communities; hence, there is no D2D path between vehicles belonging to different communities.

[1] Note that this means our definitions of *weak* and *strong* connectivity differ from the classic definitions in graph theory.

3) Parallel Louvain Method (PLM): The Louvain method [25] (like its computationally-optimized implementation PLM) detects communities in networks by maximizing the modularity score for each community. The modularity is a quality function initially introduced by Newman and Girvan [26], which quantifies the quality of a specific assignment of nodes to communities. PLM maximizes the modularity score by measuring the density of connections inside a community and comparing it with that of a random network. PLM relies on a greedy algorithm that aims to increase the modularity value by moving nodes from their community to a neighbouring community and computing the gain as

$$\Delta Q = \frac{d_i^C}{2m} - \frac{d_i \sum_{tot}}{2m^2}, \tag{4}$$

where d_i and d_i^C represent the degree of node i (in general and when belonging to community C, respectively), m is the number of edges of the graph G, and $\sum_{tot}$ denotes the total number of edges incident to community C.

4) Parallel Label Propagation (PLP): Raghavan et al. [27] propose a label propagation method that detects communities through an iterative process in which nodes, initialized with unique labels, adopt the label that the majority of their neighbours have. Potential ties are solved by uniformly randomly choosing a label. The process ends when groups of nodes reach a consensus on a unique label, thus forming a community. In contrast to the Louvain method, the label propagation approach does not require any optimization of a predefined objective function. It does, however, use the network structure defined by the graph G to guide the label propagation progress.

C. Centrality Layer

In graph theory, the node centrality measures how "central" a node is within the graph. There is no unique definition of node centrality – rather, it depends on the purpose and context in which it is applied. In our case, we want to identify the most central nodes in every community for seeding content, so as to bring content closer to any potential requesting vehicle via D2D links. The centrality layer of our architecture builds on top of the connectivity and community layers. In the following, we describe the most prominent centrality measures in graph theory, which we consider in our performance evaluation.

1) Random: The most straightforward method for selecting a seeding node inside a community is by uniformly randomly choosing it. This is our baseline approach that does not depend on the actual network connectivity graph.

2) Degree: The degree centrality of a generic node u is defined as:

$$C_{\mathrm{D}}(u) = \deg(u), \tag{5}$$

where $\deg(u)$ represents the number of links (edges) attached to u. According to our definition of G, the degree centrality measures the number of neighbouring vehicles of u.

3) Closeness: The closeness centrality of a generic node is the average distance (i.e., shortest path length) between the node itself and all other nodes in the network. Assuming a node u, the closeness centrality can be defined as

$$C_{\mathrm{C}}(u) = \frac{1}{\sum_v d(v,u)}, \tag{6}$$

where $d(v,u)$ represents the distance between nodes v and u. Notice that in our case the distance is measured as the number of hops between two vehicles.

4) Betweenness: The betweenness centrality of a node u quantifies the number of shortest paths passing through u, and is defined as follows:

$$C_{\mathrm{B}}(u) = \sum_{v \neq t \neq u} \frac{\sigma_{vt}(u)}{\sigma_{vt}}. \tag{7}$$

Here, σ_{vt} represents the total number of shortest paths between nodes v and t, while $\sigma_{vt}(u)$ is the number of shortest paths that pass through u.

5) Eigenvector: The eigenvector centrality measures the "importance" of a node in the network, which depends on the importance of its neighbours. Let $\mathbf{A} = (a_{u,v})$ be the adjacency matrix of G, i.e., $a_{u,v} = 1$ if there is a direct link between u and v (and $a_{u,v} = 0$ otherwise). Then, the relative centrality value of node u can be defined as

$$x_u = \frac{1}{\lambda} \sum_v a_{u,v} x_v, \tag{8}$$

where λ is a constant. The eigenvector equation can be written as $\mathbf{A}x = \lambda x$. At this point, the largest eigenvalue represents the desired centrality measure.

IV. Simulation Setup

We investigate our proposed fog seeding strategies using computer simulation. In particular, we use the open source vehicular network simulation framework Veins 5.0 [28], the discrete event simulation engine OMNeT++ 5.4.1, the road traffic simulator SUMO 0.32.0, and the complex network analysis tool suite NetworKit 5.0 [29].

We focus on a simulation study that uses IEEE 802.11p for D2D communication and assumes a channel of infinite capacity for downloading content from the backend service, if needed. We note that, aside from the resulting physical network topology graph, none of our results are tied to a concrete technology under investigation (i.e., results can be transferred to 5G or IEEE 802.11bd networks). In particular, the channel busy rate of our D2D channel remains in the single digit percent range.

For realistic vehicle mobility, we rely on the LuST scenario [30] which models highly realistic mobility in the city of Luxembourg. We simulate traffic at three different densities, low, medium, and high, corresponding to 70, 130 and 205 veh/km^2, respectively, as well as an average count of 13, 22 and 33 veh reachable via D2D, respectively. Table I shows an overview of key simulation parameters.

Table I
SIMULATION PARAMETERS

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

A. Implementation

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

1) Backend service operation: In order to obtain generalizable results, our application is set up to work as follows: at fixed intervals of 10 s, new content is created in the backend service. The centralized controller, based on the updated network connectivity graph, selects the best vehicles to seed the content using one of the available alternatives in each of the three layers of our architecture. Once these vehicles are identified, the content is proactively seeded via RAN links. This operation is repeated every time new content is created.

2) Vehicular network operation: The D2D content retrieval process builds on the classical ICN approach, which consists of three main components: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). At the beginning of the simulation, every vehicle can become a *requester* – that is, a vehicle that is interested in content – according to a predefined probability (the probability values are listed in Table I). Starting from a uniformly randomly chosen time, every 10 s, a requester performs an expanding ring search for this content. In particular, it incrementally sets the maximum number of hops the *interest* message can travel, starting from 1 and up to a predefined upper bound (in our case 10), until either the content is found or the upper bound is reached. Any vehicle that has the content available in its CS (either having requested and downloaded itself or having been seeded with the content) can reply. Otherwise, the vehicle records the $\langle senderId, interestId \rangle$ entry in the PIT, where *senderId* is the identifier of the vehicle that sent the interest, and *interestId* is the message identifier. If the search fails, the requester downloads the content from the backend service via the RAN.

In practice, the FIB is not used since maintaining predefined routes in vehicular networks is not feasible. Instead, we use a slightly modified version of the ETSI Contention-Based Forwarding (CBF) protocol [31], a timer-based data forwarding mechanism to disseminate interest messages across the vehicular network. In particular, every vehicle includes a 100 Byte Bloom filter [23] containing the list of its one-hop neighbours in the beaconing process. This allows a receiving vehicle to check if: (i) the ID of the sender is present in its local neighbour table, and (ii) its own ID is included in the sender's neighbour table. A vehicle is allowed to process an interest message only if these two conditions are satisfied, i.e., there is a bidirectional D2D link.

Our implementation of the content transfer process is also slightly different from the traditional ICN approach. In particular, if the interest reaches a vehicle that contains the content object, this vehicle replies with an *acknowledgement* message instead of directly sending the content object, as in the classical ICN approach. The acknowledgement is propagated back to the original requester by following the PIT entries of the intermediate forwarders. The goal of this message is twofold: (i) to inform the requester that the content was found, and (ii) to tell intermediate forwarders to switch to the Service Channel (SCH) for the content transfer process. Notice that the interest and acknowledgement messages are sent on the D2D Control Channel (CCH). The choice of the communication protocols to transfer the content on SCH is out of scope for this article. We assume the content to be successfully transferred if the acknowledgement reaches the requester. When the content reaches the requester, it is locally stored in the CS for a time duration equal to the remaining lifetime of the content and can be shared with other potential requesters.

Note that if we consider only the D2D content retrieval process described in this section (i.e., without the proactive seeding approach), we end up with a classical ICVN in which the content is stored/cached in a reactive way only at the requester side. We call this baseline approach *No Seeding* and compare it with the proposed proactive seeding strategies.

B. Metrics

We evaluate the performance of our proposed solution by measuring the following metrics:

1) Cache miss ratio: This is computed by every requester as the ratio between the number of non-satisfied content requests (via D2D links) and the total number of requests sent.

2) Content downloads: This metric measures the total number of content downloads from the backend service via RAN for all vehicles during one simulation run. It represents the sum of downloads triggered by non-satisfied D2D content requests and downloads triggered by the proactive seeding strategy. It is a measure of the total load on the RAN.

3) Content hop-count: This is computed by every requester and measures the number of hops to reach the content via D2D links. It gives an estimation of the D2D SCH load and latency.

4) Sent control messages: This counts the total number of sent interest and acknowledgement messages per vehicle. It is a measure of the overhead placed on the D2D CCH.

5) Community count: This metric measures the total number of detected communities and depends on the selected community detection algorithm.

978-1-7281-4571-6/19/$31.00 ©2019 IEEE

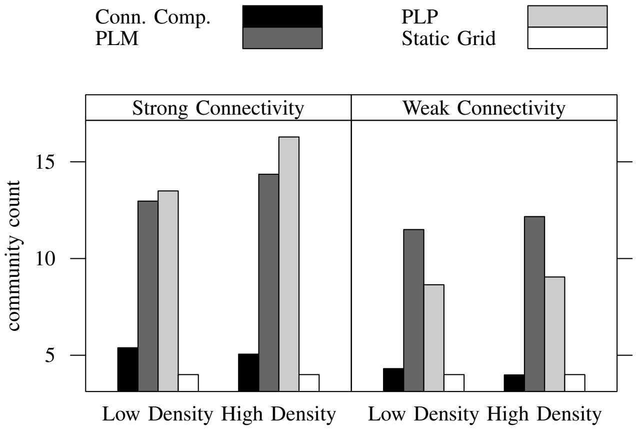


Figure 4. Mean number of communities created by different community detection algorithms based on *weak* and *strong* connectivity graphs.

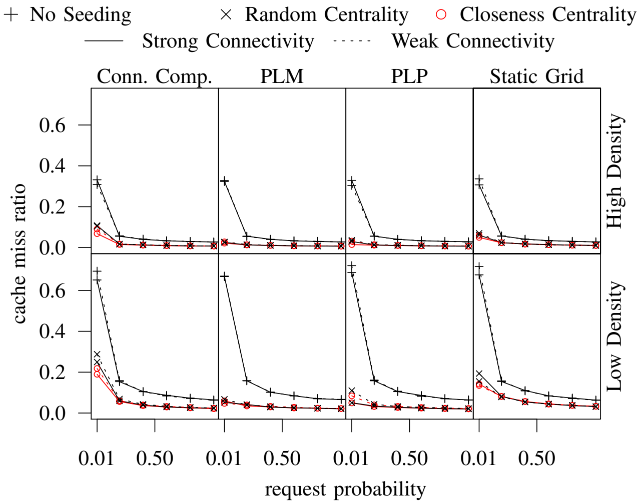


Figure 5. Cache miss ratio for different vehicular densities.

V. Performance Evaluation

Naturally, the optimal solution depends on how users and providers weigh the cost of using the RAN link vs. the D2D link. For example, it could be assumed that D2D is essentially free. Then, the only limitation is the channel capacity. In any case, there is most likely a desired trade-off between the cost of the RAN resources and the additional load generated on the D2D links. We investigate this trade-off according to the presented metrics. It should be noted that in the following figures we show only the Random and Closeness centralities, since the results obtained with the Degree, Betweenness, and Eigenvector centralities are similar to Closeness.

Figure 4 illustrates the performance of the community detection algorithms considering the strong and weak connectivity types for various vehicular densities. Notice that we consider a 2x2 static grid, which results in four communities for the Static Grid approach. The community detection algorithm that generates a similar number of communities is Connected Components, which creates between four and five communities, depending on the connectivity type. In general, given the fact that it is a more conservative approach, i.e., it creates fewer links in the graph, the strong connectivity type leads to more communities being generated. PLM and PLP tend to generate more communities than the Connected Components and Static Grid approaches. In particular, it can be noticed that when the graph is more sparse (i.e., strong connectivity), PLP creates 14–16 communities, while PLM only creates 13–14. The situation is reversed for the weak connectivity type, where PLM generates 11–12 communities while PLP creates 8–9 communities. Here, and in general, we omit results for medium vehicular density, which always fall in-between those for low and high densities.

Figure 5 illustrates the cache miss ratio for different connectivity types, community detection algorithms, centrality measures, and vehicular densities, when varying the probability of vehicles becoming requesters. The first thing that can be observed is the significant decrease in cache misses when using proactive fog seeding strategies. This is especially visible when the request probability and vehicular densities are low, where we can see a decrease of up to 94 % with respect to the No Seeding strategy. As the request probability increases, the number of requesters that will eventually retrieve and cache the content also increases. This leads to a better spatial distribution of the content, hence to a decreased cache miss ratio. A similar observation can be made for the vehicular density – the content is better distributed across the network at high vehicular densities.

A second observation is that the choice of the community detection algorithm seems to have a greater impact when the request probability is low. In particular, PLM and PLP tend to perform better in terms of cache miss ratio with respect to the Static Grid and Connected Components approaches, mainly because they create more communities, thus select more seeds, as can be seen from Figure 4. The difference between the various centrality and connectivity types is hardly noticeable for this metric. Only when the number of requesters is low, the Closeness centrality is a slightly better choice than Random, and strong connectivity yields an improvement over weak one.

These differences can be explained by analysing Figure 6, which shows the average number of hops to reach the content. Given the intrinsic properties of the Closeness centrality metric, it naturally selects vehicles as seeds which are closer to all the other vehicles in the community. This can particularly be seen for low values of request probability, where the Closeness centrality leads to fewer hops on average to reach the content than Random. This also true when considering the connectivity types: strong connectivity leads to fewer hops than weak, mainly because more communities with fewer vehicles in them are created. Clearly, all the proactive seeding strategies are better in terms of hops to reach the content than the No Seeding strategy. For example, with PLM and PLP, the content is found on average no further than 1.5 D2D hops when considering the Closeness centrality and strong connectivity. With the No

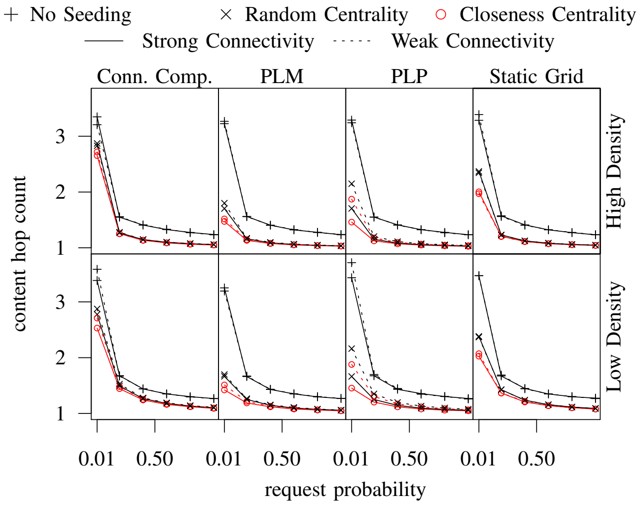


Figure 6. Mean number of hops needed to reach the content via D2D communication links.

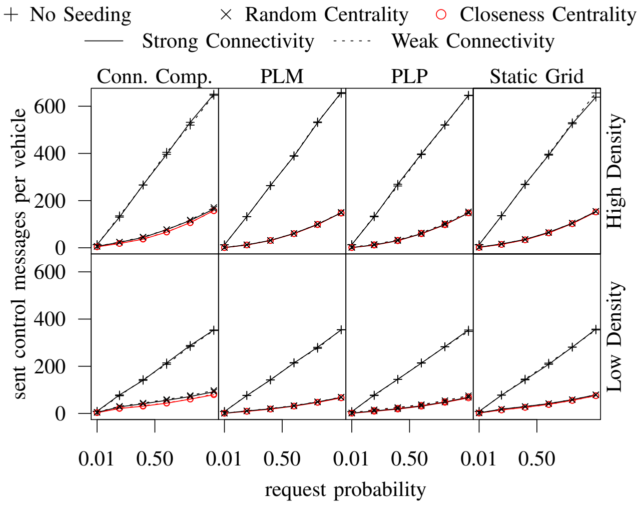


Figure 8. Mean number of control messages per vehicle sent on the D2D control channel.

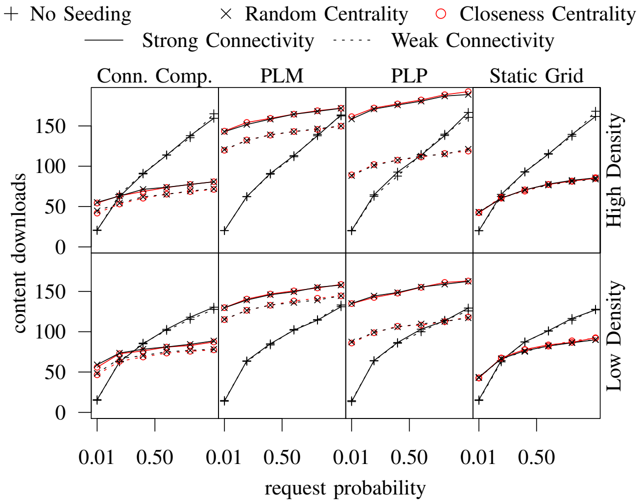


Figure 7. Mean number of content downloads from the RAN.

Seeding strategy, it may take as many as 3.5 hops on average to find the content. This can make a huge difference in terms of load on the SCH if the content to be transmitted is very large. The reason Connected Components performs worse than other approaches (for low values of request probability) is because the created communities are not evenly sized – usually it creates one or two big communities that contain most of the vehicles and several very small communities. Inside those big communities it then takes more hops to reach the content.

The load on the RAN in terms of average content downloads from the backend service is illustrated in Figure 7. An interesting observation is that the proactive seeding strategies do not always perform better. In fact, when the number of requesters is less than 20 %, proactively seeding the content is actually counterproductive, i.e., not seeding induces fewer content downloads than all the other approaches. This is because the number of vehicles interested in the content is not high enough to justify proactively seeding the content into the network, even if the cache miss ratio is high. Benefits of proactive seeding are only visible at higher request probabilities.

The results also show a strong correlation between the number of content downloads and the number of communities. In fact, given that the controller proactively and periodically (every 10 s) seeds the content on one vehicle inside every community, the algorithms that create fewer communities, i.e., Static Grid and Connected Components, perform generally better. Especially for high values of request probability, a gain of up to 50 % is achieved when compared to the No Seeding strategy. Connected Components in conjunction with a weak connectivity type seems to be the better strategy, achieving the lowest number of content downloads. In particular, for higher values of request probability, more than 10 % fewer content downloads are generated with respect to the Static Grid approach. PLM and PLP clearly perform worse, given that they also generate a higher number of communities. The connectivity type has a significant impact on the number of downloads, when it influences the number of communities, as is the case for PLM and PLP. There is no particular difference between the different centrality metrics.

Figure 8 illustrates the impact of the control messages on the CCH. It can be seen that on average a vehicle generates a significantly lower amount of control messages if one of the proactive seeding strategies is used. The main factors that affect this metric are the vehicular density and request probability: the more vehicles that are requesting the content via D2D links, the more control messages are generated. There are no significant differences among the various connectivity types, community detection algorithms, and centrality metrics.

VI. Conclusion

In this work we advocated basing the decision of when and where to perform fog seeding on a network graph that respects connectivity metrics while performing community detection and exploiting node centrality. We presented different seeding strategies assuming a centralized instance (e.g., an operator or a content provider) for decision making. We investigated their relative benefits and their benefits with respect to the baseline of traditional information-centric networking. We demonstrated that choosing sub-optimal metrics for connectivity, community detection, or node centrality can yield substantial loss of performance, whereas following a strategy adapted for the described use case can yield substantial benefits.

In more detail, we showed that proactive seeding in general is beneficial for both Radio Access Network (RAN) and Device to Device (D2D) communication channels. There is a clear advantage in selecting the seed vehicles based on a network connectivity graph, and using community detection algorithms and centrality-based metrics. In particular, by decreasing the number of hops to reach the content via D2D, latency and load on both Service Channel (SCH) and Control Channel (CCH) are significantly reduced. At the same time, proactive seeding improves D2D content retrieval by decreasing the cache miss ratio, leading to a significant reduction in the load on the RAN and, again, latency. However, creating too many communities and selecting too many seeds can be counterproductive – it can lead to a higher load on the RAN when compared to traditional information-centric networking.

Acknowledgement

This work was supported by the Luxembourg National Research Fund (FNR), CONTACT project, CORE/SWISS/15/IS/10487418.

References

[1] S. Shi, M. Hwang, V. Gupta, and R. Jana, "Latency Adaptive Streaming of 8K 360 Degree Video to Mobile VR Headsets," in *17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19)*, Seoul, South Korea, Jun. 2019, pp. 651–652.

[2] M. Peng, C. Wang, V. Lau, and H. V. Poor, "Fronthaul-constrained cloud radio access networks: insights and challenges," *IEEE Wireless Communications*, vol. 22, no. 2, pp. 152–160, Apr. 2015.

[3] H. Hudson, "5G Mobile Broadband: Spectrum Challenges for Rural Regions," *SSRN Electronic Journal*, 2019.

[4] E. Oughton, Z. Frias, T. Russell, D. Sicker, and D. D. Cleevely, "Towards 5G: Scenario-based assessment of the future supply and demand for mobile telecommunications infrastructure," *Elsevier Technological Forecasting and Social Change*, vol. 133, pp. 141–155, 2018.

[5] G. Naik, B. Choudhury, and J.-M. Park, "IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications," *IEEE Access*, 2019.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[7] R. Soua, I. Turcanu, F. Adamsky, D. Führer, and T. Engel, "Multi-Access Edge Computing for Vehicular Networks: A Position Paper," in *2018 IEEE Globecom (GLOBECOM 2018): Workshops*, Abu Dhabi, United Arab Emirates, Dec. 2018.

[8] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

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

[10] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.

[11] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks (extended version)," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 758–770, 2013.

[12] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, Jun. 2018.

[13] A. Said, S. Shah, H. Farooq, A. Mian, A. Imran, and J. Crowcroft, "Proactive Caching at the Edge Leveraging Influential User Detection in Cellular D2D Networks," *MDPI Future Internet*, vol. 10, no. 10, p. 93, 2018.

[14] 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.

[15] F. Bruno, M. Cesana, M. Gerla, G. Mauri, and G. Verticale, "Optimal content placement in ICN vehicular networks," in *International Conference and Workshop on the Network of the Future (NOF)*, Paris, France: IEEE, Dec. 2014, pp. 1–5.

[16] A. Mahmood, C. Casetti, C.-F. Chiasserini, P. Giaccone, and J. Harri, "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.

[17] D. Grewe, M. Wagner, and H. Frey, "PeRCeIVE: Proactive caching in ICN-based VANETs," in *8th IEEE Vehicular Networking Conference (VNC 2016)*, Columbus, OH: IEEE, Dec. 2016, pp. 1–8.

[18] 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.

[19] 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*, vol. 12, no. 3, pp. 55–64, Sep. 2017.

[20] F. Hagenauer, T. Higuchi, O. Altintas, and F. Dressler, "Efficient Data Handling in Vehicular Micro Clouds," *Elsevier Ad Hoc Networks*, vol. 91, p. 101 871, Aug. 2019.

[21] I. Turcanu, C. Sommer, A. Baiocchi, and F. Dressler, "Pick the Right Guy: CQI-Based LTE Forwarder Selection in VANETs," in *8th IEEE Vehicular Networking Conference (VNC 2016)*, Columbus, OH: IEEE, Dec. 2016, pp. 98–105.

[22] I. Turcanu, F. Klingler, C. Sommer, A. Baiocchi, and F. Dressler, "Duplicate Suppression for Efficient Floating Car Data Collection in Heterogeneous LTE-DSRC Vehicular Networks," *Elsevier Computer Communications*, vol. 123, pp. 54–64, Jun. 2018.

[23] F. Klingler, R. Cohen, C. Sommer, and F. Dressler, "Bloom Hopping: Bloom filter based 2-Hop Neighbor Management in VANETs," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 534–545, Mar. 2019.

[24] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Elsevier Physics Reports*, vol. 659, pp. 1–44, 2016.

[25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, P10008, 2008.

[26] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026 113, 2004.

[27] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036 106, 2007.

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

[29] C. L. Staudt, A. Sazonovs, and H. Meyerhenke, "NetworKit: A tool suite for large-scale complex network analysis," *Cambridge University Press Network Science*, vol. 4, no. 4, pp. 508–530, 2016.

[30] 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.

[31] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality," ETSI, EN 302 636-4-1 v1.3.1, Aug. 2017.