TraC: A Trajectory-aware Content Distribution Strategy for Vehicular Networks

Vitor Borges Coutinho da Silva, Miguel Elias M. Campista*,
Luís Henrique M. K. Costa

Universidade Federal do Rio de Janeiro - PEE/COPPE/GTA - DEL/POLI
P.O. Box 68504 - CEP 21941-972, Rio de Janeiro, RJ, Brazil

Abstract

In vehicular networking, contacts have short duration and may seldom occur. Hence, maximizing the amount of data transferred per contact opportunity is of utmost importance. In this work, we propose `TraC` which combines network nodes caching, typically used in Content-Centric Networks (CCNs), and users' trajectory knowledge. The idea is to improve data delivery by proactively caching the content requested by users over the access points along their trajectories to destination. To accomplish this, we use the network formed by access points to forward individual interests containing information regarding users' destination. In addition, we develop two forwarding strategies consistent with the content oriented paradigm, and a neighborhood discovery protocol, required for the operation of the proposed forwarding strategies. The results obtained through simulations in highway, urban, and rush-hour scenarios show that `TraC` can increase the fraction of interests

*Corresponding author.
Email addresses: `borges@gta.ufrj.br` (Vitor Borges Coutinho da Silva), `miguel@gta.ufrj.br` (Miguel Elias M. Campista), `luish@gta.ufrj.br` (Luís Henrique M. K. Costa)

satisfied as well as accelerates content delivery compared with a typical CCN implementations for wireless environments.
Keywords: vehicular networks, content distribution, content centric networks.

1. Introduction

Today, permanent and ubiquitous connectivity is a fundamental requirement for most users. Mobile communications can typically fulfill such needs at the cost of a monthly subscription fee for telecommunication services. Nevertheless, depending on the number of users and the amount of data transferred, mobile operators struggle to avoid network congestion. In this case, a well-known strategy to alleviate the mobile network is the utilization of traffic offloading. Indeed, statistics from 2014 confirm this trend, showing that 46% of the global mobile data traffic was offloaded, and that 54% is expected to be offloaded in 2019 [1].

A preferable technology for traffic offloading is IEEE 802.11, given the network availability and low cost. IEEE 802.11 networks, however, must deal with the limited range of access points, which leads the network to islands of connectivity [2]. In mobile scenarios, e.g., vehicular networks, these coverage gaps represent an obstacle for nodes moving at high speeds since contacts become shorter and less often. Even though mission-critical and security applications are not suitable for such conditions, infotainement applications can still have a high profit from contact opportunities. *The issue then becomes how could it be possible to maximize the amount of data transferred via multiple access points?*

Taking into account the current TCP/IP model, improving data transfers per contact opportunity opens venue for new Internet architectures since the original one was not designed for mobility. Content-Centric Networks (CCN) [3, 4] are an alternative architecture for the Internet which can also be used for content retrieval in vehicular networks. Users request a given content and the CCN takes care of finding and sending it back to the requesting node [5, 3]. The objective of CCN is to abstract the notion of IP addressing, since users are more and more interested on content, no matter where it comes from [5, 6, 7]. To avoid network flooding per content request, one key feature of the CCN architecture is content persistence in all network nodes and not only at the network edges. Such persistence can leverage data offloading in vehicular networks as a possibility to increase the number of content sources [8]. For instance, combining vehicular and content centric networks, we can think of storing content at each access point along a vehicle trajectory. This approach can potentially improve the efficiency of content retrieval at every contact opportunity between a vehicle and an AP, further maximizing users' interest satisfaction.

The main idea of TraC is to use CCN persistence to build proactive caches in all vehicular nodes based on users' trajectory. This can, at the same time, improve content delivery and circumvent vehicular mobility issues as CCN does not rely on host-oriented approaches. Previous CCN-based strategies neither rely on content caching in APs nor on users' trajectory information. Hence, the whole strategy has not been explored yet in the literature to the best of our knowledge. We rewrite the fourth paragraph of the Introduction to emphasize our paper contribution.

In this work, we propose `TraC`, a Trajectory-aware Content distribution strategy, which couples vehicular networking to the content-centric paradigm. Our main idea is to use CCN persistence to build proactive caches in all vehicular nodes based on users' trajectory. This can, at the same time, improve content delivery and circumvent vehicular mobility issues as CCN is not host oriented [2]. Previous CCN-based strategies, to the best of our knowledge, neither rely on content caching in APs nor on users' trajectory information. In `TraC`, we assume previous knowledge of users' geographical destination to forward interests toward APs that will probably be crossed along their trajectories to destination. These APs can, as a consequence, proactively download the content requested from the Internet even before the vehicle arrives. As a result, when the user associates to an AP along her trajectory, the content requested will be already available. The time needed then to request and transfer the content from the Internet to the connected AP is saved and the content can be immediately retrieved. To permit such proactive caching, we propose two strategies for vehicular interest forwarding between APs, Triangular Area Forwarding (TAF) and Distance Minimization Forwarding (DMF), and a neighborhood discovery protocol. Using TAF and DMF, received interests are only forwarded to APs along users' trajectories. To this end, APs need to be aware of users and neighbor APs position, which are obtained, respectively, with modifications to interest packets and with the proposed neighborhood discovery protocol. The performance of `TraC` is evaluated via simulations in three vehicular scenarios: urban, highway, and a realistic rush-hour scenario using the Cologne dataset [9]. In all experiments, we compute the fraction of users' interests satisfied, the content delivery ratio, and the network responsiveness in terms of how fast users' interests are satisfied. Compared with a typical implementation of CCN for wireless networks, results show that `TraC` satisfies more interests more quickly, reaching gains up to 50% in the fraction of interests satisfied.

This work is organized as follows. Section 2 introduces the CCN architecture and its utilization in the vehicular scenario. Section 3 proposes the Trajectory-aware Content Distribution (`TraC`) strategy. Section 4 describes `TraC` operation, providing more details concerning the proposed trajectory-aware strategies for content request and the Neighborhood Discovery protocol. In Section 5, we describe the simulation scenarios, parameters, and configurations. The obtained results are shown in Section 6. Section 7 lists related work. Finally, Section 8 concludes this work and discusses future directions.

2. Content-Centric Networks (CCN)

In this section, we overview the traditional CCN architecture and, in the following, we draw arguments for CCN deployment in vehicular networking. At the end, we briefly compare CCNs to Delay-Tolerant and Disruption Networks (DTNs).

2.1. CCN architecture overview

CCN is a content oriented network, which operates based on two types of packets: interest and content [4]. The first is responsible for requesting content to the network, while the second is the requested content itself. Since contents can be large, they can be divided in chunks. CCN packets are forwarded based on unique content or chunk names. The name syntax consists of a hierarchical sequence of components separated by a predefined separator, similarly to an URI. For instance, `/gta/publications/2015/10.pdf/3` identifies the third chunk of the file `10.pdf`, where `gta` is the first component of the name, `publications` is the second, and so on. An interest is then composed of the name of the content or chunk requested, in addition to other information used to orchestrate data retrieval.

The CCN architecture also defines a node structure and a node behavior to achieve its main features. The structure of a CCN node is depicted in Figure 1. Note that a CCN node is composed of three main structures: FIB (Forwarding Information Base), PIT (Pending Interest Table), and CS (Content Store). FIB is similar to an IP routing table, but instead of associating IP addresses to interfaces, it associates content names to route interest packets. PIT stores interests received, and the respective incoming interfaces, that were not yet satisfied. The CS caches content received using replacement policies such as Least Recently Used (LRU).

In CCN, network nodes send interest packets to request content. When receiving an interest, a node checks if the content chunk requested is stored in its CS. If it is not, the receiving node forwards the interest to other nodes and adds an entry to its PIT. In opposition, if there is a match, the receiving node sends back the content chunk through the interface from which the interest was received. The requested content follows the trail of PIT entries to reach the node that first issued the interest. Note that a key feature of CCN is content caching, which is a form of data persistence. This is indeed a major difference between host-oriented and content-centric paradigms. Whereas in the first one content is only content replicas can also be stored at core

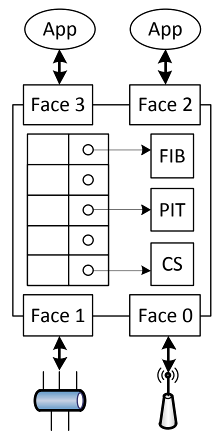


Figure 1: CCN node structure containing a FIB (Forwarding Information Base), a PIT (Pending Interest Table), and a CS (Content Store).

network nodes, e.g., routers; in the second, content can only be stored at the network edges.

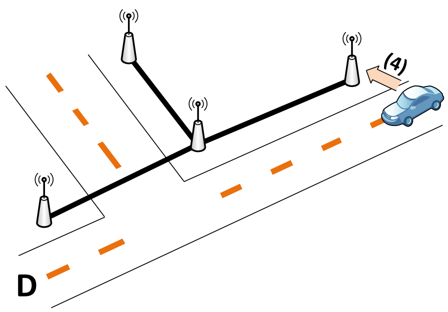
2.2. CCN in vehicular networks

Applying CCN to vehicular networks requires adaptations since the CCN architecture has not been designed to work on wireless scenarios. For example, in wired networks, interests are not forwarded to the same interface from which they were received. In multihop wireless networks, however, nodes typically rely on one wireless interface to receive and relay data, if needed. Another issue to apply the CCN architecture to wireless scenarios is the risk of broadcast storming, which occurs when a network is flooded with interest packets retransmissions. In vehicular networks, this may happen whenever a node that does not have the content requested in cache retransmits the interest to all neighbors.

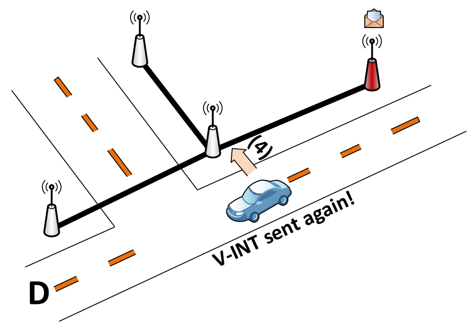
Although there is the risk of broadcast storming, broadcast transmissions can also be viewed as a positive aspect. In a broadcast medium, the same transmission can be received by multiple nodes, saving bandwidth and energy. Additionally, if nodes perform promiscuous content caching, i.e., caching content even without a pending interest, the content availability in the network can consequently increase. This characteristic represents a great advantage of applying CCN to vehicular networks since it leads to content retrieval improvements. In this work, we exploit this feature along with trajectory knowledge. Even though other works have already introduced CCN in vehicular networking [10, 11, 12, 13, 6, 14, 15, 16, 17, 18, 19], as far as we know, none of them have combined CCN and information about vehicles trajectory.

2.3. CCN vs. DTN

Delay and disruption Tolerant Networks (DTNs) are frequently used [20] in vehicular scenarios for infotainment applications, since they employ message switching with custody transfer. This is somehow similar to CCN as there is content persistence in network nodes. Hence, DTNs circumvent the connectivity problem but, unlike CCN, still require network nodes to know a priori to which node a given content must be requested. Moreover, DTNs neither consider the existence of different sources of the same content, i.e., nodes with different identifications, nor the existence of multiple Internet gateways from scratch. This means that they cannot provide the most efficient content delivery in scenarios such as those considered in this work.



(a)



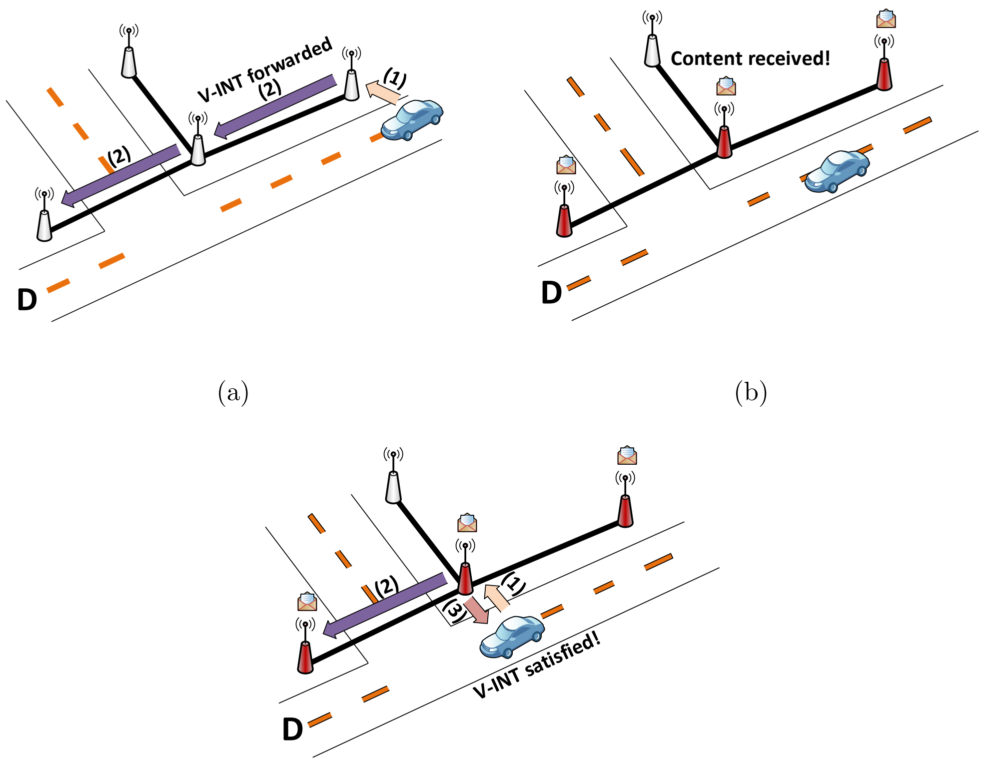
(b)

Figure 2: Without `TraC`: (a) Content request: (1) a user sends a content request to the AP connected, only the AP connected downloads the content; (b) Content request again: (2) As the user moves away from the previously connected AP, she needs to request the content again.

3. The Trajectory-aware Content Distribution (`TraC`) Strategy

In this work, we propose the use of proactive caching in access points (APs) to improve content distribution in vehicular networks. The key idea consists in *proactively* download the content requested by a user to the APs along her trajectory toward destination. The goal is to make the content available in the cache of each AP even before the requesting user arrives. This strategy contrasts to the host-oriented approach since users do not have to wait the content to be downloaded every time she connects to a different AP. In vehicular networks, the downloading time becomes more significant since contact opportunities have often shorter durations.

Figure 2 illustrates content retrieval without `TraC` utilization. Note that every time a user connects to a different AP, she has to wait for the content to be first downloaded. With `TraC`, illustrated in Figure 3, a user sends a content



(a) (b)

(c)

Figure 3: `TraC` operation: (a) Content request: (1) a user sends a content request to the AP connected, (2) this AP forwards the content request to the other AP within the trajectory of the vehicle toward the destination D; (b) Content download: APs download the content requested beforehand; (c) Content retrieval: (5) AP promptly satisfies the user request because the content is already available in cache.

request to the first AP connected and then this request is forwarded to all other APs along the user trajectory toward the destination D. Upon receiving a request, the AP download the content beforehand, promptly satisfying the user request as her vehicle arrives. It is worth mentioning that content can also be downloaded from V2V opportunities. Hence, whenever a nearby vehicle receives a content request, it can also send the content if it is already available in cache.

3.1. Assumptions

We assume in this paper that vehicles have one onboard IEEE 802.11p interface and a GPS. APs, in addition to the IEEE 802.11p interface, have a wired connection to each nearby AP and an independent interface to provide Internet access. Two APs are considered neighbors if they are within the same subnetwork, i.e., if they are connected to the same switch or connected by a wired link. The onboard wireless interface of vehicles is used by users to request content to APs and other nearby vehicles. If the content is cached, the AP or the vehicle receiving the request sends the content through the wireless interface to the requesting user. Wired connections between APs are only used to forward users' requests and, as a consequence, contents are never forwarded between APs. Missing contents are downloaded by APs using the available Internet connection.

We assume that an AP knows its geographical position and the position of all the APs in its vicinity. Its geographical position is static and can be easily configured during AP deployment. APs within its vicinity, on the other hand, are obtained through the proposed neighborhood discovery protocol, introduced in Section 4.2. In this work, all vehicles and APs are nodes of a CCN network, and as a consequence, each one has a FIB, a PIT and a CS. Also, all nodes perform promiscuous content caching and, in addition, APs do proactive caching. As in CCN, however, we consider that nodes have limited storage and therefore a caching policy must be used. The idea is to maintain only popular content in cache, so as to assure a high cache hit rate. If, however, a given content is not available on the node vicinity, it is requested and downloaded on demand up to the content source, if needed.

3.2. Control packets

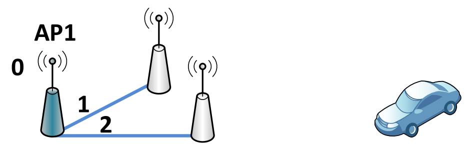
The proposed forwarding strategies, TAF and DMF, require two specific control packets called *Vehicular Interests* (`V-INT`) and *Access Point Interests* (`AP-INT`). `V-INTs` are sent by users to request content. On the other hand, `AP-INTs` are used by APs for either neighborhood discovery or vehicular interest forwarding. The former is used whenever an AP wants to discover the APs within its vicinity, whereas the latter is used to forward vehicular interests to neighbor APs within users' trajectories.

Vehicle trajectory is informed based on a 3-tuple in the header of `V-INTs`. This 3-tuple stores the user current geographical position, the geographical position of her destination, and the estimated time of arrival to the destination. Since we assume that APs know their own position and the position of their neighbors, they can forward `V-INTs` encapsulated in `AP-INTs` to neighbor APs. Therefore, the content requested by a user can be forwarded to the APs along her trajectory, even before she gets there. The current position, destination, and estimated time of arrival at the destination can be obtained from the GPS of the vehicle or a user smartphone.

3.3. Interest Naming Schemes

In CCNs, interest and content naming schemes play a crucial role since packet routing is based on names. In this work, a `V-INT` has the format: `\v-int\contentID\chunkID`. The first component (`v-int`) defines the vehicular scope; the second component (`contentID`) identifies the content requested by the user; and the third component (`chunkID`) identifies a specific content chunk. Hence, when a content is requested, a `V-INT` is generated for each chunk. Also, we define the naming format of `AP-INTs` as: `\ap-int\protocolID\outInterfaceID\apID\ap-intID`. The first component (`ap-int`) defines the scope of the `AP-INT`, limiting its use to the network between APs. The second component (`protocolID`) indicates the utilization of the `AP-INT`, which can be either for interest forwarding or neighborhood discovery. The third component (`outInterfaceID`) defines the outgoing interface used to forward an `AP-INT`. Finally, the fourth component (`apID`) identifies the AP that generated the `AP-INT` and the fifth (`ap-intID`) identifies the `AP-INT` itself.

Before describing interest forwarding in `TraC`, it is important to understand the FIB (Forwarding Information Base) structure, as illustrated in Figure 4. On the one hand, vehicles have one entry (`\v-int`) in their FIBs to forward `V-INTs` through the wireless interface. On the other hand, APs have two entries to each network interface connecting APs, in addition to the entry for `V-INT` forwarding. These entries neighbor APs follow the rule `\ap-int\protocolID\outInterfaceID\apID`, which changes the second component according to `AP-INT` utilization. If `AP-INT` is used for neighborhood discovery, the `protocolID` becomes equal to `nd`. If, however, `AP-INT` is used for interest forwarding, then the `protocolID` is `fw`. Note that independent of the utilization, `AP-INTs` are always sent hop-by-hop, from one AP to its neighbors.



Access Point FIB	
Prefix	**Out Interface**
\v-int	0 (wireless)
\ap-int\fw\1\AP1	1 (between APs)
\ap-int\nd\1\AP1	1 (between APs)
\ap-int\fw\2\AP1	2 (between APs)
\ap-int\nd\2\AP1	2 (between APs)

Vehicle FIB	
Prefix	**Out Interface**
\v-int	0 (wireless)

Figure 4: FIBs example: AP with one wireless interface and two wired interfaces between APs; vehicle with one wireless interface.

3.4. Data structures

Data structures are used to store pending Internet content requests. These structures are important because they control the Internet access of vehicles, giving priority and choosing which content chunks must be downloaded first. In `TraC`, instead of a single FIFO (First-In First-Out) data structure to handle all the incoming chunk requests, we assigned higher priority to requests from vehicles directly connected to the AP. In this case, we divide the pending download data structure in two to avoid losing contact opportunities. APs have then a higher-priority data structure to store `V-INTs` received from vehicles connected to its wireless interface, i.e., directly connected, and a structure to store `V-INTs` received from forwarded `AP-INTs`. Note that a single data structure would not suit well `TraC` because APs can download chunks for users directly connected and also for users that may possibly pass by in the future. Since there are no guarantees that this distant user will ever pass by, downloading a chunk for her before another one requested by a user directly connected, may lead to contact opportunity losses.

The high-priority data structure for directly-received `V-INTs` could use either FIFO or LIFO policies. Although FIFO is more traditional, in vehicular networks, topology changes may occur very often. As a consequence, assigning higher priority to more recently received requests may lead to a higher success probability since requesting vehicles can be still in range. In this case, using a LIFO structure is preferable in the vehicular scenario (We evaluate these two possibilities in Section 6.).

For the low-priority structure, we use a queue which sorts pending Internet content requests according to the user estimated time of arrival at a given AP ($ToA(AP)$). Therefore, requests of a user arriving sooner at the AP are served before those requests from more distant users. The $ToA(AP)$ is estimated from the information contained in `V-INT` (the user current geographical position, the geographical position of her destination, and the estimated time of arrival to the destination), as follows:

$$ToA(AP) = \left[\left(\frac{dist\,(p(user), p(AP))}{dist\,(p(user), d(user))} \right) * (ToA(d) - T(now)) \right] + T(now), \tag{1}$$

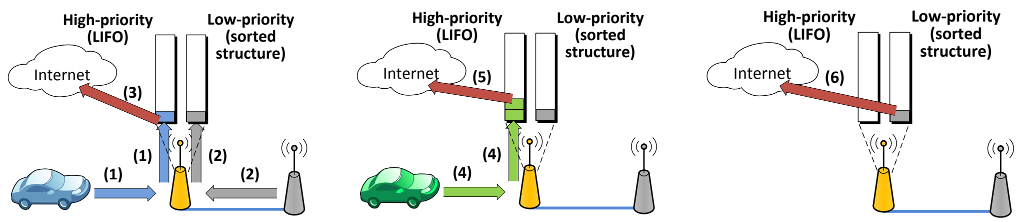
where $p(user)$, $d(user)$, $ToA(d)$, $T(now)$, $p(AP)$ and $\texttt{dist}(\bullet)$ denote, respectively, the user's current position, her destination coordinates, the estimated time of arrival at the destination, the current time, the coordinates of the AP, and a function to calculate the geographical distance between two points. Note that Equation 1 computes the proportion of time needed a node needs to arrive at an intermediate AP between its current and final positions. If at any given time, $ToA(AP) \leq T(now)$, that request is discarded because the user who has sent it has already arrived at the destination.

Considering the data structure used in the forwarding strategies, the AP first chooses chunks from the high-priority structure to download from the Internet. Only if this structure is empty, chunks from the low-priority structure is scheduled for download. All downloads are modeled without preemption, hence even higher priority requests can not interrupt an ongoing download. Figure 5 illustrates the operation of the proposed data structures. It is worth mentioning that if two or more vehicles are directly connected to an AP and all of them send a `V-INT`, the first vehicle to be satisfied is the one that sends the last `V-INT`, if considering that all requests were added to the high priority queue. If, however, the first `V-INT` received trigger the download before the other `V-INTs`, it will not be stopped as we do not consider preemption. This situation could happen if the high priority queue is empty as the first vehicle arrives.

As we can see, `TraC` design aims at maximizing the number of `V-INTs` satisfied in three ways. First, `TraC` gives priority to vehicles directly connected to APs, assuring one can always request content on demand. Secondly, `TraC` proactively caches contents, raising APs utility when no one nearby is requesting contents. Finally, `TraC` caches popular content in every network node.

4. `TraC` Operation

This section shows in more details how users inside vehicles request and receive content, and how APs operate to deliver Internet content to users.



(a) (b) (c)

Figure 5: Data structures: (a) a user sends a content request to the AP connected. The AP pushes the request to the top of the LIFO (1). An AP forwards a content request to the AP. The AP adds the forwarded request to the low-priority queue (2). The AP downloads the first chunk in the LIFO (3). (b) Another user connects and sends two content requests to the AP. The AP pushes these requests to the top of the LIFO (4). The AP downloads first the two content requests in the LIFO structure (5). (c) The AP only downloads the forwarded content request received from the other AP after the LIFO structure becomes empty (6).

4.1. User node operation

Infotainment content is typically large and may be fragmented into multiple chunks. Consequently, the user application sends `V-INTs` for each chunk composing the entire content. Despite transparent to the user, each `V-INT` sent generates a new entry in the PIT (Pending Interest Table). Only after creating such entry, the `V-INT` transmission through the wireless interface is scheduled according to the rules in the FIB. Since in vehicular and mobile scenarios one cannot guarantee that a request is always satisfied, it may be necessary to retransmit unsatisfied `V-INTs` later. The process followed by the user node when it receives a packet is presented as a flowchart in Figure 6.

After delivering a chunk to the application, the corresponding `V-INT` is removed from the data structure storing pending content requests. Hence,

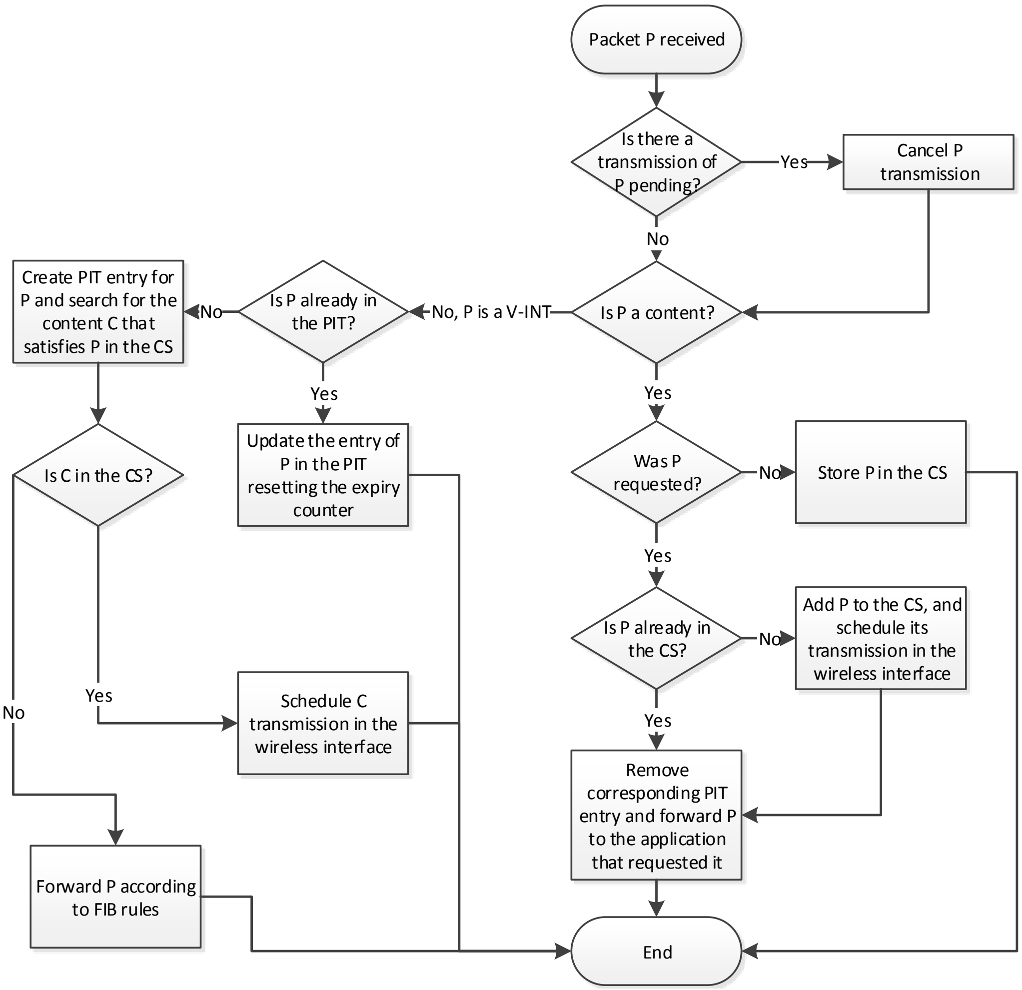


Figure 6: Flowchart of the user node operation when receiving a packet.

upon receiving all the chunks of a given content, the content is reassembled and delivered to the user. By the flowchart in Figure 6, we can see that chunks not requested by the user are stored, using promiscuous caching. Note that only requested content is retransmitted to minimize the wireless channel congestion. Next, we describe APs operation.

4.2. Access Point operation

Without proactive caching, APs would only download Internet content that are requested by users directly connected to the wireless interface. This strategy, however, coupled to short contact durations, may hinder content delivery. We then propose that APs forward user requests to other APs. In this case, however, because forwarding every `V-INT` to all APs is inefficient, we propose a forwarding strategy to send `V-INTs` only to the APs more likely within each user trajectory. To accomplish that, we assume that APs are aware of each user current position and destination. The former can be obtained by AP proximity, whereas the second is informed at the packet header. Since packets do not provide the whole trajectory, to avoid increasing the packet size, `V-INTs` can be forwarded to areas where the user will not pass by. In this case, retransmissions of unsatisfied `V-INTs` represent a new opportunity for the forwarding strategy to better match each user trajectory.

When receiving a `V-INT` through the wireless interface, the AP schedules the retransmission of that `V-INT`, extracts the information inserted in the header and adds the `V-INT` in a data structure that stores pending content requests. Next, the AP forwards the `V-INT` encapsulated in an `AP-INT` to other APs according to the forwarding strategy. It is worth noting that the forwarding `AP-INTs` are not "satisfied", and simply expire after reception, because they do not have content associated to them. To choose the AP to forward the `V-INT`, one AP must know its neighbors. As a consequence, we develop the Neighborhood Discovery Protocol, mixing the content-oriented paradigm with CCN.

Neighborhood Discovery Protocol: The proposed protocol allows each

AP to know the position of its neighbors and the network interface through which each neighbor is reachable. This information, in addition to FIB configuration, allows the geographical `V-INT` forwarding.

The neighborhood discovery protocol begins as the AP is turned on. The AP sends discovery `AP-INTs` in all network interfaces connecting to other APs. When an AP receives a discovery `AP-INT`, it answers the request with its own ID and geographical position. The answer name is set equal to the name of the discovery `AP-INT`, which contains the interface used for transmission. Then, for each neighbor discovered, the AP registers the neighbor ID, position, and the interface used to reach that neighbor. Neighbor ID and position are obtained from responses to the discovery request while the interface used is retrieved from the response name. Once the neighborhood is discovered, APs are capable of deciding to which of its neighbors a `V-INT` encapsulated as `AP-INT` must be forwarded according to a predefined forwarding strategy.

Trajectory-aware forwarding strategies: We present two forwarding strategies developed, the Triangular Area Forwarding (TAF) and the Distance Minimization Forwarding (DMF), as follows.

- *Triangular Area Forwarding (TAF):* The rationale of this strategy is to use a geometric shape to delimit the area where `V-INTs` are received and forwarded. This area must coincide as much as possible with the vehicle trajectory to destination. Nevertheless, the probability of correctly predicting a vehicle position decreases as the distance to the last known position grows. As a consequence, the farther the distance, the higher the number of APs that must be considered, even though we assume that vehicles likely use the shortest possible trajectory toward its destination which can be approximated to a straight line. For these reasons, we choose the triangle as the delimiting shape, connecting the last known vehicle position to the destination, as a flashlight would do if we could point out one from the last known position to the vehicle destination.

In the Triangular Area Forwarding (TAF) strategy, an AP creates a coverage triangle whenever it receives a `V-INT` through the wireless interface. The coverage triangle is defined by the position R of the AP that received the `V-INT`; the geographical destination D of the vehicle that has sent the `V-INT`; and an opening angle α. The coverage triangle vertices are then R, D_1, and D_2, where R is known (AP position), but the other two vertices must be calculated. The opening angle α defines the angles $\angle D_1RD$ and $\angle DRD_2$. Given these angles and considering an isosceles coverage triangle ($\overline{D_1R} \cong \overline{D_2R}$), the position of D_1 and D_2 can be calculated, forming the triangle $\triangle RD_1D_2$. An example of coverage triangle is shown in Figure 7.

To calculate vertices D_1 and D_2, first we obtain the distance d from D to R. For the angles $\angle RDD_1$ and $\angle RDD_2$, we need to find the vector perpendicular to the vector from D to R. For this, we first calculate the unit vector that points from D to R, called $\vec{v}$, which is of the form $\vec{v} = (v_x, v_y)$. Then we compute the vector perpendicular to $\vec{v}$, $\vec{u}$, as $\vec{u} = (-v_y, v_x)$. With $\vec{u}$ pointing to the direction needed and the

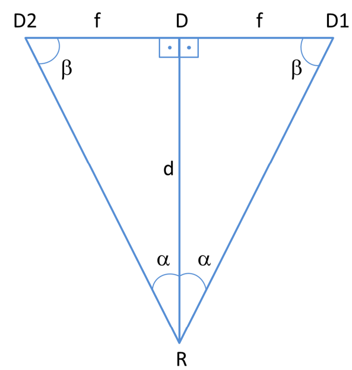


Figure 7: Coverage triangle $\triangle RD_1D_2$, where R is the position of the vehicle and D is the position of the destination.

distance d, we can obtain the distance f between D and D_1 as:

$$f = tan\,(\alpha) * d. \tag{2}$$

With the position D, the vector $\vec{u}$, and the distance f, we can calculate D_1 and D_2 using Equations 3 and 4, respectively.

$$D_1 = D - f * \vec{u}. \tag{3}$$

$$D_2 = D + f * \vec{u}. \tag{4}$$

After defining the coverage triangle, the AP creates an `AP-INT` containing the vertices of the triangle and the `V-INT` received and forwards it to other APs. The forwarding process of `AP-INTs` in the triangular area is based on two rules: if an AP is inside the coverage triangle, it forwards the `AP-INT` to its entire one-hop neighborhood. Otherwise, it forwards the `AP-INT` just to its one-hop neighbors that are inside the coverage triangle. This is possible because the AP knows the triangle and the geographical position of its neighbors. The forwarding process is initiated whenever an AP receives a `V-INT` through the wireless interface. This AP is the vertex R of the coverage triangle and, by definition, forwards the `AP-INT` to all of its neighbors. The detailed flowchart of this forwarding process can be seen in Figure 8.

As the flowchart in Figure 8 shows, when an AP receives an `AP-INT` and checks that itself is inside the `AP-INT` coverage triangle, it adds the encapsulated `V-INT` to the structure storing pending Internet content requests. This is because the AP is inside the area where the requesting user is expected to pass by along her way to destination. Also, each AP verifies if the received `AP-INT` is in the list of last seen `AP-INTs`. `AP-INTs` already on the list are not forwarding to avoid duplicates. The `AP-INT` list maintains the last K different `AP-INTs` received.

The triangular forwarding strategy has the advantage of covering a broader area. Nevertheless, it may introduce `AP-INT` replicas. To minimize this issue, we define next a forwarding strategy that chooses the AP to forward an `AP-INT` according to the minimum distance to the destination. Even though more susceptible to coverage failures, this second strategy produces fewer `AP-INT` transmissions and possibly fewer replicas.

- *Distance Minimization Forwarding (DMF):* The purpose of this strategy is to reduce the number of APs to which `AP-INTs` are forwarded

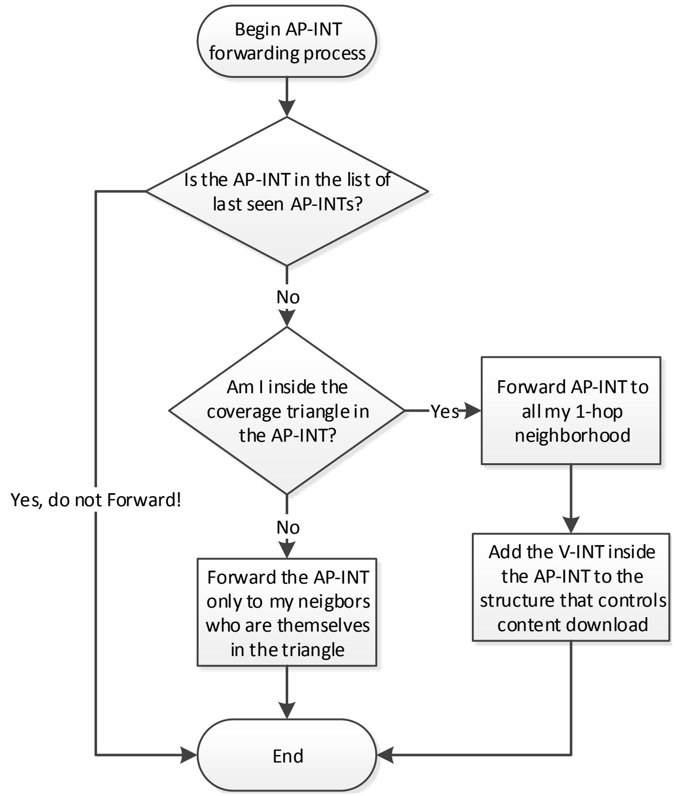


Figure 8: Flowchart of the `AP-INT` forwarding process in triangular areas.

and, consequently, the number of APs requesting the same content. The number of forwardings becomes lower because there are no more replicas and because the DMF strategy forwards `AP-INTs` to at most one interface per hop.

Upon receiving a `V-INT` through the wireless interface, the AP selects the neighbor AP closest to the destination to forward the produced `AP-INT`. If the selected neighbor is farther than the AP itself, the forwarding process ends. An AP receiving a forwarded `AP-INT` repeats the same procedure or stop forwarding. Unlike the triangular coverage forwarding strategy, all nodes receiving the `AP-INT` insert the `V-INT` encapsulated in the data structure of pending content requests.

Note that in both trajectory-aware forwarding strategies, i.e., TAF and DMF, privacy issues may be raised. Nevertheless, it is worth mentioning that as `TraC` relies on the CCN framework, only contents are identified. The identification of users based on trajectories requires additional information not available in `TraC`. Hence, considering that the trajectory privacy is a concern, we have indeed an issue as `TraC` relies on anonymous users' destination. Note, however, that the users' trajectory is never advertised in `TraC`, it can only be computed as an estimation from users' destination. Hence, we can fix the problem by obfuscating such information. This could be done by adding noise to the position of users' destination, as in `TraC` such precise information is not needed. If even after adding noise to users' destination the sequence of RSUs remains the same, we can at the same time reduce the privacy issue and maintain `TraC` performance without precise trajectory information. Even though a solution is possible, we do not touch this problem in this paper.

Content reception: APs either request chunks to or receive from the Internet. Nevertheless, an AP can receive a content chunk through the wireless interface if it promiscuously overhears chunk transmissions or if, upon retransmitting a `V-INT`, there is a user that possesses the requested chunk within its range. If a content chunk is received through the wireless interface, the AP checks if there is a PIT entry for that chunk and if the chunk is already in the CS. If it is not, it is added and have a retransmission scheduled to the wireless interface. In addition, we check if the chunk being downloaded from the Internet matches the chunk that has just been received from the wireless network. If it does, the download is canceled and the download of the next chunk is scheduled, according to the data structure of pending Internet content requests.

In the case of a content chunk being received from the Internet, again, APs check if there is a PIT entry for that chunk and if the chunk is already in the CS. If a matching request is present in the PIT, the chunk is scheduled for transmission in the wireless interface. It is worth mentioning that for a PIT entry to exist, a `V-INT` must have been directly sent to the AP by a vehicle. Forwarded `V-INTs` encapsulated inside `AP-INTs` are requested to the Internet, even though they do not create PIT entries. This procedure avoids the transmission of content chunks received from the Internet in the wireless channel by APs where the vehicle is not necessarily reachable. This way, the used proactive strategy caches the content in the APs before the user arrives. Delivering the content, however, requires the user to first request the content directly to the AP or to a vehicle that has the requested content cached.

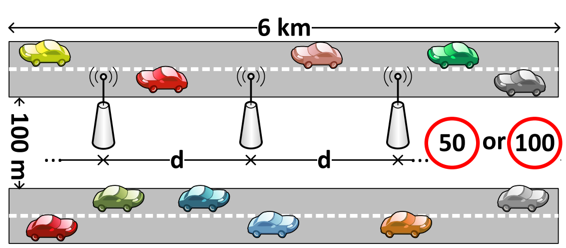
5. Simulation Settings

This section describes the different scenarios and traffic patterns used in our simulations (Section 5.1). In addition, we also introduce the network parameters and the data structures used (Section 5.2).

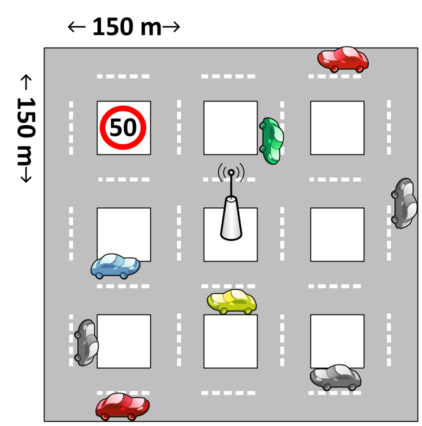
5.1. Scenarios

The proposed `TraC` strategy was implemented in the ndnSIM simulator. The performance was assessed using two fundamental scenarios, *Highway* and *Urban*, and a more realistic one called *Cologne*. The first two were created using SUMO (*Simulation of Urban MObility*), while the third one was obtained from the Cologne dataset [9]. It is worth mentioning that in all scenarios, when a vehicle reaches its destination, it is removed from the simulation and no longer receives or sends packets.

In the highway scenario, we set the maximum speed allowed to 50 km/h and 100 km/h to evaluate the influence of speed in content delivery. The highway scenario is composed of two double lane highways of 6 km in opposite directions. Between the highways there is a 100 meters gap, where APs are uniformly distributed (Figure 9(a)). When the simulation starts, one vehicle per second is added to the scenario until the total number of vehicles is reached. Vehicles are added at the edges of the highway with equal probability. Simulation finishes after all vehicles have gone through the entire highway.



(a) Highway scenario.



(b) Urban scenario segment.

Figure 9: Scenarios produced with SUMO.

In the urban scenario, streets have a maximum allowed speed of 50 km/h. This scenario has a total area of 2,250 m $\times$ 2,250 m, divided in 15 $\times$ 15 Manhattan-like grid, where each block is 150 m wide and 150 m long. The APs are uniformly distributed forming an equally spaced grid in the X, Y plane. An example of an urban scenario composed by a 3 $\times$ x 3 Manhattan-like grid can be seen in Figure 9(b). Similarly to the highway scenario, when the simulation starts, one vehicle is added to the scenario per second until the total number of vehicles is reached. In the urban scenario, however, the initial and the final position of the vehicles are randomly distributed in the simulation area. Simulation finishes after all vehicles reach their destination.

Finally, we took a fraction of the Cologne dataset [9] to evaluate the proposed forwarding strategies under a more realistic road structure and traffic conditions. The Cologne dataset is a synthetic trace of vehicles at the city of Cologne, provided by the German Aerospace Center (DLR). The dataset covers a period of 2 hours from 6 am to 8 am, including rush hours. We used a 10-minute period during the rush-hour of an area that has both a highway and an urban road structure (Figure 10). The fraction of the dataset used has an area of 4.72 km^2 in which 688 individual trips occur. As well as in the urban scenario, the APs are uniformly distributed forming an equally spaced grid in the X, Y plane.

5.2. Parameters and configurations

Since applying CCN to vehicular networks requires adaptations, we modified CCN to use the collision-avoidance timer and the application retransmission timer present in [6]. The first timer cancels packet transmission if a node hears another node transmitting the same packet, while the last

Figure 10: Cologne scenario.

timer retransmits unsatisfied interests at the application layer. `TraC` is implemented on top of the adapted CCN version.

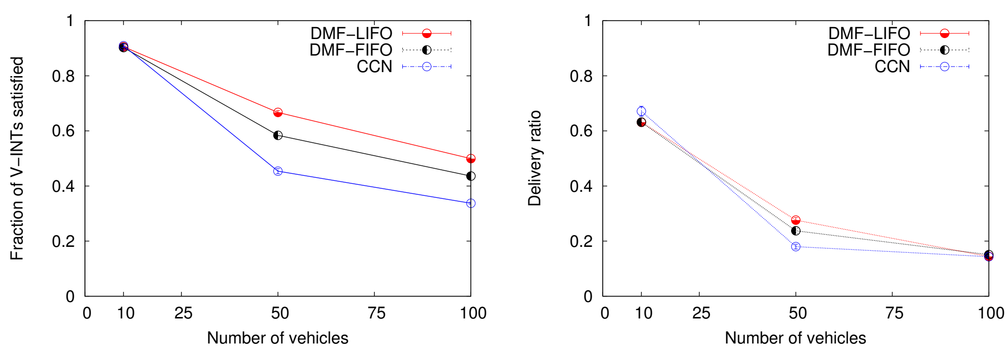
Both simulated vehicles and APs use 802.11p wireless interfaces configured with Minstrel rate control algorithm. The wireless channel is set with constant speed propagation delay model, three-log-distance propagation model, and the Nakagami-m propagation model. The point-to-point interfaces connecting APs have bandwidth of 5 Mb/s and propagation delay of 2 ms. Assuming that in a large city with 1000 km^2, the aggregated Internet bandwidth for the vehicular access is 1 Gb/s and that there are at least 4 APs per km^2 for content offloading, then we can assume that each AP is connected to the Internet through an independent link of 256 kb/s, unless otherwise specified. The `V-INTs` have the scope field equal to 2, two hops at most, to avoid broadcast storm in conjunction with the collision avoidance timer proposed in [6]. In addition, `V-INTs` expire after 10 seconds in the PIT and are retransmitted by the application after 11 seconds, if they have not been satisfied. We set this timer to expire one second later to avoid premature retransmissions. In the simulation, there are 1,000 distinct contents that are divided into several chunks of 1,500 bytes. Users request a new content on average every 10 seconds, according to an exponential random variable [21]. The requested content follows a Zipf distribution with parameter 0.8, this distribution is the content popularity according to the empirical analysis in [22]. All network nodes perform promiscuous caching, storing content even without registered interests. Nevertheless, only the APs running `TraC` do proactive caching, requesting Internet content along users' trajectories. This requires `V-INT` forwarding, which is executed only by `TraC` APs. All network nodes use the Least Recently Used (LRU) caching policy.

The size of the pending Internet download data structures is limited to 10,000 requests to avoid overflows. When using `TraC`, the sizes of the higher and lower priority structures are 300 and 9,700, respectively, producing the same limit imposed to the adapted CCN.

6. Results

In our results, `DMF` refers to `TraC` using the distance minimization forwarding, `TAF` refers to `TraC` with the triangular area forwarding, and `CCN` is the adapted version of CCN to the vehicular scenario. Note that in the adapted CCN version, APs neither perform proactive caching nor forward `V-INTs` to neighbors. Also, in the adapted CCN, APs use a single FIFO structure to handle all incoming content requests. Whenever applicable, results show a vertical error bar representing a confidence interval of 95% computed over 30 simulation runs.

We have first conducted preliminary experiments to verify which data structure would better serve user requests from those proposed to store directly received `V-INTs`: LIFO or FIFO. We used the urban scenario and varied the number of vehicles from 10 to 100, with 36 APs uniformly distributed in the scenario. Content sizes are set to 180 kb and, consequently, any content is made of 120 chunks, each one producing one distinct `V-INT`. We measured the fraction of `V-INTs` satisfied and the delivery ratio, comparing the adapted CCN, which uses a single FIFO structure, with the DMF strategy using a FIFO (DMF-FIFO) and using a LIFO (DMF-LIFO) data structures for high-priority chunk requests.



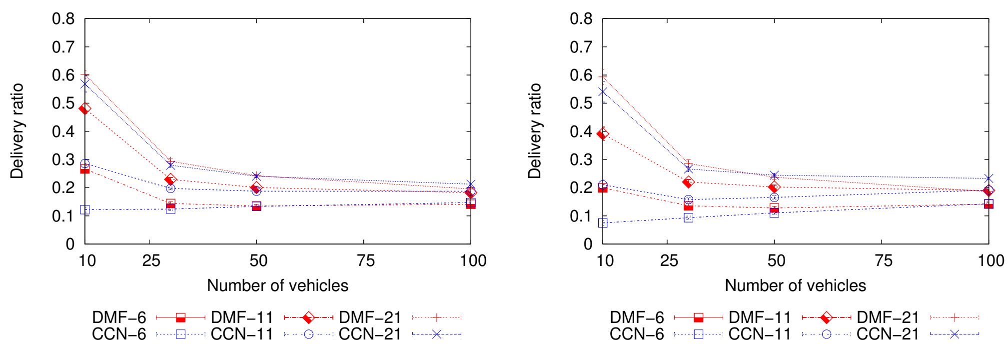
(a) (b)

Figure 11: (a) Fraction of `V-INTs` satisfied varying the number of vehicles; (b) Content delivery ratio varying the number of vehicles.

In Figures 11(a) and 11(b), we can observe that the use of the DMF strategy is more efficient in both the delivery ratio and the fraction of `V-INTs` satisfied compared with CCN. This result is independent of the data structure used. In addition, DMF-LIFO outperforms DMF-FIFO, confirming the intuition behind using a LIFO structure. Therefore, in all subsequent simulations, the pending Internet download data structure used by `TraC` is LIFO along with the low-priority queue for requests from users not connected to the wireless interface.

6.1. Highway Scenario

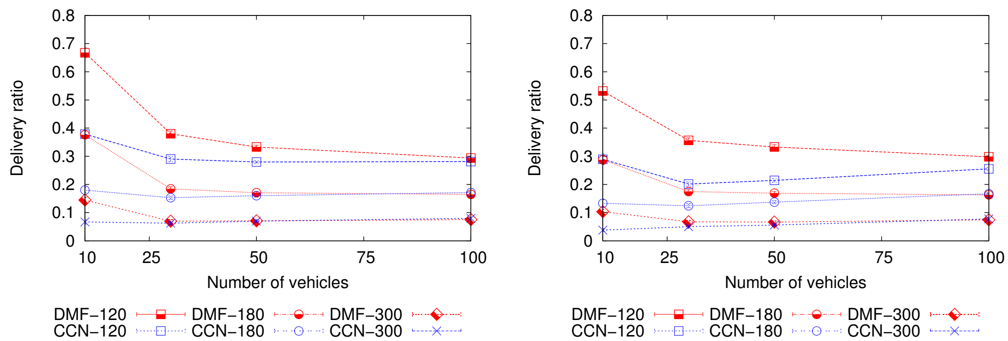
The first scenario aims at evaluating the performance of `TraC` in scenarios where vehicles can reach higher speeds and, at the same time, have less degree of freedom for their movements. The last characteristic makes movements more predictable. In the highway scenario, a user in her vehicle always passes by an AP where her content was scheduled because there is only one single straight line route. In this case, both DMF and TAF have the same performance, since the triangle and the minimum distance forward `V-INTs` hop-by-hop to the same set of APs until the closest one to the destination is reached. Thus, we only show the results of the DMF strategy in this scenario.



(a) Highway speed limit: 50 km/h. (b) Highway speed limit: 100 km/h.

Figure 12: Content delivery ratio varying the number of APs in the highway with fixed content size of 180 kB.

Figures 12 and 13 show the content delivery ratio achieved for an increasing number of vehicles. In Figure 12, the number of APs distributed along the highway varies from 6 to 21 (DMF-6 and CCN-6 to DMF-21 and CCN-21). For the same number of vehicles, we can see that the greater the number



(a) Highway speed limit: 50 km/h.

(b) Highway speed limit: 100 km/h.

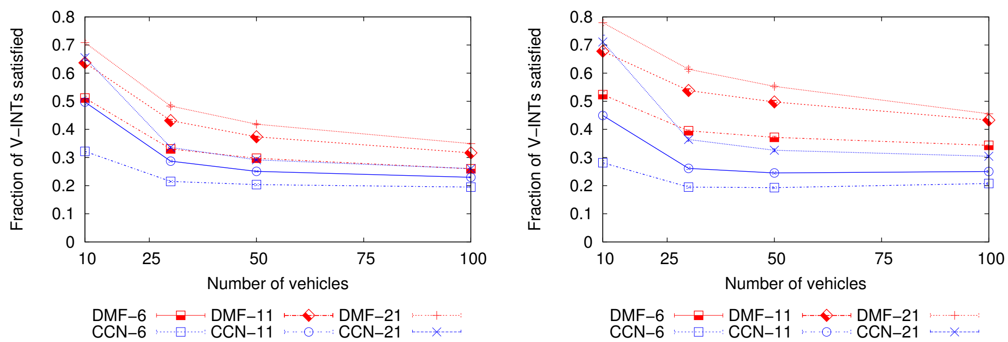
Figure 13: Content delivery ratio varying content sizes in kB with 8 APs distributed in the highway.

of APs, the higher the delivery ratio using or not `TraC`. This happens because the access network capacity becomes higher when more APs are used and, as a consequence, there are more opportunities for vehicles to retrieve content. Nevertheless, when comparing DMF with CCN, we can see that the impact on the delivery ratio is higher for `TraC` as the number of APs increases. This is because `TraC` can increase content availability proportionally to the number of APs. Additionally, when comparing the results of the highway with 50 km/h speed limit with the results of 100 km/h, we can see that DMF is less affected than CCN by the higher speed. At higher speeds, the contact duration is lower, leading CCN to a poor performance since vehicles have to request and wait for the content at the same AP. Hence, unless another vehicle has already requested the same content earlier and this is still in cache, contacts are less efficiently used. On the other hand, in DMF, the content will be proactively available in cache more often because of the forwarding strategies. This can provide a more efficient use of contacts.

In Figure 13, the size of the content requested by users varies from 120 to 300 kB (DMF-120 and CCN-120 to DMF-300 and CCN-300)[1]. The goal of this experiment is to evaluate the performance of `TraC` under different network loads. Thus, because users in our experiments request content on average each 10 seconds, the content size used is sufficient to simulate the network load of users requesting: small videos, e.g., TV commercials and vines; audios, e.g., music teasers or MP3 files; medium resolution images, e.g., nearby establishments and touristic points-of-interest; and text documents, e.g., cinema timetables and daily news. Our results show that, given the same number of vehicles, the larger the content, the smaller the delivery ratio. Such reduction is a consequence of a higher network load. In addition, when comparing DMF with CCN, we can conclude that the forwarding strategy makes the delivery ratio less susceptible to an increasing network load, showing only in the worst case the same performance of CCN. Figure 13 shows that increasing the number of vehicles, the delivery ratio achieved by DMF decreases as a consequence of a higher network load. Nevertheless, there is an increase of the CCN delivery ratio, resulting in a tie between CCN and DMF for higher number of vehicles. These results are explained by caching popular content in both strategies. Whereas in CCN a vehicle needs to count on a previous user requesting the same content, when using DMF, one vehicle can ask for the content and obtain it at the subsequent APs within its trajectory. Therefore, the caching efficiency depends on the number of vehicles and on the number of overlapping trajectories in CCN,

[1] The benchmark video Akiyo_cif, frequently used in the literature, has total size of 353 kB [23].

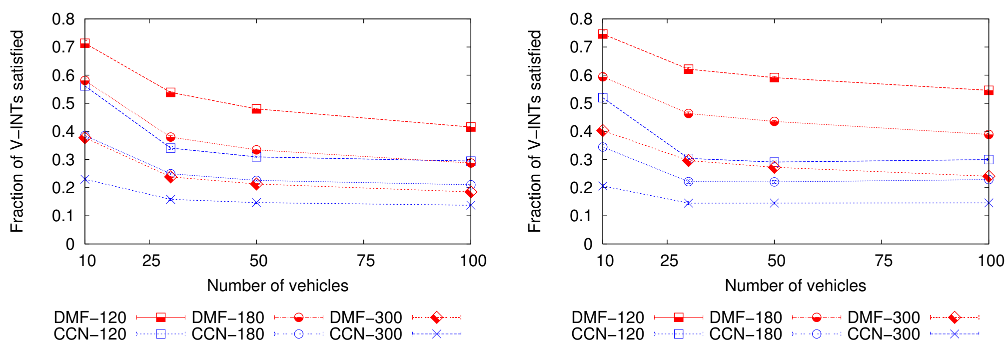
more than in DMF.



(a) Highway speed limit: 50 km/h.

(b) Highway speed limit: 100 km/h.

Figure 14: Fraction of satisfied `V-INTs` varying the number of APs in the highway with fixed content size of 180 kB.

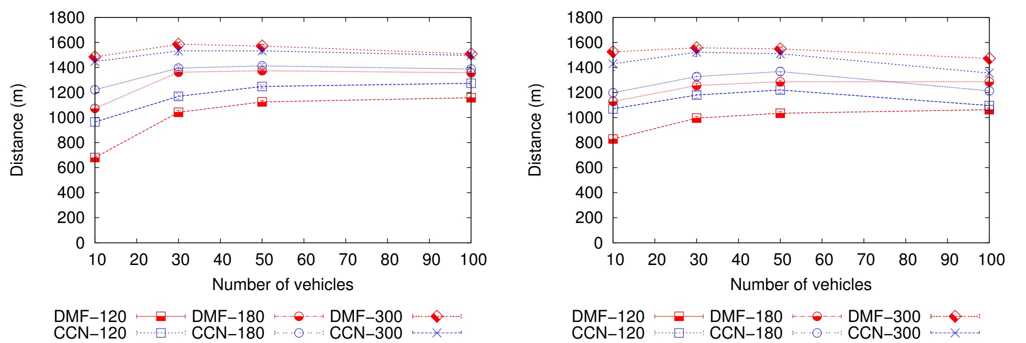


(a) Highway speed limit: 50 km/h.

(b) Highway speed limit: 100 km/h.

Figure 15: Fraction of satisfied `V-INTs` varying content sizes in kB with 8 APs distributed in the highway.

In urban scenarios, APs represent an alternative to provide Internet connectivity to vehicular users [13]. Therefore, it is worth evaluating the influence of the proposal in the fraction of satisfied `V-INTs` to observe the amount of traffic that could be offloaded from other networks, e.g., cellular networks [13]. Figures 14 and 15 show a behavior similar to the one observed for the delivery ratio. Increasing the access network capacity, the fraction of satisfied `V-INTs` grows. In opposition, if the network load increases, the fraction of satisfied `V-INTs` decreases. The benefits of using the proposed forwarding strategies, however, are more prominent for the fraction of satisfied `V-INTs` than for the delivery ratio. We can observe in Figures 14 and 15 that, on the one hand, the fraction of satisfied `V- INTs` drops a bit when using CCN at higher speeds but, on the other hand, it increases at higher speeds using DMF. This happens because higher speeds result in lower network load since vehicles go through the highway faster, further reducing the time to request content. Whereas DMF and TAF can benefit from lower network loads, CCN cannot do the same also as a consequence of the less efficient use of short-lived contacts.

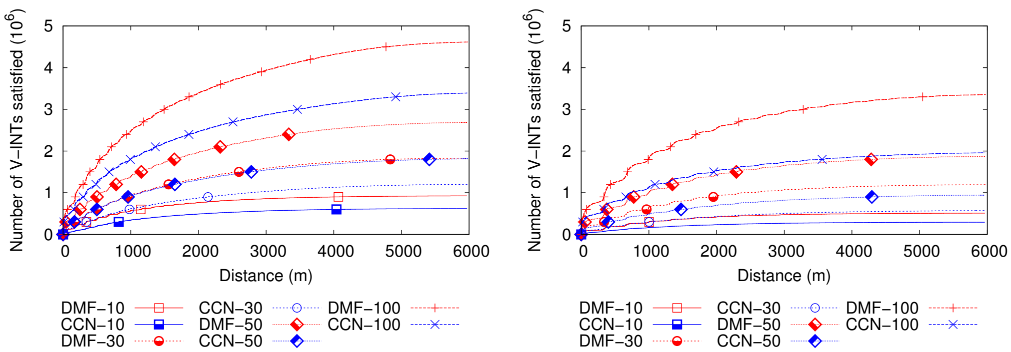


(a) Highway speed limit: 50 km/h.

(b) Highway speed limit: 100 km/h.

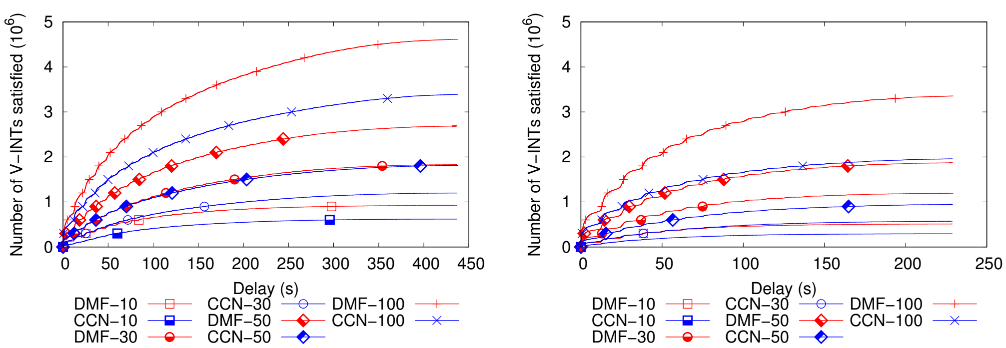
Figure 16: Average distance between the location where a `V-INT` was requested and the location where it was satisfied. We vary the content size in kB with 8 APs distributed in the highway.

Another metric of interest is how fast users' requests are satisfied. We



(a) Highway speed limit: 50 km/h. (b) Highway speed limit: 100 km/h.

Figure 17: Accumulated number of `V-INTs` satisfied according to the distance between where a `V-INT` was requested and where it was satisfied. We set content size to 180 kB with 8 APs distributed in the highway.



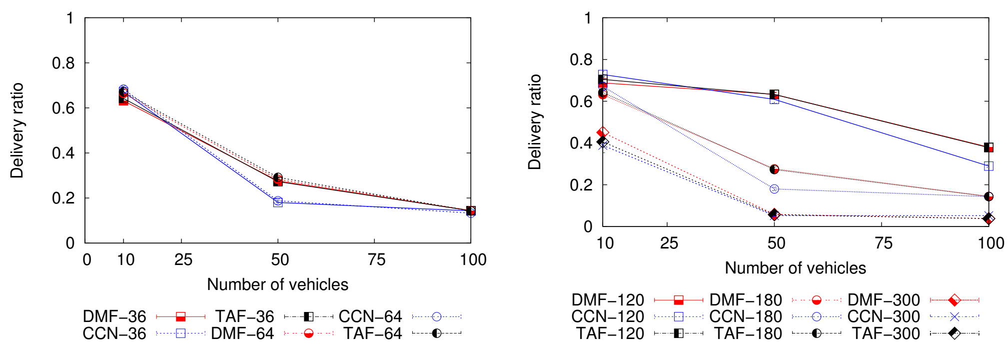
(a) Highway speed limit: 50 km/h. (b) Highway speed limit: 100 km/h.

Figure 18: Accumulated number of `V-INTs` satisfied according to the average delay. The delay in X-axis is calculated as the distance between where a `V-INT` was requested and where it was satisfied divided by the average speed of vehicles in the simulation. We set content size to 180 kB with 8 APs distributed in the highway.

analyze this metric by measuring the average geographical distance between the location where a content chunk is first requested and the location where it is finally received. Figure 16 shows the obtained results. We note that when the network load increases, the distance between requesting and receiving a content chunk also increases. With a higher network load, the data structures of APs (Section 5.2) are more populated and as a consequence a request will have to wait longer to be satisfied. We can also observe that DMF delivers content chunks faster than CCN on average for smaller content sizes. On the other hand, for larger contents, DMF delivers chunks slower than CCN on average. The key factor to understand this behavior is the number of satisfied `V-INTs` and how they are satisfied along users' trajectories with respect to the distance. To clarify this point, Figure 17 plots the absolute number of satisfied `V-INTs` per distance from the location where it was first requested until the location it was satisfied. From Figures 17(a) and 17(b), we can observe that when using DMF, more `V-INTs` are satisfied at any distance, be the distance small or large, which makes the average sometimes greater in Figure 16. Therefore, we conclude that using `TraC`, not only more `V-INTs` are satisfied, but also more `V-INTs` are satisfied faster and along users' trajectories even at larger distances. The same metric can be observed from the average delay perspective. To calculate the average delay until users' requests are satisfied we take the distances in Figure 17 and divide by the vehicles average speed, resulting in Figure 18. Vehicles average speed in the Highway scenario with 50 km/h speed limit is 13.66 m/s, while the average speed in the Highway scenario with 100 km/h speed limit is 26.06 m/s.

6.2. *Urban Scenario*

Although in the urban scenario vehicles do not reach speeds as high as in highways, their movements are more diverse. Hence, urban scenarios are less predictable than highways, permitting a better evaluation of `TraC` and the proposed trajectory-aware forwarding strategies for content retrieval. Note that unlike the highway scenario, it is possible that a user in her vehicle does not pass by an AP where her content was scheduled. We will evaluate both strategies proposed, DMF and TAF, and compare them with the adapted CCN.

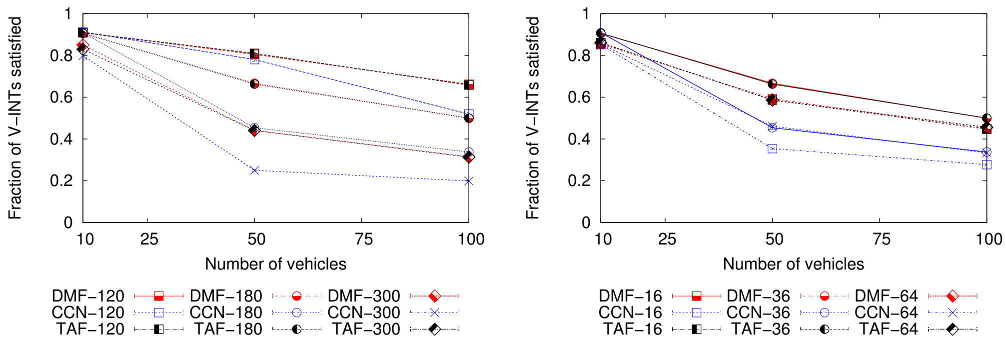


(a) (b)

Figure 19: Content delivery ratio: (a) Varying the number of APs in the urban scenario with fixed content size of 180 kB; (b) Varying content size in kB with 8 APs distributed in the urban scenario.

Figure 19 shows how the content delivery ratio is affected by different number of vehicles, number of APs (Figure 19(a)), and content sizes (Figure 19(b)). Results show that unlike the highway scenario, in the urban scenario, more APs do not necessarily mean higher delivery ratio. In the highway scenario, vehicles are distributed along a straight line, i.e., in 1 dimension. In the urban scenario, however, they are distributed over 2 dimensions. As a consequence, the number of vehicles inside the range of an AP is higher, even though the density of vehicles is maintained. More nodes then contend for the medium, leading to saturated conditions and possibly lower transmission rates. It is worth recalling that APs mainly send content chunks, whereas vehicles mainly send interests. This means that APs are prone to occupy the medium for longer periods than vehicles because chunks are typically bigger than interests. Hence, one must carefully choose the number of APs to maximize content dissemination along a user trajectory and, at the same time, avoid wireless medium congestion. We simulated different numbers of APs distributed over an urban scenario and concluded that the best results are achieved using 36 APs. This value is then used as a constant for this scenario as well as for the next one, the Cologne scenario, which has approximately the same area. When comparing DMF, TAF, and CCN, we can see that DMF and TAF results overlap. Moreover, we observe that the utilization of forwarding strategies makes the delivery ratio less affected by the number of vehicles. For instance, in Figure 19(a), with 50 vehicles, TAF and DMF strategies perform better than CCN.

Next, we evaluate the performance of the proposed strategies for different network loads. In Figure 19(b), the size of the content requested by users varies and we can note that for the same number of vehicles, the bigger the content, the smaller the delivery ratio, similarly to the highway scenario. In addition, comparing DMF and TAF with CCN, we conclude that the delivery ratio of CCN reduces faster with a higher network load than the forwarding strategies. As well as in the highway scenario, this result can be considered positive in most cases.



(a) (b)

Figure 20: Satisfied `V-INTs` fraction: (a) Varying content sizes in kB with 36 APs distributed in the urban scenario; (b) Varying the number of APs in the urban scenario with fixed content size of 180 kB.

We also evaluate the fraction of interests satisfied for the urban scenario. Figure 20(a) shows that increasing the number of vehicles and content sizes, the proposed forwarding strategies are able to deliver more content chunks to the user. Hence, we conclude that DMF and TAF are less affected by load variations and are able to provide better quality of service either for increasing network load or a higher number of users. Results of Figure 20(b) confirm that the higher number of APs has a negative impact on CCN, DMF, and TAF results due to the higher medium saturation. It is worth mentioning that we also simulated the scenario with 121 APs distributed and the results were even worse than those for 16 APs. We can also see in Figure 20(b) that independently of the number of APs, the use of the forwarding strategies outperforms CCN. Moreover, we observe again that the difference between the results of DMF and TAF is subtle. This small difference is because of the short retransmission timer used in simulations. Retransmissions of unsatisfied `V-INTs` give another chance to the forwarding strategies to predict the user trajectory correctly. For instance, if DMF fails to correctly predict the trajectory of a user, at each retransmission (11 seconds later), DMF has another chance to correctly predict. Finally, Table 1 compares the results of CCN with `TraC` and the respective forwarding strategies with 36 APs distributed in the scenario and content size of 180 kB. In the last column we can see that the use of the forwarding strategies results in gains of almost 50% when the number of vehicles grows, allowing better user content offloading.

Table 1: Comparison of the fraction of satisfied `V-INTs` with 36 APs and 180 kB content.

Number of vehicles	**Fraction of `V-INTs` satisfied**			$\frac{\textbf{DMF}}{\textbf{CCN}}$
	TAF	**DMF**	**CCN**	
10	0.91	0.91	0.91	1
50	0.67	0.67	0.45	1.49
100	0.50	0.50	0.34	1.47

Tables 2 and 3 present, respectively, the Internet network load requested by users and the maximum Internet network capacity. On the one hand, the Internet network load is computed as the sum of the size of all contents requested by users from the Internet during the simulation. Note that not necessarily all contents requested were delivered. On the other hand, the maximum Internet network capacity is computed as the maximum amount of data that the entire available infrastructure can download from the Internet during the simulation time. These results are obtained to better understand the low fraction of satisfied `V-INTs` and delivery ratio, observed in some settings. For example, with 100 vehicles and 300 kB content, CCN satisfies only 20% of the requested `V-INTs`, while TAF and DMF satisfies only 31%. Obviously, the maximum load would only be obtained if all the APs were downloading content during the entire simulation, which does not happen. In addition, this maximum capacity would only be harnessed by vehicles if the content downloaded were always of their interest, and if vehicles were always able to retrieve the content downloaded by APs. From Tables 2 and 3, we can see that with 100 vehicles and 300 kB content, even in ideal conditions, APs would not be able to download all the content requested. These results show that the greater the difference between network capacity and network load, the better the network performance. Nonetheless, as we have seen before, increasing the network capacity through the addition of APs does not always produce better results. The number of APs has a threshold after which adding more APs only degrade the network performance. Therefore, the other way to increase the network capacity is to increase the Internet bandwidth of each AP. This is investigated in the Cologne scenario in the next section.

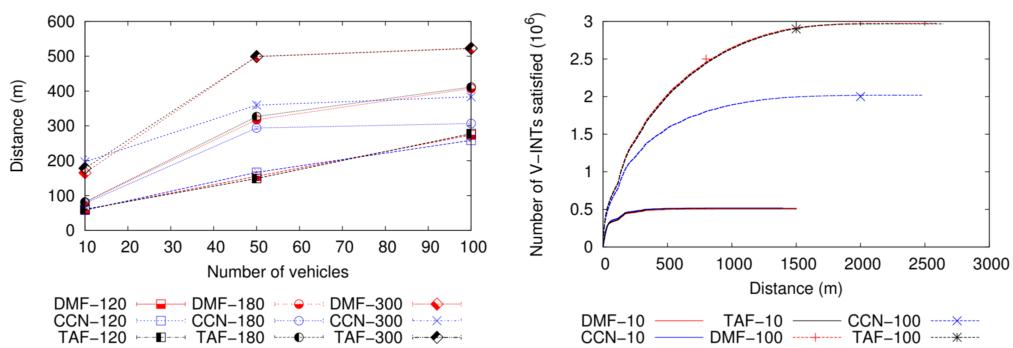
Table 2: Internet network load in the urban scenario.

Number of vehicles	**Average number of content requested**			**Internet network load (MB)**		
	Content size (kB)			**Content size (kB)**		
	120	**180**	**300**	**120**	**180**	**300**
10	159	158	160	19.08	28.44	48.00
50	876	874	874	105.12	157.32	262.20
100	1656	1658	1663	198.72	298.44	498.90

In Figure 21(a), to evaluate how quick `V-INTs` are satisfied, we vary the content size in kB with 36 APs distributed in the urban scenario. We then

Table 3: Maximum Internet network capacity in the urban scenario.

Number of vehicles	Simulation duration (s)	Internet bandwidth per AP (kb/s)	Number of APs	Maximum Internet network capacity (MB)
10	316	256	36	364.03
50	327	256	36	376.70
100	370	256	36	426.24



(a) (b)

Figure 21: (a) Average distance between where a `V-INT` was requested and where it was satisfied. (b) `V-INTs` satisfied according to the distance.

analyze the distance between the location where the content chunk was first requested and the location where it was satisfied. The results in Figure 21(a) show that when the network load increases, the distance between requesting and receiving a content chunk also increases, as already seen in the highway scenario. In addition, we also note that both DMF and TAF deliver content chunks slower than CCN for larger content sizes. The same behavior is observed in the highway scenario, and as also there, it happens because the forwarding strategies satisfy more `V-INTs` at small and large distances, which can produce a larger average. This can be seen in Figure 21(b) where we present the absolute number of `V-INTs` satisfied per distance from the point where they were requested for the first time until the point where they were satisfied. All the results so far show that TAF does not significantly outperform DMF while, as seen in Figure 22, the overhead inserted in the network between APs is much higher for TAF. Thus, we choose to only use DMF in the Cologne scenario presented next. The number of `V-INTs` satisfied per average delay metric can be calculated as the division of the distances in Figure 21(b) by vehicles average speed, which are 11.77 m/s and 11.48 m/s when 10 and 100 vehicles are in the urban scenario, respectively, as we did for Figure 18.

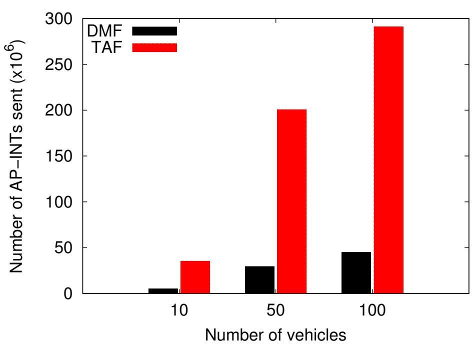
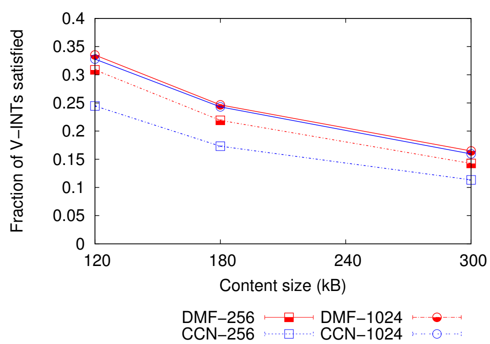


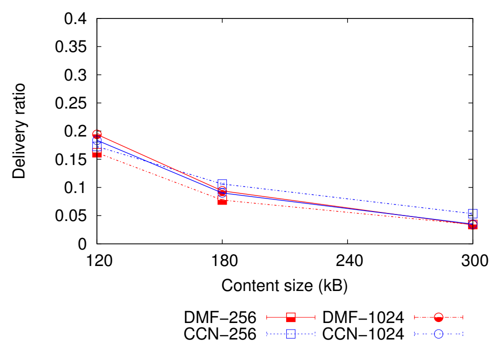
Figure 22: Forwarding strategies overhead: total number of `AP-INTs` sent over 30 runs of simulation.

6.3. Cologne Scenario

We use the Cologne dataset [9] to evaluate the strategies in a realistic road structure under rush-hour traffic conditions, i.e., high density scenario. The number of vehicles in the dataset used is 688 and the number of APs is set to 36, the same as the urban scenario, given their similar area. Content sizes vary from 120 kB to 300 kB. It is worth noting that since the number of vehicles is constant we change the X-axis to be the content size. To understand the effects of increasing the access network capacity to the Internet through the increase of APs Internet bandwidth, we use Internet links of 256 kb/s and 1024 kb/s. The first value is the one used in all previous simulations.



(a)



(b)

Figure 23: (a) Satisfied `V-INTs` fraction varying APs Internet bandwidth in kb/s; (b) Content delivery ratio varying APs Internet bandwidth in kb/s.

Figure 23(a) shows the fraction of satisfied `V-INTs`, while Figure 23(b) plots the delivery ratio. As we can see in Figure 23(a), when using the slower access link, the DMF strategy outperforms CCN by a larger margin than when using the faster access link. Nevertheless, the network capacity has increased four times, and the differences from DMF with smaller Internet bandwidth and DMF or CCN with larger Internet bandwidth are not that significant. For instance, the difference using DMF-256 and DMF-1024 is only 2.7%, and using DMF-256 and CCN-1024, the difference is only 2.4%. This indicates that the high density of nodes in the wireless medium affects the network performance in such a way that the network is not able to deliver more content to vehicles, despite the content already cached in APs. In Figure 23(a), CCN seems to better take advantage of the increased access link speed. This is because CCN has more room to improve contact efficiency, since its efficiency in using contacts depends more of the download waiting time, which decreases as the Internet bandwidth grows. We show in Tables 4 and 5 the Internet network load and the maximum Internet network capacity, respectively, to confirm that the wireless channel is the root of the problem. The tables show that using the 256-kb/s Internet AP links, the network, even in ideal conditions, would not be able to retrieve all the content requested. On the other hand, when using 1024 kb/s, the network is able to do it for the simulations of 120 kB and 180 kB content sizes. Nevertheless, the results of Figure 23(a) shows that the DMF with lower access link speed, although not capable of providing all the content, already reaches a performance close to 1024 kb/s as APs Internet bandwidth, which theoretically could support delivering all the content. It is worth mentioning that any kind of content prefetching, independent of the forwarding strategy, can likely provide benefits if the bottleneck of the network is between the APs and the Internet. Nevertheless, if the bottleneck is in the wireless medium content, prefetching may not result in any benefit [8]. As long as the wireless medium does not become the bottleneck, `TraC` scales better than CCN regardless of content size and number of nodes requesting contents.

In Figures 24(a) and 24(b), we evaluate how fast `V-INTs` are satisfied in the high density scenario using 256 kb/s and 1024 kb/s AP Internet bandwidths, respectively. The number of `V-INTs` satisfied per average delay met-

Table 4: Internet network load in the Cologne scenario.

Number of vehicles	Content size (kB)	Average number of content requested	Internet network load (MB)
688	120	9499	1139.88
688	180	9476	1705.68
688	300	9462	2838.60

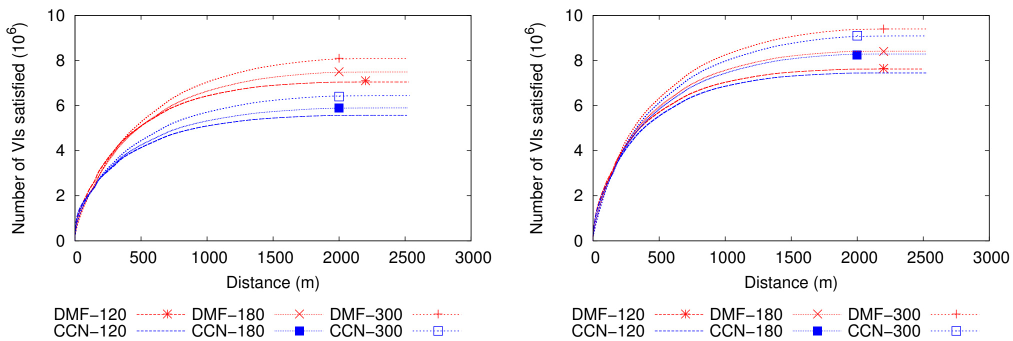
Table 5: Maximum Internet network capacity in the Cologne scenario.

Number of vehicles	Simulation duration (s)	Internet bandwidth per RSU (kbps)	Number of RSUs	Internet network capacity (MB)
688	600	256	36	691.20
688	600	1024	36	2764.80

ric can be calculated as the division of the distances in Figure 24 by vehicles average speed in the Cologne scenario, which is 14.27 m/s. In the first Figure 24(a), we observe that even the worst result of the DMF strategy is better than the best result of CCN. Results are better if we consider both how fast and how many `V-INTs` are satisfied. When the Internet bandwidth grows, the saturated wireless medium avoids `TraC` to improve even more its benefits. Nevertheless, the use of `TraC` still results in gains in both delivery speed and delivery ratio when compared with the adapted version of CCN.

7. Related Work

Works that employ CCN in vehicular scenarios usually take care of the medium access to avoid multiple transmissions of the same interest in a short period of time [6, 10]. Therefore, nodes schedule interest transmissions and



(a)

(b)

Figure 24: V-INTs satisfied according to the distance varying content sizes: (a) Internet bandwidth per AP = 256 kb/s; (b) Internet bandwidth per AP = 1024 kb/s.

listen to the medium during a random amount of time. If a node overhears another node transmitting the same scheduled interest, it simply cancels the transmission. The same holds for content chunks, which have also to avoid duplicated retransmissions.

Using the CCN architecture in vehicular scenarios has been typically investigated by comparison with the host-oriented paradigm. The first work in the area [15, 11] analyzed the viability of replacing IP with the CCN architecture in the vehicular scenario. Chen *et al.* [12] mainly focus on evaluating the influence of different caching policies in CCN nodes in vehicular scenarios. Wang *et al.* [15], on the other hand, developed a content naming scheme to vehicular networks, showing an application of traffic information dissemination as a use case. A TTL equal to 1 is set in interest packets to avoid forwarding. In addition, Wang *et al.* do not assume the use of APs to propagate the information. Hence, the only way a vehicle can receive information is encountering a vehicle acting as a data mule. `TraC` also uses content caching in APs to improve content delivery in vehicular networks.

Amadeo *et al.* [11] proposed the Content-Centric Vehicular Networking (CCVN) architecture, which is based on CCN with adaptations to the vehicular environment. In CCVN, interest packets are divided into two categories, `B-Int` (Basic Interest) and `A-Int` (Advanced Interest). A `B-Int` is sent to request the first content chunk. Meanwhile, nodes that have that chunk are discovered, identified, and stored in a structure called Content Provider Table (CPT). An `A-Int` is then sent to one of the nodes registered in the CPT to request the remaining content chunks. This approaches the host-oriented paradigm since individual nodes are contacted to retrieve the remaining chunks. CCVN does not use the Forward Information Base (FIB) of the CCN architecture. Authors argue that since nodes only have one wireless interface, the FIB would be useless. Nevertheless, the first assumption does not hold for any scenario. For instance, APs usually have more than one network interface. The work compares the CCVN architecture with TCP/IP in the vehicular scenario through simulation and evaluates the delivery ratio and delivery delay. Results show that the CCN based architecture outperforms the TCP/IP. Even though CCVN also adapts the CCN approach for vehicular networks, the architecture is not as content-oriented as `TraC`. In `TraC`, we do not need to individually identify nodes.

In [6], an application of traffic information dissemination is developed using the content naming framework proposed in [15]. As the wireless medium cannot guarantee collision detection and as transmissions are broadcast and hence do not offer any type of acknowledgment, the authors propose the use of the following timers: collision-avoidance timer, pushing timer, NDN-layer retransmission timer, and application retransmission timer. The first timer cancels packet transmission if a node overhears another node transmitting the same packet. The second timer prioritizes the transmission of nodes further away from the information producer. The third timer schedules message retransmissions at lower network layers, while the last timer retransmits unsatisfied interests at the application layer. Authors evaluate the developed application through simulation in ndnSIM [24], a content-centric network simulator implemented over the ns-3 network simulator. From the set of timers, the first and the last ones are used in `TraC`. The goal of `TraC` is different from this application, since we are interested on Internet traffic offloading and not on information dissemination.

The work [19] proposes three mechanisms to minimize the broadcast storm problem in CCN wireless mesh networks. The first, called Probabilistic Interest Forwarding (PIF), defines a probability to forward interest packets when a node is in high density areas. The idea is to avoid the broadcast storm problem, forwarding interests following CCN rules only in lower density areas. The second mechanism, Retransmission-Counter-based Interest Forwarding (ReCIF), adds one field to the header of interest packets to count the number of times the interest was forwarded. Whenever in a high density area, the node checks if the received interest was forwarded more times than a threshold before retransmitting. In case the counter exceeds the threshold, the interest is not forwarded. The last mechanism proposed is the combination of PIF and ReCIF. Results show that the mechanism can reduce the broadcast storm problem and increase content delivery with lower delay. `TraC` also tackles the broadcast problem by using the proposed trajectory-aware mechanisms.

Vehicular scenarios have short-lived contacts as a main concern. Thus, if contents are not retrieved fast enough, contacts can become inefficient or even useless. Previous works have already experimented network coding in VANETs [16, 14]. CCN, however, can retrieve contents faster than gossiping associated to network coding at the cost of possible broadcast storms caused by interest packets as stated more recently in [13]. Therefore, if we minimize the broadcast storm problem, CCN then becomes a better option for vehicular scenarios, as it raises contacts efficiency. Malandrino *et al.* [18] propose a fog-of-war mobility prediction model in addition to an architecture that integrates both cellular and IEEE 802.11 infrastructure. This architecture uses centralized servers to handle vehicle traffic information and to manage user queries. Authors evaluate the benefits of using mobility prediction models to prefetch content along users' trajectories. Because a network protocol was not defined, we could not compare this proposal with `TraC`. Malandrino *et al.* [18] uses a centralized architecture, whereas `TraC` handles both user queries and vehicle trajectory prediction using a decentralized architecture. In addition, we define all the network protocols needed for our proposal, which allows future performance comparisons.

As a summary, although there is already a number of studies proposing the utilization of CCN in vehicular networks, our work is the first one to explore the presence of APs as an alternative to increase the content delivery ratio [6, 10, 15, 17]. Moreover, the present work uses the information of vehicle trajectory to make the CCN network aware of users' destination, akin to host-oriented networks [8].

8. Conclusions and Future Work

This paper proposed `TraC`, a Trajectory-aware Content Distribution strategy, which uses proactive caching in Access Points (APs) to increase the probability of content delivery in vehicular scenarios. We have proposed two geographical strategies for vehicular interest forwarding between APs, and an additional neighborhood discovery protocol for content-oriented networks. The forwarding strategies were evaluated in three vehicular scenarios: highway, urban, and a rush-hour segment of the Cologne dataset, with respect to the fraction of interests satisfied, the content delivery ratio, and how fast content and interests are satisfied. We also evaluated the impact of an increasing number of vehicles, network load, and capacity on the performance of `TraC` and of the adapted version of CCN. Our simulation results show that the forwarding strategies of `TraC` always satisfy more interests than CCN, achieving gains of almost 50% in the number of `V-INTs` satisfied in the urban scenario. This shows that the decentralized strategies proposed benefit users in all scenarios. Our results also show that the forwarding strategies satisfy `V-INTs` faster than CCN, and along user trajectory even at larger distances. We also conclude that the larger the difference between network capacity and load, the better the performance of the strategies proposed. Additionally, the use of the forwarding strategies makes the delivery ratio and the fraction of satisfied `V-INTs` more independent with respect to the network load and vehicles speed. Also, the content caching efficiency becomes less dependent with respect to the number of vehicles and trajectory overlap. We observe that the DMF strategy performance is always close to the TAF performance. Since the second introduces a larger overhead in the network, TMF was the strategy of choice for the Cologne scenario.

As future work, we plan to extend the tests for other segments of the Cologne dataset. By enlarging the simulation area, it would be interesting to make vehicles send user trajectory checkpoints in `V-INTs` instead of only the final destination. As a consequence, routing strategies would be able to better track the current trajectory of the vehicle. Also, it would be interesting to choose different geometric shapes to delimit the geographical area where the user is expected to pass when going to the destination and evaluate the influence of this change. Other fruitful future works are considering vehicles and users heterogeneity in the scheduling algorithm, for instance in terms of speed or download urgency.

Acknowledgments

The authors would like to thank FAPERJ, CNPq, and CAPES Brazilian research agencies for partially funding this work.

References

[1] Cisco, . Cisco visual networking index: Global mobile data traffic forecast update 2014-2019 white paper. Available at http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html; 2015.

[2] Cheng, N., Lu, N., Zhang, N., Shen, X.S., Mark, J.W.. Vehicular WiFi offloading: Challenges and solutions. Elsevier Vehicular Communications 2014;1(1):13–21.

[3] Choi, J., Han, J., Cho, E., Kwon, T., Choi, Y.. Survey on content-oriented networking for efficient content delivery. IEEE Communications Magazine 2011;49(3):121–127.

[4] Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.. Networking named content. In: ACM CoNEXT. 2009, p. 1–12.

[5] Vasilakos, A.V., Li, Z., Simon, G., You, W.. Information centric network: Research challenges and opportunities. Journal of Network and Computer Applications 2015;52(1):1–10.

[6] Wang, L., Afanasyev, A., Kuntz, R., Vuyyuru, R., Wakikawa, R., Zhang, L.. Rapid traffic information dissemination using named data. In: ACM NoM. 2012, p. 7–12.

[7] Amadeo, M., Campolo, C., Molinaro, A.. Multi-source data retrieval in iot via named data networking. In: Proceedings of the 1st International Conference on Information-centric Networking. 2014, p. 67–76.

[8] Deshpande, P., Kashyap, A., Sung, C., Das, S.R.. Predictive methods for improved vehicular wifi access. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services. MobiSys'09; 2009, p. 263–276.

[9] Uppoor, S., Fiore, M.. Large-scale urban vehicular mobility for networking research. In: Vehicular Networking Conference (VNC), 2011 IEEE. 2011, p. 62–69.

[10] Amadeo, M., Campolo, C., Molinaro, A.. Enhancing content-centric networking for vehicular environments. Computer Networks 2013;57(16):3222 – 3234.

[11] Amadeo, M., Campolo, C., Molinaro, A.. Content-centric networking: Is that a solution for upcoming vehicular networks? In: ACM VANET'12. 2012, p. 99–102.

[12] Chen, M., Mau, D.O., Zhang, Y., Taleb, T., Leung, V.C.. Vendnet: VEhicular named data NETwork. Vehicular Communications 2014;1(4):208 – 213.

[13] Gerla, M., Wu, C., Pau, G., Zhu, X.. Content distribution in VANETs. Vehicular Communications 2014;1(1):3 – 12.

[14] Razzaq, A., Mehaoua, A.. Video transport over vanets: Multi-stream coding with multi-path and network coding. In: Local Computer Networks (LCN), 2010 IEEE 35th Conference on. 2010, p. 32–39.

[15] Wang, L., Wakikawa, R., Kuntz, R., Vuyyuru, R., Zhang, L.. Data naming in vehicle-to-vehicle communications. In: IEEE NOMEN. 2012, p. 328–333.

[16] Zhang, D., Yeo, C.K.. Enabling efficient wifi-based vehicular content distribution. Parallel and Distributed Systems, IEEE Transactions on 2013;24(3):479–492.

[17] Amadeo, M., Campolo, C., Molinaro, A.. Crown: Content-centric networking in vehicular ad hoc networks. IEEE Communications Letters 2012;16(9):1380–1383.

[18] Malandrino, F., Casetti, C., Chiasserini, C.F., Fiore, M.. Content download in vehicular networks in presence of noisy mobility prediction. Mobile Computing, IEEE Transactions on 2014;13(5):1007–1021.

[19] Matos Mascarenhas, D., Monteiro Moraes, I.. Limiting the interest-packet forwarding in information-centric wireless mesh networks. In: Wireless Days (WD), 2014 IFIP. 2014, p. 1–6.

[20] da Silva, V.B., da Silva, F.O., Campista, M.E.M., Costa, L.H.M.. A trajectory-based approach to improve delivery in drive-thru internet scenarios. Workshop on Emerging Vehicular Networks: V2V/V2I and Railroad Communications - IEEE International Conference on Communications 2013;:499–504.

[21] Chlebus, E., Brazier, J.. Nonstationary poisson modeling of web browsing session arrivals. Inf Process Lett 2007;102(5):187–190.

[22] Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.. Web caching and Zipf-like distributions: Evidence and implications. In: IEEE INFOCOM; vol. 1. 1999, p. 126–134.

[23] Rezende, C., Boukerche, A., Ramos, H., Loureiro, A.. A reactive and scalable unicast solution for video streaming over vanets. Computers, IEEE Transactions on 2015;64(3):614–626.

[24] Afanasyev, A., Moiseenko, I., Zhang, L., et al. ndnsim: Ndn simulator for ns-3. University of California, Los Angeles, Tech Rep 2012;.