Article

A Novel Cloud Approach for Connected Vehicles

Geoffrey Wilhelm [1], Marwane Ayaida [2] and Hacène Fouchal [1,*]

1 Université de Reims Champagne-Ardenne, 51097 Reims, France; geoffrey.wilhelm@univ-reims.fr
2 Université Polytechnique Hauts-de-France, 59300 Valenciennes, France; marwane.ayaida@uphf.fr
* Correspondence: hacene.fouchal@univ-reims.fr

Abstract: Cooperative intelligent transport systems (C-ITSs) are being deployed all around the world. Shortly, in addition to vehicles, bicycles, pedestrians, buses, and all moving equipment will be compatible with C-ITS. These systems are connected through wireless local area networks based on WIFI IEEE 802.11p. The large number of C-ITSs and services will lead to a glut in the bandwidth of wireless networks. To overcome this limitation, we propose in this paper a new approach using the information-centric networking (ICN) paradigm which allows vehicles to communicate with the cloud environment. This scheme is denoted by vehicular central data networking (GeoVCDN). Our approach aims to reduce bandwidth consumption and improve data freshness by taking benefit from the existing application beacons and the geographical routing used by C-ITS actors. We have compared the performances (in terms of the network overhead and data freshness) of our solution to two other well-known ICN-based solutions. Each of them represents one of ICN categories, in particular, rendez-vous network (RENE) and named data networking (NDN). To do so, we have proposed a probabilistic model that allows us to evaluate the freshness and the load of the network. Furthermore, we have implemented these methods in a simulator. Our proposal outperforms the other methods in terms of network overhead and data freshness.

Keywords: C-ITS; connected vehicles; vehicle cloud; ICN

1. Introduction

The deployment of cooperative intelligent transport systems (C-ITSs) started some years ago. They are critical for road safety. Therefore, they need to maintain high quality of service such as low latency, low packet loss, and low bandwidth consumption. For this purpose, one of the proposed strategies consists of submitting request packets in the network in order to collect named data; this mechanism is denoted as information-centric networking (ICN) [1]. When the latter is combined with the cloud approach, it provides the vehicular cloud networking (VCN) [2] paradigm. This paradigm helps to handle mobility thanks to different features (in-network caching, no session, etc.). However, we noticed that in a secure cloud environment, an overhead is generated in the network due to the security mechanisms. We recall that the bandwidth is 5.6 Mb/s for the IEEE 802.11 p MAC layer considered here.

In this paper, we propose an approach, called vehicular central data networking (GeoVCDN), which takes advantage of the VCN and improves the network's performances in terms of network throughput and data freshness, compared to other well-known ICN solutions. To do so, our approach takes benefit from the application beacons sent by every C-ITS (vehicles, roadside units (RSU), pedestrians [3], etc.) between 1 and 10 times per second. In fact, instead of sending new specific packets, the ICN exchanges are inserted in the application beacons. This proposal allows us to reduce bandwidth consumption. To improve data freshness, we use multi-hop forwarding, which allows us to spread the data more efficiently over the network.

In order to prove the efficiency of our approach, we have compared it with two other well-known solutions. The first one is the rendez-vous network (RENE), which is push-

oriented. This means that a client will subscribe for named data and the data will be pushed as soon as it is available. The second one, denoted as named data networking (NDN), is pull-oriented. This means that a client will submit a request for named data and will receive the data as a response if it is available. These comparisons have been carried out through analytic evaluations based on a probabilistic model that we have proposed as well as extensive simulations. The latter has allowed us to confirm the analytic results. Results have shown that GeoVCDN improves some performance indicators, such as data freshness, while reducing the network overhead. Finally, we proposed a comparison between the different approaches regarding a use case: shortest pathfinding in a smart connected city.

Our approach exploits standardized cooperative awareness messages (CAMs). Additionally, it merges the different sensor data into one set of metadata and includes a sequence number that is incremented whenever the data are updated. It implements a multi-hop feature to help extend the zone where the data are spread. All of these allow GeoVCDN to spread frequently updated data through the network without overloading the bandwidth. Moreover, we modeled our smart city as Manhattan topology and proposed an analytic model to estimate the network overhead and data freshness considering the mobility of vehicles. We compared all these approaches using a custom simulator.

The remainder of this paper is organized as follows: Section 2 describes the related works on ICN, C-ITSs, vehicular cloud computing (VCC), and C-ITS standards. Section 3 details our approach's mechanism and architecture. Section 4 defines the probabilistic model that we proposed and gives the results for each approach in terms of the network overhead and data freshness. Section 5 describes our simulator, the results that we obtained for each approach, and the use case study. Finally, Section 6 concludes the paper.

2. Related Works

Vehicular networks are at the intersection of multiple research fields. A specific communication stack has been standardized by the ETSI and is detailed in Section 2.1. Many solutions for vehicular communications are based on a cloud approach. They are part of vehicular cloud computing (VCC) and are proposed in Section 2.2. These networks provide high numbers of messages over the cloud. Indeed, information-centric networking will allow to efficiently exchange these messages within a mobile cloud environment from numerous connected devices. This part is detailed in Section 2.3.

2.1. ETSI Standards for C-ITS

ETSI defined different layers used by the intelligent transport systems (ITSs) to communicate [4] (Figure 1). In the figure, the access is composed of the physical layer [5–7] and the MAC layer [8,9]. The networking and transport layer is ensured by the geonetworking (GN) protocol for the routing and basic transport protocol (BTP) for the transport. Then, the facilities layer specifies the messages exchanged on the channel: CAM (cooperative awareness message) [10], DENM (decentralized environmental notification message) [11], SPAT (signal phase and time), and MAP (map data) [12]. Finally, we find the applications layer (e.g., Android, cloud). As a vertical layer, there are the security [13] and the management layers. These are the standards currently used for the deployment of C-ITSs in Europe and our approach will be based on these ones.

Figure 1, extracted from [9], presents the ETSI layers architecture explained above. The bubbles MA, FA, SA, etc., refer to the protocols and access points to associated services that we will not detail here. The G5 technology is based on the standard IEEE 802.11p, a kind of Wi-Fi (IEEE 802.11) dedicated to short-range communications, considering high-speed mobility. In the current study, we will focus on three types of communications: V2V (vehicle to vehicle), V2I (vehicle to infrastructure), and I2V (infrastructure to vehicle). ITS-G5 is another name for the access layer composed of the physical layer 802.11p and the MAC layer. The geonetworking protocol, associated with BTP, is used to route and transport the data. The geonetworking protocol is a geographical routing protocol, which means a packet is not destined for a specific node but for a region, usually called a destination area. There

are three kinds of routing algorithms: the simple one, contention base forwarding (CBF), and the advanced one. The three algorithms use the same process to forward the packet to the destination area: they all select the closest neighbor to the area (greedy process). When the packet is received in the destination area, it is broadcast again to everyone. In the current study, we will consider simple geonetworking, since it is the one currently used for ITS.
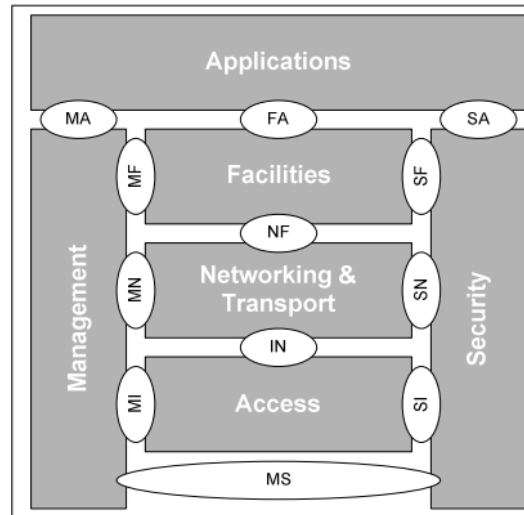

Figure 1. ITS station architecture.

Security data, used to verify the integrity and authenticity of the message, are included in geonetworking headers. Certificates are provided by a public key infrastructure (PKI) and a turnover in a pool of certificates guarantees the privacy of the user. In our study, we focus on CAM (cooperative awareness message) and DENM (decentralized event notification message), which are the most common messages. CAMs are a kind of super beacon emitted by the vehicle and give its speed, location, heading, path history, etc. They are emitted in a single hop (i.e., only to the neighbors). These messages are sent at a frequency varying between 1 Hz and 10 Hz (10 Hz most of the time).

Its structure is defined as being extensible, allowing it to include proprietary content without changing the nature of the message. Thus, we will take advantage of the CAM's structure in order to include additional information, such as interests and data.

2.2. Vehicular Cloud Computing

The smart city is one of the main scopes of IoT. As is reminded by the authors of [14] and [15], the world population is more and more concentrated in urban areas, increasing at the same time the number of connected objects. As it is explained in [16], the IoT is booming: over four years, the amount of stored data was seven times higher. In the meantime, the amount of data sent through the network has been multiplied by three. Thus, it is necessary to design adapted network architectures to handle this amount of messages [17]. The authors of [18–20] agree on the fact that cloud computing is an efficient solution. In addition, there is a need for a platform to manage all the connected systems, which could be data providers, available resources for computing, etc. According to [14], the trend would be to implement a middleware acting as a link between a global platform and a kind of sensor network.

The paradigm of sensor cloud results in two services: sensing as a service (SnaaS) and sensor event as a service (SEaaS) [18,21]. SnaaS is a pull-oriented service whereas SEaaS is push-oriented. Using one or another will have an impact on the network overhead: this impact will be detailed later. Sensor clouds' objectives are the computing of complex data (coming from the sensors' data), ensuring scalability, and guaranteeing real-time data processing. The authors of [18] assume that every connected object will have an IP address and they do not consider the other kind of architectures such as information-

centric networking (ICN). An important difference between the two services and which is not mentioned in the paper is that SnaaS is pull-oriented and SEaaS is push-oriented, it is important because using one or the other of the strategy will have an impact on the network overhead. The authors of [14] explain that the cloud of things (CoT) combines several models: software as a service (SaaS) for the user application, platform as a service (PaaS) to allow publishers to handle their sensor networks, and infrastructure as a service (IaaS) to manage the data produced by the sensors. Additionally, they slice IoT into five layers: device, network, middleware, application, and business. The challenge that security represents in the sensor network is often mentioned. They consider that ICN architecture brings concrete solutions for security and interoperability as the middleware layer.

While the number of connected objects is growing, their computing and storage capacity is growing as well. This allows local handling of a part of the IaaS and reduces the amount of messages of data dedicated to being sent to remote clouds [22]. This principle is handled by vehicular cloud computing, which is a category of mobile cloud computing. In [23], the authors handle two issues: the orientation of the network toward the data and edge computing [24]. ICN totally covers the first issue and partially covers the second one by keeping the data at the closest location to their relevant area. Vehicular cloud computing can take many forms: cloud computing for the vehicles (e.g., the cloud of infrastructure, distant cloud) [25] and cloud computing by the vehicles (e.g., dynamic cloud, context cloud) [26]. In [20], the authors present how mobile cloud computing differs from cloud computing. The latter aims to provide resources on demand with minimal management and low latency, whereas the first one focuses on mobile devices, and one uses remote clouds to achieve some tasks. According to this paper, vehicular clouds can either be mobile clouds with the previously given definition or dynamic clouds composed of vehicles providing the same services found on the Internet. In [20], three ways to form a cloud have been presented:

- static way (i.e., parking);
- with infrastructure as a leader;
- dynamic way (on the demand of moving vehicles).

According to the needs of architectures that handle the sensors and their data, some parameter values have been proposed. In [15], the authors present a framework in order to easily access ITS components as cloud nodes. It is composed of three layers. The end user refers to either a user or a server/data center. A node submits requests and receives responses through the communication layer. The communication layer aims to select the relevant link to be used between two actors depending on the component features. Additionally, it provides access to the next layer, the cloud. The cloud layer is either a classic data center or a dynamic cloud composed of ITS actors. Even if this method seems to be easy to implement, it will induce network overhead since it generates a high amount of data [27].

In [28], the authors propose to combine distant cloud and vehicular cloud into their architecture. In this architecture, the nature of the cloud type (static, dynamic, or infrastructure) has no impact on the solution. Indeed, it is based on the virtualization of the components in order to handle heterogeneous resources.

The paradigms of context and ecosystem for the vehicles are presented in [29]. The context is a logical vision of one or several physical entities (e.g., a mall with its parking). The ecosystem which is defined as a set of contexts bound by a pattern (e.g., mobility pattern, habit pattern). The ecosystem aims to bring data closer to the user of another context. It also allows the slicing of geographic areas in hierarchical ecosystems (e.g., country and its regions, region and its cities). Geographical relevance is an important criterion for vehicles. Additionally, this hierarchical vision is aligned with the naming method adopted for information-centric networking.

2.3. Information-Centric Networking

In 1999, a new paradigm appeared, information-centric networking, which rethinks the classic IP network architecture, host oriented, to focus on the data itself (TRIAD project [30]). This new paradigm is adapted to the Internet of Things and mobility, two features of vehicular networks. Some articles already introduce the ICN concept for the vehicular network, such as [31,32], but none of them take into account neither the nature of the link of communication between the vehicles or between the road infrastructures and the vehicles, neither the ETSI standards currently used for the deployment of C-ITSs in Europe. Furthermore, there is consensus on the method to use for the security aspects. The first principle of ICN is the naming of the data (named data objects—NDO). The naming is the foundation of the ICN exchanges, it ensures the identification of data (like the way of identifying a server with its IP), the routing of interest, and possibly the verification of the integrity of the data [33]. Each approach differs in how the naming is used and the naming method. Still, two methods can be highlighted: hierarchical (such as for URL) and flat (not readable for the human, it can carry the data needed to ensure the authenticity and integrity of the message). Exchanges are done according to a request/reply model, which is denoted by interest/data in ICN. The client/consumer submits the name of the data to the network and the network replies with the data when they have been found. The are two phases of routing, the one used to route the interest and the one used to return the data to the client. The routing of the interests can be done in different ways: every node of the network broadcasts the interest until the data are found or by using a name resolution system (such as DNS) or by making a smart routing based on the data name. The data reply is sent by using the path of the interest or can use a better road if it is possible.

Another principle is in-network caching, which is done during the process of the data return. Devices that are able to do it, store the data, and this allows to provide the data faster for the next interests, instead of routing the interest again. Then, In-network caching allows to reduce the number of exchanges on the whole network [34,35] and to better handle the mobility of the nodes. The naming permits storing data on several devices without considering each copy as unique data. Furthermore, for the security aspects, it is not necessary to trust the server anymore because the data are secured itself. Finally, Ref. [33] presents and compares four main approaches. The first one is DONA (data-oriented network architecture), which consists of routing an interest based on the name and replying on the same or a different route. The second one is CCN (content-centric networking), which routes the interest in a hierarchical way and sends back the data on the same path. The third one is PSIRP (publish-subscribe internet routing paradigm), based on a "rendez-vous" (RENE) principle [36]: when a node publishes data, it checks the "rendez-vous" server to know which node is subscribed and provides the data to each client using the routes given by the server. Then, the last one is NetInf [37] (network of information), based on a name resolution system. All these principles are generic; there exist many ways to implement them as it is shown in these projects. Furthermore, we think that the nature of the communication link and the exchanged data nature cannot be ignored to design an efficient ICN method [38].

Further explanation of these principles is given in [39], and the authors bring additional information about the fundamental aspects of the ICN architecture. One of these aspects concerns resilience to mobility. Current networks have been designed to satisfy communication between two static points. The appearance of wireless and mobile networks leads to a change in the nature of the connections. These ones become fleeting, even opportunistic. Furthermore, thanks to in-network caching and the absence of a session, ICN is designed to handle mobility. Since the Internet has not been natively designed to counter malicious users, it is needed to design some mechanisms to handle them. Often, the security mechanisms are deployed on the server and do not penetrate deeply into the network. So, there continuously are malicious messages in the network. By applying the security part to the data itself, ICN brings an efficient solution to this issue: every node is able to identify a malicious packet and not treat it at the entry of the network instead of

the end. Then, the authors present a list of unsolved research axes. Among them, we find naming. The naming method is crucial to take care of the security and the overhead. There is also a need to improve the performance of the routing and caching methods. Finally, it is important to determine how to best exploit each kind of physical communication link. This study provides an update on the ICN architecture in order to replace or complete the existing mechanisms currently used on the Internet. However, the other types of networks in which ICN would natively fit (e.g., sensor networks) are not mentioned in this paper. In terms of open issues, the authors talked about the possibility of using ICN on different layers. This idea is also mentioned in [40] and that also enters in our proposal. The authors of [41,42] discuss the intake of ICN for the Internet of Things. In these studies, the ICN is not seen as an alternative method to the existing one, but as a supplement for the emerging domain of the connected objects. Indeed, connected objects generate data in a context where the object itself has no importance. The ICN would, then, be an architecture for connected objects and these would access the existing structures by using the classical host-oriented network. The ICN allows for improving the delivery delay of data and the network overhead. Among other reasons, we could cite scalability, quality of service, energy efficiency, and heterogeneity [41]. All these items are very important for C-ITSs.

In ICN, the routing methods are classified into three categories: proactive routing, reactive routing, and opportunistic routing [42]. Proactive routing consists, for a node, of transmitting regularly its route; thus the network's nodes can keep their routing table updated. Reactive routing consists of discovering the network when it is needed. Opportunistic routing consists of using received data to update the routing table (e.g., a beacon received by a vehicle). Concerning naming, according to [40], hierarchical naming makes consensus, but the method remains to be determined. Finally, they discuss the way to keep the network updated (at the data level): either by flooding the network with interests (NDN), or by the initiative of the producer of the data (RENE). They think that the first method is better fitted for VANETs but it needs additional mechanisms to not overload the network, a problem that we address with our approach. Thus, Ref. [43] proposes a method to minimize the number of sent interests. The solution is based on the election of a common neighbor to forward the interests. The method looks efficient for the proposed performance indicators, yet we think it creates a new problem: the electing process is done on the G5 channel and is going to create a new overhead. The authors of [44] also propose a method, called enhanced vehicular named data networking (EVNDN), based on the election of a neighbor and compares it with rapid named data networking (R-NDN) [45]. R-NDN consists of a client emitting an interest and selecting the farthest node among its neighbors to forward its interest. The authors of [44] raise the problem of signal quality and then propose to select a less far node. However, we think that methods based on neighbor elections are risky because one can not know in which direction of our neighborhood the data are closer. Additionally, the communication link is not necessarily bidirectional between two nodes (i.e., node X can be part of the neighborhood of node Y while X is not able to sense Y because they have different transmission ranges). The research on naming, routing, and security are highlighted as the first steps and the physical communication link (based on the broadcast) as the future research area [42,46]. Our approach treats all four of these topics.

ICN has been seen by some researchers as a substitute for IP architecture. For others, it is a complement to specific domains such as IOT. Concerning the internet of vehicles, none of the existing propositions take into account the current choices made in the deployment of the C-ITS in Europe or the nature of the communication, which is done over Wi-Fi with geographical routing. In [47], the authors conclude that it is necessary to bring modifications to the C-ITS if we want them to use ICN. However, with our approach, we succeeded in including ICN without modifying ITS standards.

3. GeoVCDN Approach

Our approach is denoted GeoVCDN for geographical vehicular central data networking. Geographical for the geographical routing aspect, and central data networking because of the processing applied to the data and the way to deliver them, as it is explained in this section.

3.1. Centralized Context Cloud Architecture

In the field of intelligent transport systems (ITSs), the data's relevance is determined by its location and is limited in time. Considering this, the context cloud paradigm gives the possibility to filter the data based on their locations (district, town, harbor, ski resort, etc.). Furthermore, the context cloud is hierarchical (e.g., a building belongs to a district that belongs to a town). According to the literature, the most efficient approach for naming in information-centric networking is the hierarchical way. In our architecture, the cloud is composed of a static part (RSUs and C-ITS central) and a dynamic part (the vehicles). We consider a vehicle as being a part of the cloud when it is linked, directly or indirectly, to an RSU over the G5 channel. Note that data management within the cloud is out of the scope of this study.

Figure 2 describes the message exchanges between each actor of the architecture.

- Step 1: Sensors send their data to the C-ITS central
- Step 2: The C-ITS central generates the metadata
- Step 3: The metadata are distributed over RSUs
- Step 4/6: Vehicles broadcast their interests over the CAMs
- Step 5: RSUs broadcast the metadata after receiving interest from vehicles (Step 4)
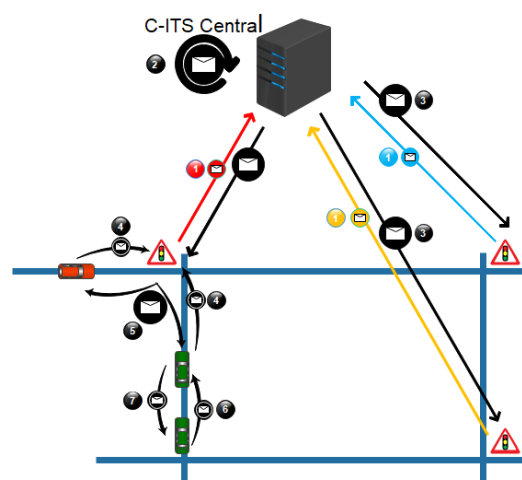- Step 7: Vehicles broadcast the metadata after receiving interest from vehicles (Step 6)

Figure 2. GeoVCDN exchanges.

3.1.1. Static Part

The static part of the context cloud is composed of the roadside units (RSU) and a central station named C-ITS-C (cooperative intelligent transport system central). RSUs are the gateways between the two parts of the cloud. They communicate using ITS-G5 with the mobile node and are wired to the C-ITS-C, which is responsible for the computing and distribution of the data in the static part. We consider that in the connectivity between the RSUs and the C-ITS-C, the computing resources do not suffer from any constraints: the central station can be distributed regarding the need and the size of a context. This is a topic related to edge computing that we do not treat in this paper. This would also address the different issues related to centralization (bottleneck, delay, failures, etc.).

3.1.2. Dynamic Part

The mobile part of the cloud is composed of vehicles. A vehicle is considered as being a part of the cloud if it has access, via the G5 channel, to the static part. This communication channel and the ephemeral nature of the connection raise issues. First, because the bandwidth is limited (750 kB/s) and is not duplicable (wireless network) and because a vehicle is not permanently connected to the cloud. Additionally, for data to be useful by a vehicle, it is important that they are received rapidly.

3.1.3. Centralization

Data centralization is a key process in an environment with a lot of sensors. It allows to aggregate of the data if relevant, minimizes the exchanges needed for each node of the network to be updated, and optimizes the synchronization of every node of the static part. So, the C-ITS-C eases the creation of metadata (which gathers a set of smaller data), the application of a sequence number, and the computing of a hop limit, three mechanisms that we propose in our approach that reduce the use of the bandwidth. In addition, for future work, the central node has an omniscient view and can learn and adapt its naming strategy or form a new context cloud based on the need of each service. To determine if a mechanism (metadata or sequence number) should be used for a specific service, in a specific context, we present two expressions based on the parameters presented in Table **??**.

Table 1. Mechanisms' parameters.

Designation	Description	Value
C	Number of sensors concerned	
P_i	Data's id length (bytes)	1
P_v	Data's sequence number length (bytes)	1
P_d	Data's length (bytes)	4
F_{CAM}	CAM frequency (Hz)	10
F_d	Data's update frequency (Hz)	1
N_v	Number of vehicle in range of an RSU	

3.1.4. Metadata

With our approach, we propose to merge a set of sensors' data as one set of metadata. Every time a sensor's record changes, the information is sent to the C-ITS-C. A new metadata set is produced with the same id and a new sequence number. This new metadata set is then distributed to the routers (RSUs) of the static part in the context cloud. Grouping several data as one presents an asset: it benefits from a common name and a sequence number. For example, in a city context, if we want to get information on one hundred traffic lights, only one ICN interest is needed instead of one hundred. Oppositely, the data returned contain information about every traffic light, even the ones that did not change. Thus, we have to be sure that the metadata process will not be harmful to the overhead and this depends on the context, the number of sensors, and the data update frequency. This following inequation $C * F_{CAM} * (Pi + Pv) * Nv > C * (Pi + Pv + Pd) * F_d$ can be used to determine if the metadata will be useful. The inequation's left side represents the weight of the interests generated every second by all the vehicles in the range of an RSU in the case that we do not build metadata. The right side gives the weight of the sent metadata on the G5 channel by one RSU at the frequency Fd.

For example, let us take $Nv = 100$, and $C = 100$, then:

$$\begin{cases} C * F_{CAM} * (Pi + Pv) * Nv = 200000 \text{ bytes} \\ C * (Pi + Pv + Pd) * Fd = 600 \text{ bytes} \end{cases}$$

In this system, it is preferred to build the metadata.

If $Pd = 100$, $Fd = 10$ and $Nv = 10$, then:

$$\begin{cases} C * F_{CAM} * (Pi + Pv) * Nv = 20000 \text{ bytes} \\ C * (Pi + Pv + Pd) * Fd = 102000 \text{ bytes} \end{cases}$$

This time, it is better to not use metadata.

3.1.5. Sequence Number

The sequence number allows us to avoid sending non-updated data. It is an additional value that causes an overhead, but in some cases this overhead is compensated. When $Pv * Nv * F_{CAM} < (Pd + Pi) * C * F_{CAM}$, the sequence number mechanism reduces the overhead related to the ICN exchanges on the G5 channel.

Here is another example, with $Pv = 4$, $Nv = 20$, and $C = 10$:

$$\begin{cases} Pv * Nv * F_{CAM} = 800 \text{ bytes} \\ (Pd + Pi) * C * F_{CAM} = 500 \text{ bytes} \end{cases}$$

In this case, the sequence number will not be good for the overhead.

Now, if $C = 50$, then $(Pd + Pi) * C * F_{CAM} = 2500$ bytes, so the sequence number helps to reduce the overhead.

3.1.6. Hop Limit

The hop limit is an essential feature when there are exchanges between vehicles. It permits us to avoid propagating data too far from its relevance area or after they expire. Indeed, a vehicle far from an RSU is not aware of the validity state of the data being propagated. Natively, the CAM is a single-hop message. Its routing header does not have any information about the hop limit. Thus, it is necessary to add it to the ICN payload. Let us note that if the hop limit is 1, we can use a default value instead of adding the information.

3.2. Geographical Routing ICN Procotol

The cloud computing part handles the pre-treatment of the data. To deliver the data to the dynamic part of the cloud, we designed an ICN approach that we describe here.

3.2.1. Packet Structure

As introduced in the related works, CAMs are sent on the G5 channel every 100 ms. These messages are encapsulated with security data. The CAM structure is extendable and so it is possible to insert some extra data. We choose to use this feature to include the ICN interest and the ICN data inside the CAM (Figure 3). Thus, we take advantage of the already existing security data. The overhead related to the ICN exchanges is, then, only dependent on the size and the name of the data. Analysis of the overhead is provided further in this study.

Figure 4 shows the structure of the ICN payload in our approach. Compared to NDN packets [48], we removed the security data (this part is handled by the CAM) and we put new information about the sequence number and the hop limit (in order to bypass the single-hop nature of the CAM).

3.2.2. Packet Naming

In the classical internet architecture, to retrieve data, it is first necessary to reach a host. The naming of the host is understandable for the network. Oppositely, the ICN data name is totally opaque for the network. Every application is, then, free to use its own naming scheme for the data. It is understood that flat name is theoretically usable in the ICN network. However, there is a consensus on using hierarchical names. The hierarchical name helps scalability, which is very important in IOT. Additionally, it perfectly fits our

context cloud architecture, which is also hierarchical. The name of data has only to be unique in a specific context and not in the entire network.
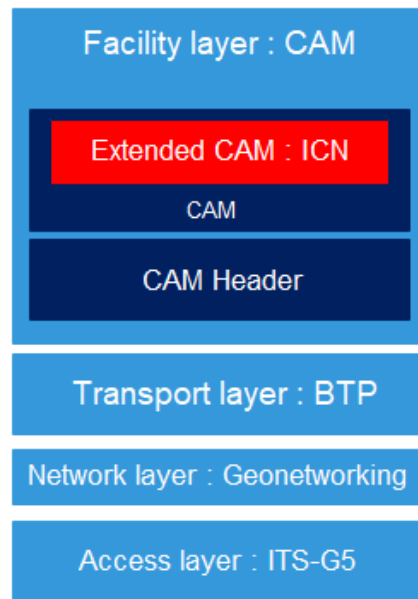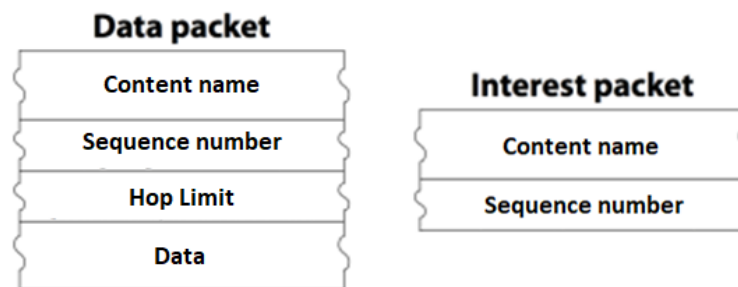

Figure 3. ICN Data in the CAM.

Figure 4. ICN Packets.

3.2.3. Content Store (CS)

The content store contains the data, the sequence number, and the current hop limit associated with the data. Before storing data, a mobile node checks if the number of the hop is higher than 1, else it is the last hop. Additionally, before updating the content store with received data, the mobile node checks if the current sequence number is lower than the received one. If these two conditions are fulfilled, the received data are stored with the sequence number and the hop limit decremented by one (Figure 5). For an RSU, the hop limit has to decrease over time, because the longer the data has been stored, the closer they are to expiration. Additionally, an RSU will always update the data coming from C-ITS-C, because C-ITS-C only broadcast the data when they change. However, if the data are coming from the mobile nodes, they will not update.

3.2.4. Pending Data Table (PDT)

The pending data table references which data of the content store have to be included in the next CAM. When a node receives data with the same sequence number as the one it possesses, then the data are removed from the PDT (Figure 5). When a node receives an interest for data with a sequence number lower than the one it possesses, then it adds the data to the PDT (Figure 6).

3.2.5. Pending Interest Table (PIT)

The pending interest table is fulfilled by applications. When an application needs data, it identifies its name and fills the PIT with the name and the sequence number if a version of the data has already been received. The transmission of interests is done from vehicle to vehicle and from vehicle to RSU. It is done at regular intervals in order to check if there is an update of the associated data (Figures 7 and 8).
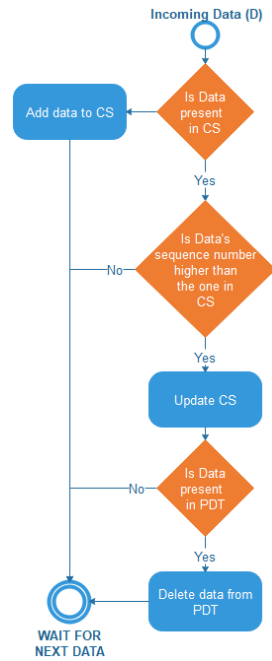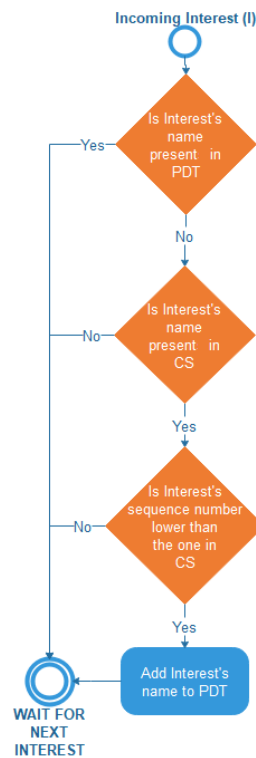
Figure 5. GeoVCDN Incoming Data.
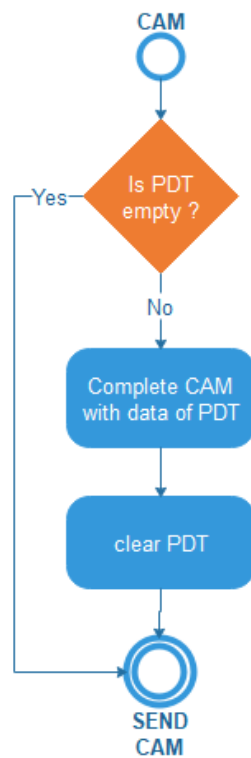
Figure 6. GeoVCDN Incoming Interest.
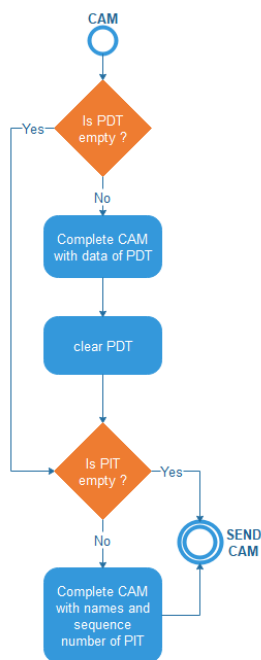
Figure 7. GeoVCDN UBR Sending CAM.

Figure 8. GeoVCDN Vehicle Sending CAM.

3.3. Discussion

To summarize, vehicular cloud computing gives the opportunity to manipulate the data provided by a set of sensors in order to optimize the way they are delivered: the creation of metadata, adding a sequence number, and computing a hop limit. These three mechanisms are going to affect the network overhead and the data freshness. The next part of the paper partially consists of expressing and measuring these effects. Additionally,

we present our ICN approach which consists of taking advantage of the CAMs' security data. Our strategy allows us to reduce the size of our ICN packets. The association of the cloud architecture and of the ICN approach gives a vehicular cloud networking (VCN [2]) method, which flawlessly fits the communications architecture and the European standards currently used in the C-ITS deployment projects in Europe.

4. Analytic Model

In order to evaluate and compare our approach to some well-known existing ones, we use an analytic model. First, we present the framework. In this framework, we model the network, the vehicles' motion, and the dissemination of the message. Then, we use these models to give the analytic expression of the studied approaches. These approaches are compared using two factors: the network overhead generated by the ICN exchanges on the G5 channel and the data freshness, which is the percentage of vehicles in the network, which obtain the data before their expiration. In order to study the impact of each mechanism of our approach (GeoVCDN), we also evaluate two light versions of our proposed approach, namely, vehicular centralized data networking (VCDN) and vehicular centralized data networking over CAM (VCDNoCAM). VCDN works as NDN but instead of having one data per sensor, we have one versioned metadata that gathers the data of the same kind of sensors. With VCDNoCAM, the ICN exchanges are integrated into the CAMs. Additionally, the GeoVCDN extends VCDNoCAM by extending the vehicle's roles to act as a router. So, they can also disseminate the data in their CAMs. In this model, we made two assumptions:

- To ease the representation, without distorting it, the used network topology is Manhattan.
- For our approach (GeoVCDN), we express the worst case for each performance indicator (i.e., for the overhead, we give the upper bound, and for the data freshness the lower bound).
- Vehicles are homogeneously spread in the network.
- Vehicles' speeds are constant.

4.1. Model Description

The framework mainly consists of giving the tool to express the behavior of each approach in the network. At first, we model the network itself: size, number of RSUs, coverage, etc. (Figure 9). Then, we model the settings needed to express the exchanges of the messages in the network. Every parameter is listed and explained in a separate table at the beginning of each part.
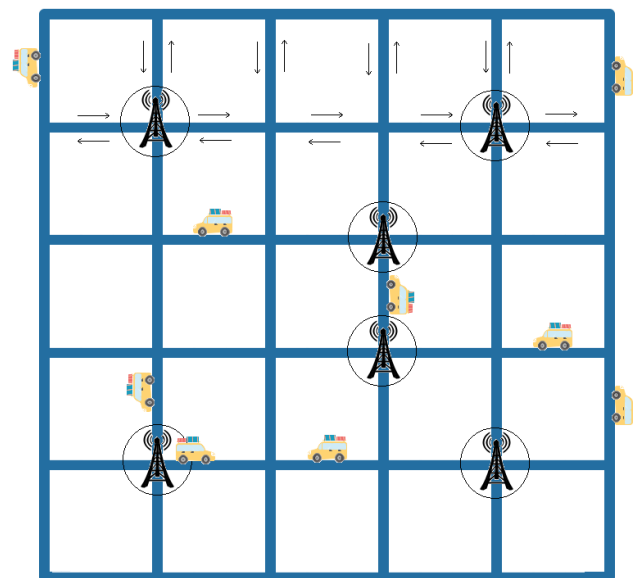

Figure 9. Smart City Scheme.

4.1.1. Network Modeling

The parameters needed to model the network are presented and described in Table 2.

Table 2. Network parameters.

Designation	Description
R_C	Junction's length (m)
L	Lenght of a segment (m)
D	Size of the grid
L_R	Overall road's length of the network (m)
N_S	Number of road segments
ϵ	Number of junctions
β	Number of RSUs
L_{RSU}	Length of road covered by RSUs (m)
φ	Average number of segments per junction
δ_{RSU}	RSU's coverage area (%)
P_{RSU}	RSU's range of transmission (m)

Definition of R_c

R_c is the distance between two bounded junctions divided by two. We could also define it as the radius of a junction.

$$R_c = L/2 \tag{1}$$

Definition of ϵ

In a square Manhattan model, the number of junctions ϵ is given by the squared dimension D, then we have the following expression:

$$\epsilon = D^2 \tag{2}$$

Definition of L_R

First, we note that we have four corners which have two segments (L_1). Furthermore, each edge of the city (there are four) contains as many junctions as the dimension of the grid to which we have to remove the corners, and each one of these junctions has three segments (L_2). Finally, we remove two from the dimension of the grid (which corresponds to the edges) and once it is squared, we obtain the number of junctions with four segments (L_3). The length of road L_R is obtained by multiplying the sum $L_1 + L_2 + L_3$ with the length of a junction R_C (1). We have to distinguish the junctions with two roads (corners), the ones with three roads (edges), and the others with four roads.

$$L_1 = 4 * 2 \tag{3a}$$

$$L_2 = 4(D - 2) * 3 \tag{3b}$$

$$L_3 = (D - 2)^2 * 4 \tag{3c}$$

$$L_R = (L_1 + L_2 + L_3) * R_C \tag{3d}$$

Definition of φ

We compute the percentage of each kind of junction using the following expressions:

- $\varphi_4 = (D - 2)^2 / D^2 * 100$, the percentage of junctions linked to four segments.
- $\varphi_3 = 4(D - 2) / D^2 * 100$, the percentage of junctions linked to three roads.
- $\varphi_2 = 4/D^2 * 100$, the percentage of junctions linked to two roads.

The average number φ of segments linked to a junction is expressed as follows:

$$\varphi = ((4 * 2) + (D - 2) * 4 * 3 + (D - 2)^2 * 4)/\epsilon \tag{4a}$$

$$\varphi = \frac{4D^2 - 4D}{D^2} \tag{4b}$$

$$\varphi = 4 * \frac{D - 1}{D} \tag{4c}$$

Definition of L_{RSU}
The length of the road covered by the RSUs is expressed based on φ (4) and the range of an RSU (P_{RSU}) which we multiply by the number of RSUs.

$$L_{RSU} = \varphi * P_{RSU} * \beta \tag{5}$$

Definition of δ_{RSU}
The coverage area is the percentage of the total road covered by an RSU.

$$\delta_{RSU} = L_{RSU}/L_R * 100 \tag{6}$$

4.1.2. Communication Environment Modeling

The parameters needed to model the communication between the actors of the environment are presented and described in Table 3.

Table 3. Communication parameters.

Designation	Description
P_V	Vehicles' range of communication (m)
V	Vehicles' speed (m/s)
α	Number of vehicles
σ	Data generation frequency (Hz)
ω	Number of sensors
γ	Inter-vehicle distance (m)
ρ	Vehicle density (vehicle/m)
δ_V	Network's percentage in which vehicles are connected to an RSU
λ	Vehicle arrival rate (m/s)
F_V	Frequency of new vehicle arrival in range of one RSU (Hz)
F_λ	Number of new vehicles in the coverage (Hz)
$\theta(A, B)$	Function which gives A if $A < B$ and B otherwise

Definition of γ
Vehicles are homogeneously distributed over the whole network. Thus, we define the inter-vehicle distance as:

$$\gamma = \alpha/L_R \tag{7}$$

Definition of ρ
We have the following expression, which gives the density ρ of vehicles, with the speed $V \in [a; b]$:

$$\rho = \lambda \int_a^b (F(V)/V)dV \tag{8}$$

Definition of λ
Finally, we denote λ the arrival rate of vehicles on one road. Thus, we can compute λ since speed V is constant.

$$\rho = \lambda / V \tag{9a}$$

$$\lambda = \rho * V \tag{9b}$$

Definition δ_V

Usually, the range of a vehicle is lower than the range of an RSU. Thus, even if a vehicle is in range of an RSU, the RSU is not necessarily able to sense it. We expressed the number δ_V as the percentage of the whole network where the RSU are able to receive the vehicle's messages.

$$\delta_V = (\varphi * P_V * \beta) / L_R * 100 \tag{10}$$

Definition F_λ

The number of vehicles entering the coverage area (and also leaving it) every second is obtained by the following expression.

$$F_\lambda = (\lambda * \varphi * \beta) \tag{11}$$

Definition F_V

By diving F_λ (11) by the number of RSUs, we obtain the number of vehicles entering in the range of one RSU every second.

$$F_V = (\lambda * \varphi) \tag{12}$$

Definition θ

θ is a function, taking two parameters and returning parameter 1 if it is lower than parameter 2 and returning parameter 2 otherwise.

1. Based on the following observation :

$$\forall v < 1, [v]/v = 0, \lceil [v]/v \rceil = 0$$

and,

$$\forall v \geq 1, [v] \leq v, 0 < [v]/v \leq 1, \lceil [v]/v \rceil = 1$$

That is to say, if v is lower than 0, its integral part is equal to 0, and thus, no matter what its divisor the result will be 0, and its round-up will be 0. Else, we will obtain 1.

2. Additionally, we generalize this using the following function, which will give A if A<B and B otherwise:

$$\Theta(A, B) = B - (B - A) \lceil [B/A]/(B/A) \rceil \tag{13}$$

4.2. Message Dissemination

We model the message dissemination in the network to consider vehicle-to-vehicle communications. Every second, some vehicles send the data to other vehicles. So, we have to estimate how many vehicles are going to send the data every second. For that, the RSU distribution is important.

Figure 10a,b illustrate the difference between two different distributions of RSUs. Each color represents one step of dissemination (i.e., the list of the new segments where the data are disseminated), for example, Step 1 is marked in red, Step 2 is marked in blue, etc. In the second case (Figure 10b) more exchanges will be needed (seven steps) to cover the whole network than for the first case (Figure10a) for which only four steps are needed. Additionally, the number of segments reached at step two is different: 24 (Figure 10a) versus 6 (Figure 10b).

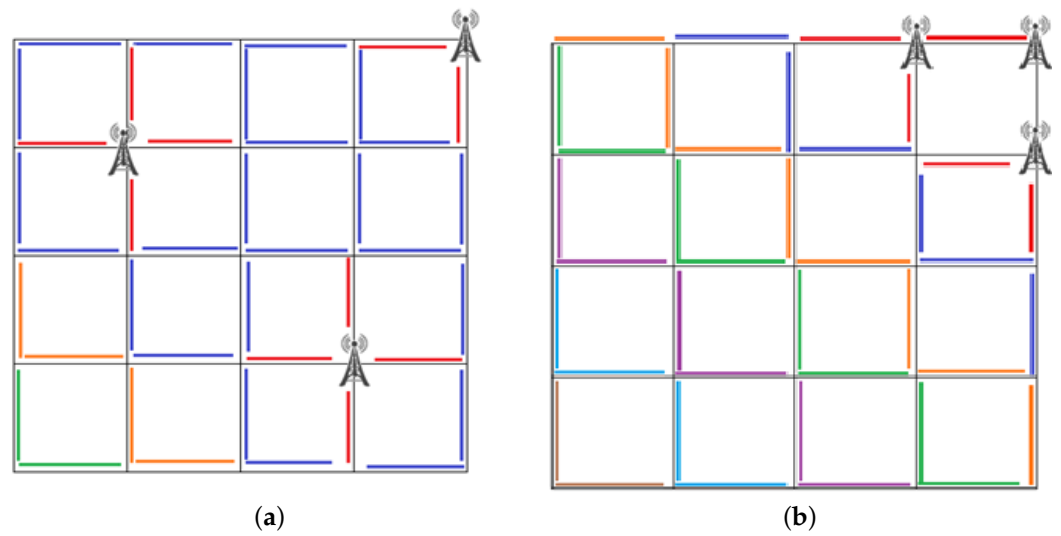Table 4 presents the parameters needed to describe the messages dissemination.

Figure 10. (**a**) Best dissemination scenario; (**b**) Worst dissemination scenario.

Table 4. Dissemination parameters.

Designation	Description
L	Length of a segment (m)
X_C	Indicate the presence (1) or the absence (0) of an RSU on the junction C
S_X	Probability for a segment to have X RSU
M_X	Minimum number of V2V communications needed to cover a segment for configuration X
H_X	Highest number of V2V communications needed to cover a segment for configuration X
HM_X	Average number of V2V communications needed to cover a segment for configuration X
O_X	Number of segment linked to X other segments
N_0	Number of segments with one RSU
N_{RSU}	Number of segments with two RSUs

The first time, we select a segment of the city, which is composed of two junctions at its extremities and the road that links them. Then, we will extrapolate the results to the entire network (city) (Figure 11).
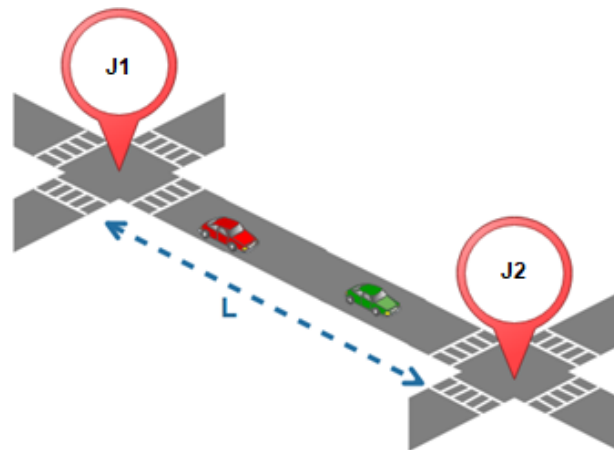

Figure 11. Segment.

Definition of L

Figure 11 schematizes the generic configuration of a segment, with

$$L = 2 * R_C \tag{14}$$

4.2.1. RSU Distribution

To be able to model the dissemination, we first have to determine how the RSUs are placed in the network. Figure 12 presents the different possible configurations for a segment.
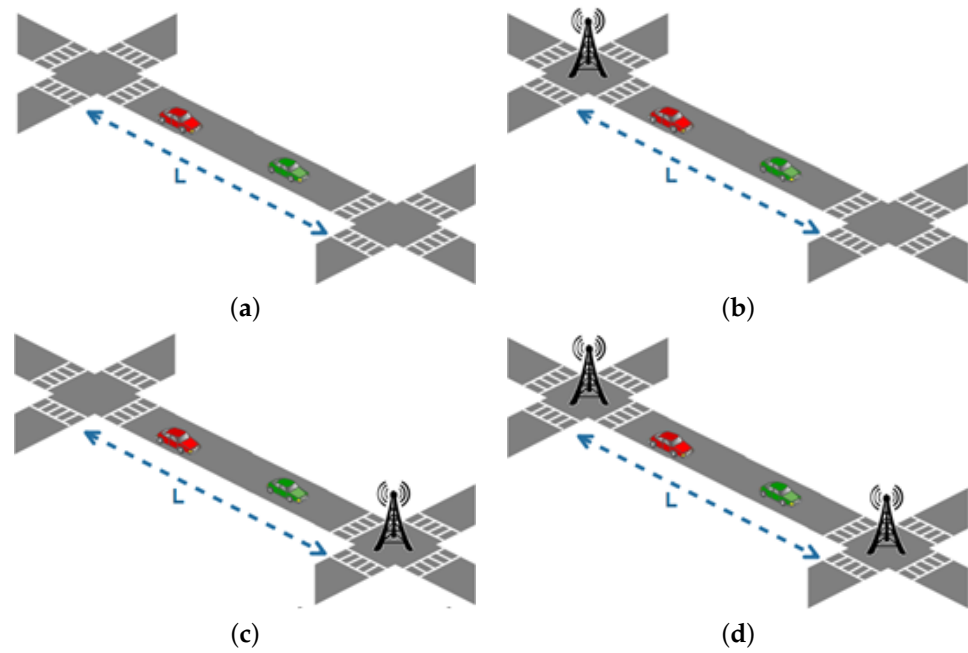

Figure 12. Possible Segment Configurations. (**a**) Configuration 1; (**b**) Configuration 2.1; (**c**) Configuration 2.2; (**d**) Configuration 3.

Figure 12b,c represent segments with only one RSU. Figure 12a represents a segment without any RSU, while Figure 12d represents a segment with two RSUs. Let us note that Figure 12b,c are actually the same.

Definition of X

We denote X_C as the variable indicating the presence or the absence of an RSU at a junction. $X_C = 1$ if there is an RSU and $X_C = 0$ else. Since a segment is composed by two junctions, we will work with $X_C 1$ and $X_C 2$. The probability for X_C to have an RSU is given by $P(X_C = 1)$ and the probability for X_C to not have an RSU is given by $P(X_C = 0)$. The probability of having an RSU on a junction follows the rule shown in Table 5.

Table 5. RSU Probability.

	X_{c1}	X_{c2}	$P(X_{c1})$	$P(X_{c2})$
Config. 1	0	0	$\frac{\epsilon - \beta}{\epsilon}$	$\frac{\epsilon - \beta - 1}{\epsilon - 1}$
Config. 2.1	1	0	$\frac{\beta}{\epsilon}$	$\frac{\epsilon - \beta}{\epsilon - 1}$
Config. 2.2	0	1	$\frac{\epsilon - \beta}{\epsilon}$	$\frac{\beta}{\epsilon - 1}$
Config. 3	1	1	$\frac{\beta}{\epsilon}$	$\frac{\beta - 1}{\epsilon - 1}$

Definition of S_X

S_0 is the probability for a segment to not have any RSU, S_1 is the probability to have one, and S_2 is the probability of having two RSUs. These probabilities are defined as follows:

$$S_0 = (\frac{\epsilon - \beta}{\epsilon})(\frac{\epsilon - 1 - \beta}{\epsilon - 1}) \tag{15a}$$

$$S_1 = 2(\frac{\epsilon - \beta}{\epsilon})(\frac{\beta}{\epsilon - 1}) \tag{15b}$$

$$S_2 = (\frac{\beta}{\epsilon})(\frac{\beta - 1}{\epsilon - 1}) \tag{15c}$$

4.2.2. Vehicles in Transmission Range of an RSU

We denote P_{RX} as the probability for a vehicle to be in the range of an RSU for the configuration X. Furthermore, we call P_{ORX} the probability for a vehicle to be out of range of an RSU for the configuration X. For Configuration 3 (with two RSUs), the probability for a vehicle to be in the range of an RSU is twice P_{RSU}, the range of an RSU, divided by the length of segment L. The probability of not being in the range of an RSU in this same configuration is given by subtracting twice the range of an RSU from the length of a segment, which we divide by this same length as detailed in Table 6:

Table 6. Segment probability.

	P_R	P_{OR}
Config. 1	0	1
Config. 2	$\frac{P_{RSU}}{L}$	$\frac{L - P_{RSU}}{L}$
Config. 3	$\frac{2P_{RSU}}{L}$	$\frac{L - 2P_{RSU}}{L}$

4.2.3. Segments

As shown by Figure 13, a segment may be bound to three (Zone 1), four (Zone 2), five (Zone 3), or six (Zone 4) segments. All possible configurations are presented in Figure 14.
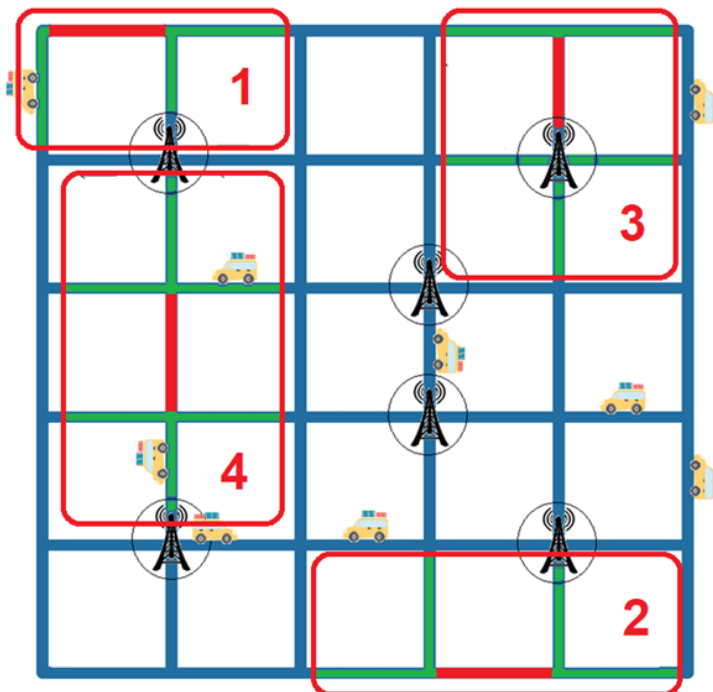

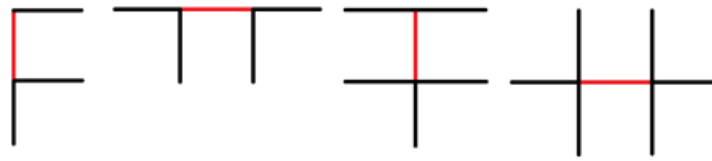**Figure 13.** Segment Configurations example.

Figure 14. Possible Segment Configurations.

Definition of O_X
O_X is the number of segments bound to X segments.
$O_3 = 8$ is the number of segments bound to three other segments.
$O_4 = (D-3) * 4$ is the number of segments bound to four other segments.
$O_5 = (D-2) * 4$ is the number of segments bound to five other segments.
$O_6 = 2 * (D-3) * (D-2)$, is the number of segments bound to six other segments.
Definition of N_0
The number of segments with only one RSU is defined by $N_0 = N_S * S_1$
Definition of N_{RSU}
The number of segments with 2 RSUs is defined by $N_{RSU} = N_S * S_2$

4.2.4. Dissemination Model

For each configuration X, we are looking for H_X, which is the maximum number of hops needed to cover the segment, and M_X which is the minimum number of hops. HM_X describes the average number of needed hops. We obtain for each configuration the values H, M, and HM as detailed in Table 7.

Table 7. Number of Hops.

	H_X	M_X	HM_X
Config. 1	$\lceil \frac{L}{P_V - \gamma} \rceil$	$\lceil \frac{L}{P_V} \rceil$	$(\frac{H_1 - M_1}{2})$
Config. 2	$\lceil \frac{L - P_{RSU}}{P_V - \gamma} \rceil$	$\lceil \frac{L - P_{RSU}}{P_V} \rceil$	$(\frac{H_2 - M_2}{2})$
Config. 3	$\lceil \frac{L - 2P_{RSU}}{2(P_V - \gamma)} \rceil$	$\lceil \frac{L - 2P_{RSU}}{2P_V} \rceil$	$(\frac{H_3 - M_3}{2})$

We want to compute $N_1, N_2, ..., N_n$, with N_1 being the number of segments without any RSU which are bound to at least one segment with one RSU. Additionally, N_2 is the number of segments without an RSU that are bound to at least one segment $\in N_1$, without being bound to a segment $\in N_0$, etc. To get the number of segments at level n, we separate this calculation depending on the number of bound roads, and at the end we sum them. A segment can be bound from $x = 3$ to 6 segments and N_{n_x} is the probability for a segment that is bound to x roads to be also bound to at least one segment of level $n - 1$. We deduce the following expression:

$$N_n = \sum_{x=3}^{6} O_x N_{n_x} = 1 \tag{16}$$

and N_{n_x} is expressed as follow:

$$N_{n_x} = \sum_{y=1}^{x} C_x^y \left(\frac{A * B}{C} \right) \tag{17}$$

With: $A = \left(\prod_{w=0}^{y-1} N_{n-1} - w \right)$,
$B = \left(\prod_{m=0}^{x-y-1} N_S - N_{RSU} - \sum_{k=0}^{n-1} N_k - m \right)$ and
$C = \prod_{z=1}^{x} N_S - N_{RSU} - z - \sum_{k=0}^{n-2} N_k$

The variable x refers to the number of roads bound to a segment, this is given by the global formula (16) and, then, takes a value between 3 and 6, and y refers to the

number of these roads which are from level $n-1$ and takes a value between 1 and x. Then, N_{n_x} (17) is computed by summing the probability of having a segment linked to a road of level $n-1$ with the one of being bound with two roads of level $n-1$ and three roads of level $n-1$, etc., until x roads. The computing results of the probability of having y roads of level $n-1$ has to be multiplied by the number of combinations of y among x, because the road or the road of level $n-1$ may have different locations. For example, if a segment from O_5, which is bound to five roads, is adjacent to one road of level $n-1$, this last one could take five different locations. The formula $\prod_{w=0}^{y-1} N_{n-1} - w$, expresses the probability of being bound to w segment of level $n-1$, to which we have to subtract, at each stage, the segments already processed, that is why w is used. Then, we obtain the following result : $(N_{n-1} - 0) * N_{n-1} - 1) * ... * (N_{n-1} - w)$. We multiply the result of this expression with the probability for the other segments of not being at a lower level than n. This is expressed as follows: $\prod_{m=0}^{x-y-1} N_S - N_{RSU} - \sum_{k=0}^{n-1} N_k - m$. $N_s - N_U BR - \sum_{k=0}^{n-1} N_k$, this allows expressing the number of segments of a level higher than n, which means the total number of segments in the network to which we subtract the segments having two RSUs and the segments of a level lower than n. We also have to subtract the segments of a level higher than n that we already processed in the computing, and they are represented by m. The variable may take a value between 0 and $x - y - 1$, and if we would develop, we would obtain an expression with the following shape: $(N_s - N_{RSU} \sum_{k=0}^{n-1} N_k) * (N_s - N_{RSU} - \sum_{k=0}^{n-1} N_k - m) * ... * (N_s - N_{RSU} - \sum_{k=0}^{n-1} N_k - m)$.

Then, we divide it by the number of segments of a level higher than $n-1$. This number is obtained by subtracting the number of segments with two RSUs and the number of segments of a level lower or equal to $n-2$ $\left(\sum_{k=0}^{n-2} N_k\right)$ from the total number of segments N_s. In the same way, we have to remove the segments already processed at each step of the multiplication, that is why z is used, which has a value between 1 and x. We begin at 1 because we have to subtract the segment that we are processing (which appears in red in Figure 14).

4.3. Approaches

To compare the network overhead (CR) and the data freshness (FR) with each approach, we propose an analytic study, involving the different variables of Tables 2–4. For each approach, we calculate the expression of each variable and we then refer to the definition of CR and FR to give the final expression of the overhead and the data freshness of the associated approach.

Definition of CR

To measure the overhead generated by ICN exchanges, on the G5 channel, we define the network overhead as being the sum of the network overhead caused by the interests CR_i and the network overhead caused by the data CRd.

$$CR = CR_i + CR_d \tag{18}$$

Definition of FR

We define the data freshness FR as being the percentage of the total vehicles which receive the data before its expiration.

$$FR = \frac{\alpha_D}{\alpha} * 100 \tag{19}$$

4.3.1. Network Overhead and Data Freshness Expressions for NDN

A vehicle sends a request for each datum it needs and will repeat it at regular intervals to be sure that it is up to date. Each request and each reply is exchanged as a message which has to be signed and accompanied by its certificate. With this approach, the bandwidth is busy with heavy requests and we retrieve data for every request, even if the data have not been updated. Thus, the number of useless packets is high. Figure 15 shows the different exchanges that take place with the NDN approach in our model.
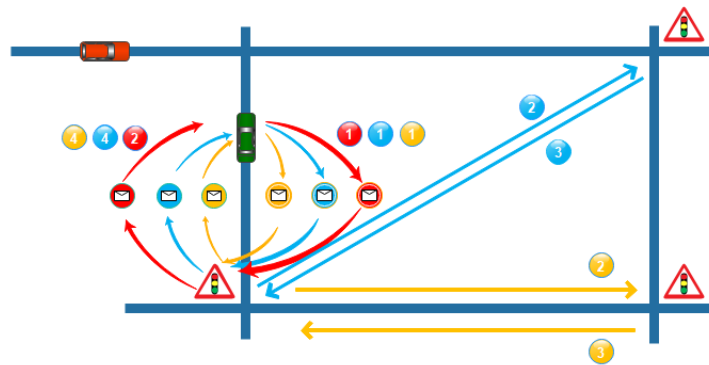
Figure 15. NDN Exchanges.

Firstly, a vehicle sends an interest for each sensor that it needs (1). Then, the traffic light equipped with RSU receives the interests and answers with its data as a sensor and routes the other interests toward the relevant traffic light as a resolution handler (RH) (2). Each requested sensor answers the RH with its own data (3). Then, the RH can transmit the data from the other traffic lights to the vehicle (4). Additionally, these operations are regularly repeated.

Definition of CR

We defined CR as being the sum of CR_i and CR_d (18), so for NDN:

$$CR = PO_i * I_s + I_{RSU} * PO_d \qquad (20)$$

Thus, we get:

$$
\begin{aligned}
CR = (S + P_{if}) * \omega(\alpha/100 * \delta_{RSU}) * F_i + \\
\omega(\alpha/100 * \delta_V) * F_i * (S + P_{df} + P_{if}),
\end{aligned}
\qquad (21)
$$

Definition of α_D

Using NDN, the number of vehicles α_D represents the number of vehicles that are in the transmission range of an RSU, to that we added the vehicles entering the range of an RSU between two data generations.

$$\alpha_D = \frac{\alpha}{100} * \delta_V + \frac{F_\lambda}{\sigma} \qquad (22)$$

Definition of FR

Which we apply to the broad definition of the freshness (19) and give us:

$$FR = \frac{\frac{\alpha}{100} * \delta_V + \frac{F_\lambda}{\sigma}}{\alpha} * 100 \qquad (23)$$

4.3.2. Network Overhead and Data Freshness Expressions for RENE

RENE brings another sight of the ICN for vehicular networks. With this approach, the data are retrieved in "push" mode. Unlike the NDN model, the vehicle does not request for the data but the latter are brought when updated. Without the repetition of the interests, we reduce the use of the G5 bandwidth. Using the information pushing when generated, we also reduce the latency. However, the vehicle has to be in the range of a router when the data are generated, otherwise, it will never receive them. Figure 16 illustrates the exchanges that occur when using the RENE approach in our smart city model.

At first (1), a vehicle subscribes to the server for the sensors that it is interested in. Then, a sensor generates new data (2). This sensor makes a request to the server to retrieve the list of subscribers and how to reach them (3). The server answers with this information (4).

Then, the sensor is able to transmit its data to the ad hoc routers (5), which will forward the data to the relevant vehicles (6).
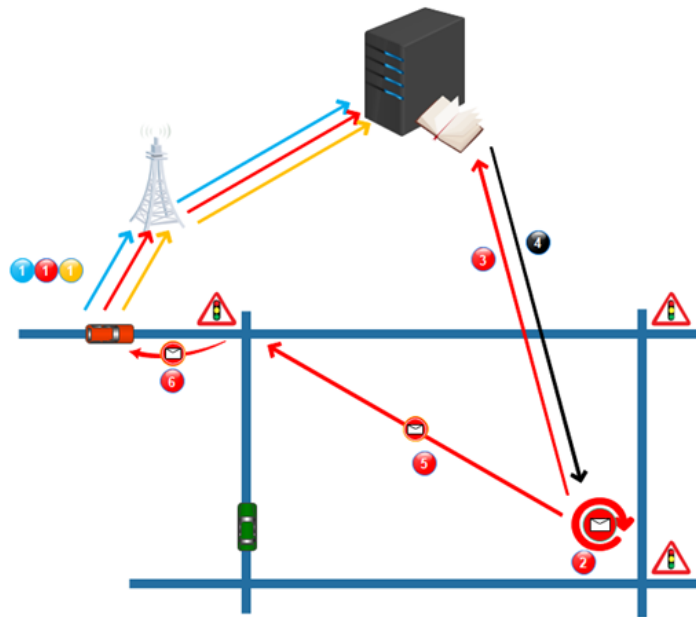

Figure 16. RENE Exchanges.

Definition of CR

According to previous definitions (18), the network overhead dedicated to the ICN exchanges using the RENE approach is given by the following expression:

$$CR = PO_i * I_s + \left(\frac{\alpha}{100} * \delta_{RENE} \right) * \sigma * PO_d \tag{24}$$

Which is equal to:

$$
\begin{aligned}
CR = (S + P_{if}) * \omega * F_\lambda + \\
\left(\frac{\alpha}{100} * (\delta_V + \delta_{RSU})/2 \right) * \\
\sigma * (S + P_{if} + P_{df}),
\end{aligned}
\tag{25}
$$

Definition of α_D

In this approach, α_D is given by the number of vehicles in the area when the data are generated, this gives the following expression:

$$\alpha_D = \frac{\alpha}{100} * \delta_{RENE} \tag{26}$$

Definition of FR

We apply α_D to the general definition of the network freshness (19), we obtain:

$$FR = \frac{\frac{\alpha}{100} * (\delta_V + \delta_{RSU}/2)}{\alpha} * 100 \tag{27}$$

4.3.3. Network Overhead and Data Freshness Expressions for VCDN

The VCDN approach consists of grouping the data of several sensors into one metadata set and applying it as a sequence number.

Definition of CR

In our approach, if we go back to our general expression (18):

$$CR = I_S * PO_i + \left(\frac{\alpha}{100} * \delta_V\right) * PO_d * \sigma + F_\lambda * PO_d \qquad (28)$$

Definition of α_D

In this approach, α_D is obtained by calculating the sum of the vehicles in the covered area when the data are generated and the number of vehicles entering the area before the data expire.

$$\alpha_D = \left(\frac{\alpha}{100} * \delta_V\right) + \frac{F_\lambda}{\sigma} \qquad (29)$$

Definition of FR

We apply the definition of α_D to the general definition of the data freshness (19) to obtain the following expression:

$$FR = \frac{\left(\frac{\alpha}{100} * \delta_V\right) + \frac{F_\lambda}{\sigma}}{\alpha} * 100 \qquad (30)$$

4.3.4. Network Overhead and Data Freshness Expressions for VCDNoCAM

By including the messages in the CAM, we do not need a dedicated security layer for the ICN messages anymore and we do not send a message to a vehicle but we broadcast to all the vehicles in the area.

Definition of CR

Combining CR_d and CR_i, we obtain the overall overhead generated by the VCDNo-CAM approach.

$$\begin{aligned} CR &= \left(\frac{\alpha}{100} * \delta_{RSU}\right) * F_{CAM} * PO_i + \\ &\quad \beta * PO_d * (\theta(F_{CAM}, \sigma) + \theta(F_{CAM}, F_v)), \end{aligned} \qquad (31)$$

Definition of α_D

For this approach, α_D is calculated using the sum of the number of vehicles in the covered area when the data are generated and the number of vehicles entering the area before the expiration of the data. We thus have:

$$\alpha_D = \left(\frac{\alpha}{100} * \delta_{RSU}\right) + \frac{F_\lambda}{\theta(\sigma, F_{CAM})} \qquad (32)$$

Definition of FR

We apply this expression to the general definition of the network freshness (19):

$$FR = \left(\frac{\alpha}{100} * \delta_{RSU}\right) + \frac{F_\lambda}{\theta(\sigma, F_{CAM})} \qquad (33)$$

4.3.5. Network Overhead and Data Freshness Expressions for GeoVCDN

With GeoVCDN, we keep the same mechanism as VCDNoCAM, and we add the V2V exchanges. This new feature requires adding new information. This information is the maximum number of hops that data can make between vehicles.

Definition of CR

Then, we obtain the network overhead which is caused by our approach:

$$CR = CR_i + CR_{d0} + CR_{d1} + CR_{d2}$$

The obtained result is an upper limit of the expected result, because we maximized the number of segments, without an RSU, which are contaminated at each stage.

Definition of α_D

For the freshness, the worst case is the one in which the segments without an RSU do not receive the information. So, we suppose the following hypothesis: the length of a junction is higher than a vehicle's range multiplied by HL, which can be expressed by:

$$L - P_{RSU} > P_V * HL \tag{34}$$

The freshness is given by the number of vehicles being in the coverage area when the data are generated, to which we add, either the number of vehicles entering the coverage area, or the vehicles which receive the information through other vehicles, depending on the relation between the inter-distance and the range of the vehicles. Thus, on the one hand, if the inter-distance is higher than the vehicles' transmission range, α_D is expressed as follows:

$$\alpha_D = \frac{\alpha}{100} * \delta_{RSU} + \frac{F_\lambda}{\theta(\sigma, F_{CAM})} \tag{35}$$

On the other hand, if the inter-distance is lower than the vehicles' transmission range, α_D is obtained by the following expression:

$$\begin{aligned} \alpha_D = \left(\frac{\alpha}{100} * \delta_{RSU}\right) + \frac{\alpha}{N_S} * P_{OR3} * N_{RSU} * \theta\left(\frac{HM_1}{HL}, 1\right) + \\ \frac{\alpha}{N_S} * P_{OR2} * N_0 * \theta\left(\frac{HM_2}{HL}, 1\right) \end{aligned} \tag{36}$$

Definition of FR

Then, if we come back to the general definition of the freshness (19), if the inter-distance is higher than the vehicles' range: $FR = \frac{\frac{\alpha}{100} * \delta_{RSU} + \frac{F_\lambda}{\theta(\sigma, F_{CAM})}}{\alpha} * 100$. Otherwise:

$$\begin{aligned} FR = \frac{\left(\frac{\alpha}{100} * \delta_{RSU}\right) + \frac{\alpha}{N_S} * P_{OR3} * N_{RSU} * \theta\left(\frac{HM_1}{HL}, 1\right)}{\alpha} + \\ \frac{\frac{\alpha}{N_S} * P_{OR2} * N_0 * \theta\left(\frac{HM_2}{HL}, 1\right)}{\alpha} * 100 \end{aligned} \tag{37}$$

4.4. Discussion

In this section, we present a framework in which we can express the behavior of different ICN approaches. This framework takes into account the network topology and the mobility of the vehicles. Additionally, we propose a model for the propagation of the messages between vehicles over the network. Then, we focus on two performance indicators which are the network overhead related to the ICN exchanges and the data freshness. The framework definition allows us to give the analytic model of any ICN approach in order to compare them, as we did with two intermediate versions of our GeoVCDN approach. These versions have been introduced in order to see the benefits of each mechanism introduced in GeoVCDN. In the next section, we compare the different approaches based on the analytic models and the results obtained with a simulator we designed.

5. Evaluation

In this section, we compare the NDN, RENE, VCDN, VCDNoCAM, and GeoVCDN approaches based on the models proposed in the previous section. These comparisons are done on two performance indicators: network overhead and data freshness. These comparisons aim to confirm and evaluate the benefits of our approach, compared to the existing methods, at the network level. Furthermore, to validate the proposed analytic framework, we develop an ad hoc simulator that takes into account some aspects of vehicular communication and the mobility of the vehicles. Finally, we tested NDN, RENE, and GeoVCDN on the shortest pathfinding use case in a smart city. This aims to validate

the benefits of our approach at the application level. Let us remind the reader that our approach is composed of three versions: the first one (VCDN) groups a set of data as one set of metadata, the second one (VCDNoCAM) does the same and also includes ICN exchanges in the CAMs, and the third one (GeoVCDN) extends the second one by adding the V2V communications. In this section, we propose, for each parameter, a table in which every approach is compared using a specific value of these parameters. Expected values are obtained by applying the equations defined in Section 4.3 in the model described in Section 4.1 considering the mobility as described in Section 4.2. Observed values are the ones obtained with our simulator.

5.1. Simulation Results

5.1.1. Simulator Description

In order to validate the proposed models, we developed a simulator from scratch and Figure 17 is an example of a screenshot.
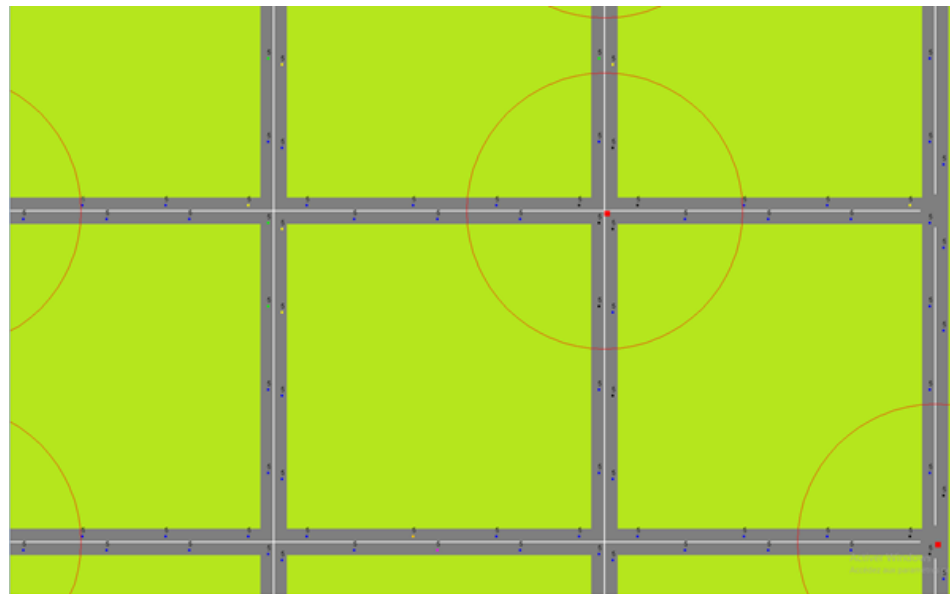

Figure 17. Simulation Extract.

This time-discrete simulator (with a step of 100 ms) takes into account the communication range of the vehicles and the RSUs. The mobility of the vehicles is also handled. Each vehicle emulates the full ETSI stack (geonetworking, security layer, and cooperative awareness message (CAM)) and simulates NDN, RENE, and our proposed approaches. According to the already discussed working hypothesis, it uses a Manhattan city topology and the vehicles are homogeneously spread. At the start, RSUs are randomly placed on the junctions and the vehicles are stationary. After the setting up, the core of the simulator is awakened, making the vehicles move at a constant speed and proceeding to the ICN exchanges between the ITS stations. After a threshold, the RSUs start transmitting at a regular frequency for 10 s. At the end of these 10 s, we save the results on the network overhead and the data freshness. Then, the process is repeated 20 times from the RSU placement step. Finally, we average the results.

Table 8 shows the specifications of the computer that the simulations have been run on. Due to these performances, we were limited to small-scale default values for every variable, which are given in Table 9). However, despite these limitations, we manage to distinguish the curve's trend. Let us note that the values are realistic and based on our observations in European C-ITS pre-deployment projects Scoop, InterCor, and C-Roads.

Table 8. Computer Configuration.

Component	Model
CPU	i7 @3.6 GHZ
RAM	32Go
OS	Windows 10 $\times$64
GPU	Intel HD Graphics 630
IDE	Eclipse neon 3
Programming Language	Java 8

Table 9. Variables and default values.

Designation	Description	Value
R_C	Junction range	300
P_V	Vehicle's range	200
P_{RSU}	RSU's range	250
D	Grid size	4
α	Number of vehicles	267
β	Number of RSUs	12
σ	Data update frequency	1
ω	Number of sensor	12

5.1.2. Network Overhead Simulation Results

We define the network overhead as being the data size that is shared between the different actors, over the G5 channel, on the whole network, and which is associated with the ICN exchanges. For the following figures, we will not show the NDN approach, because its results are on average hundred times higher than the other approaches, which makes it impossible to observe the differences between the other ones (as is shown in Figure 18). Vehicles play a core role in the G5 network overhead. They represent the biggest part in terms of stations, so it is important that the proposed ICN approach scales with a high number of clients. In this section, we will see how the number of vehicles and their communication range impact the network overhead for each approach.
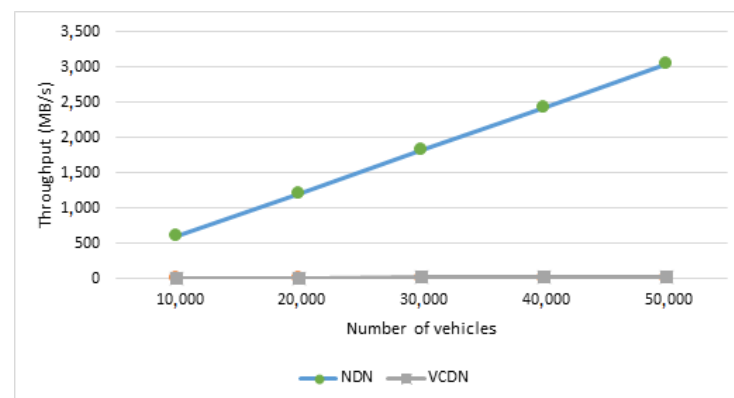
Figure 18. Network Overhead—Number of Vehicles.

First, in Figure 18, we compare NDN and VCDN when we increase the number of vehicles. For NDN, the amount of data and exchanged interests proportionally increases with the number of vehicles in the network. Using VCDN, the number of interests also increases in the same way, but slower because there is only one interest per vehicle. On one hand, for NDN, the data are sent once per vehicle, and every time a vehicle asks, for VCDN the data are only sent when updated. On the other hand, VCDN generates a gain on the security layer. Indeed, for NDN, we have as many security headers as sensors, whereas for VCDN there is only one security header whatever the number of data is. We see that the mechanisms of metadata and sequence number make a significant difference by reducing

the network overhead related to ICN exchanges and making ICN more scalable. For 20,000 vehicles, NDN generates 1.2 GB/s of data, whereas for VCDN it only generates 12 MB/s.

We will compare the behavior of each approach by changing the parameters associated with the vehicles (number and transmission range), the RSUs (number and transmission range), the network (size and coverage area), and the data (frequency, size, and the number of sensors).

In Figure 19, we compare all the approaches excluding NDN. We already show that NDN is worse than VCDN and it appears that VCDN is the worst approach of the remaining ones, which that is why in the following figures we will not show NDN. Yet, the results will appear in the comparison tables. Using RENE, the vehicles send more interest than VCDN (one per sensor), but only the updated sensors send their data. For VCDN, the data of every sensor are sent when at least one sensor is updated. Because the data packet's size is higher than the interest's one, the performances of RENE are better than VCDN when the number of vehicles increases. The best performance is obtained for VCDNoCAM since the number of vehicles has no effect on the network overhead caused by the data sending. Using GeoVCDN, we add the exchanges between vehicles out of the coverage area and that is why GeoVCDN's overhead is higher than VCDNoCAM's. Still, despite this mechanism that increases the network overhead, GeoVCDN is more scalable than RENE.
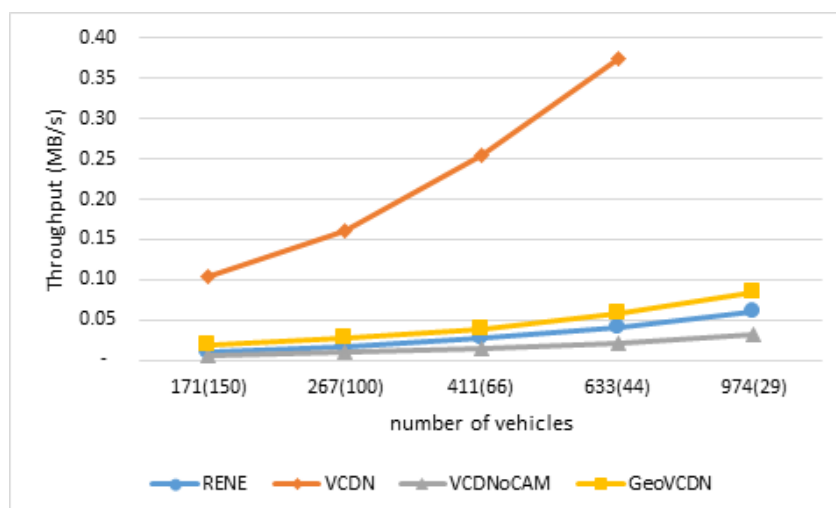

Figure 19. Number of Vehicles.

In order to validate the proposed model, we present, in Table 10, a comparison between the expected values from the analytical model and the observed values from the simulations for 411 vehicles.

Table 10. Network Overhead Comparison for Number of Vehicles.

Approach	Expected Value	Observed Value	Difference
NDN	3,507,200.00	3,712,913.93	+5.86%
RENE	27,021.88	27,453.20	+1.5%
VCDN	252,934.88	254,684.26	+0.69%
VCDNoCAM	14,596.58	14,678.86	+0.56%
GeoVCDN	40,220.45	39,822.73	−0.99%

We note a difference around of 1%, which confirms our model's behavior based on the number of vehicles.

Furthermore, considering that the range of the vehicle is lower than the RSU's (which is a realistic consideration), this only has an impact on the GeoVCDN approach, because it is the only one which handles the V2V communications, as we can see in Figure 20. We have a smoother increase between 50 and 100, which refers to the phase when the inter-distance

is almost sufficient enough to have all vehicles connected together. In this simulation, we had a 107-meter inter-distance. After this threshold, we observe a slow decrease, as the number of hops to cover a segment decreases.
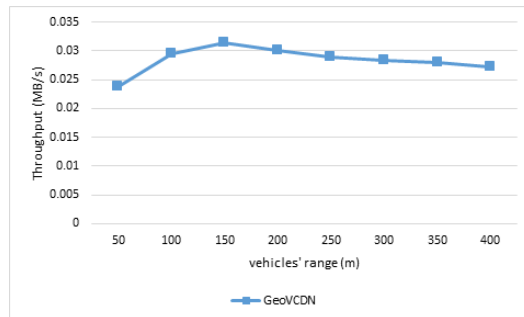

Figure 20. Vehicles' range.

For a vehicle range of 150 m, in Table 11, the difference between the model and the simulation is very low (about 0.5%), except for the NDN.

Table 11. Network Overhead Comparison for Vehicles' Range.

Approach	Expected Value	Observed Value	Difference
NDN	5,126,400.00	5,422,778.06	+6%
RENE	28,483.56	28,486.66	+0.01%
VCDN	251,951.88	253,434.53	+0.59%
VCDNoCAM	15,049.23	15,079.13	+0.20%
GeoVCDN	31,614.90	31,383	−0.73%

In an intelligent transport system network, RSUs are the gateway between the vehicles and the rest of the world. Increasing the number or the range of RSUs will increase the number of data access points. Thus, in most cases, the network overhead will increase. We will show how the network overhead is affected by the number or the quality of RSUs for each approach.

Increasing the number of RSUs makes the coverage area bigger, which increases the number of vehicles concerned by the ICN messages. Furthermore, it also increases the number of entering points to the coverage area. The increase in the number of RSUs has a good impact on the GeoVCDN approach, as we can see in Figure 21. The more RSUs there are, the less there are exchanges between vehicles, since the RSUs already cover a part of the city. Additionally, we see that for RENE and VCDN, raising the number of RSUs leads to a significant increase in the overhead. For RENE, the overhead generated with interests is related to the number of entering points to the coverage area, whereas for VCDN it is about the coverage area itself. Concerning the data, with RENE and VCDN there are more vehicles to which the data must be delivered.
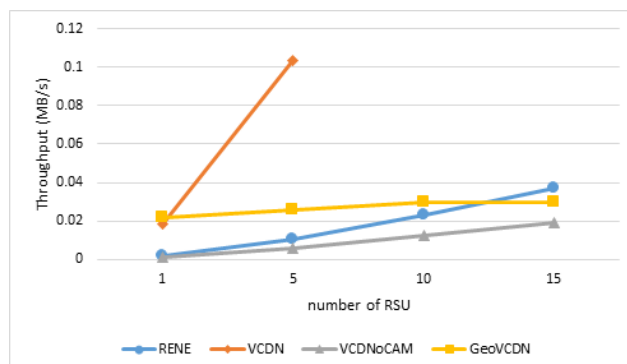

Figure 21. Number of RSUs.

Table 12 lists a comparison between the expected values and the observed ones for five RSUs.

Table 12. Network Overhead Comparison for Number of RSUs.

Approach	Expected Value	Observed Value	Difference
NDN	890,000.00	956,702.13	+7.49%
RENE	10,298.97	10,469.80	+1.69%
VCDN	100,984.96	103,212.86	+2.2%
VCDNoCAM	5889.28	6001.46	+1.90%
GeoVCDN	27,812.43	25,939.06	−6.73%

For RENE, VCDN, and VCDNoCAM, the difference between the analytic model and the simulation's results is around 2% (Table 12). For the NDN approach, the simulation gives a result 7% higher, whereas for GeoVCDN, the result of the simulation is lower than the one given by the analytic model (7%).

Furthermore, the RSUs' range modifies the coverage area and so the number of vehicles receiving the data (Figure 22). For the VCDN case, the consequence is the same as if we increase the number of vehicles: more interest and more data. For RENE, the coverage area does not affect the overhead related to the interests, which represents the biggest part of the overhead for this approach. However, we send the data to more vehicles, which is why we can observe a slight increase. Using VCDNoCAM, the effect of the coverage area is null on the overhead related to the data. It only affects the overhead related to the interests. GeoVCDN is the unique approach, tending to decrease the throughput when increasing the RSUs' range because fewer V2V exchanges are needed to spread the data.
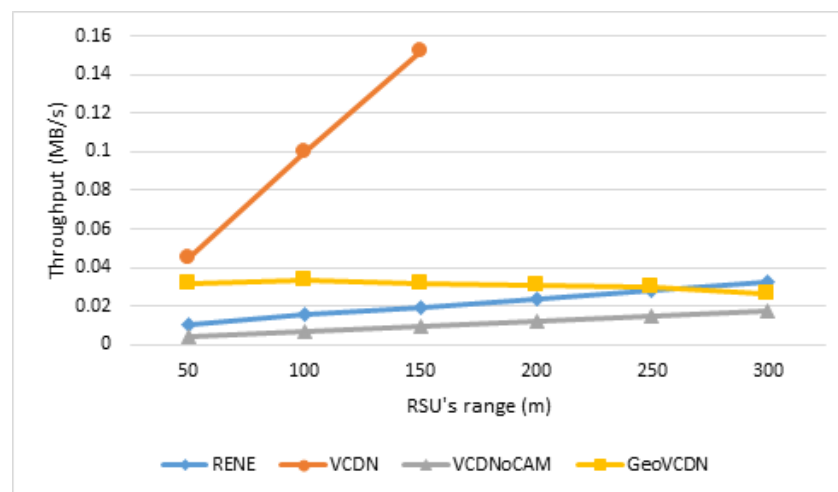

Figure 22. RSUs' range.

Table 13 presents a comparison between the expected values and the observed values with the RSU's range of 150 m.

Table 13. Network Overhead Comparison for RSUs' Range.

Approach	Expected Value	Observed Value	Difference
NDN	3,075,840.00	3,240,097.20	+5.34%
RENE	19,672.56	19,734.53	+0.32%
VCDN	151,559.88	151,847.26	+0.19%
VCDNoCAM	9709.23	9680.13	−0.30%
GeoVCDN	30,725.85	31,360.93	+2.07%

Again, the table confirms the analytic model when the RSU's range changes. Indeed, we can see that the difference between the model and the simulation is very low (0–2%).

For VCDN, RENE, and VCDNoCAM, the increase of the data generation frequency leads to a similar behavior for the overhead related to it, as we can see in Figure 23: with the increase of the data generation frequency, we increase the data amount transiting in the network. Using VCDNoCAM, we are limited by the CAM frequency (10 Hz), which is why the network overhead is constantly passed. We can see that with GeoVCDN, the higher frequency is, the fewer exchanges between vehicles. Passed 10 Hz, the exchanges between vehicles decrease (because the maximum hop limit is computed based on the data generation frequency). For this reason, from this point, the higher the frequency, the lower the consumed bandwidth within the GeoVCDN approach. Table 14 confirms that the difference between the expected and measured values for a frequency X is very low (between 1 and 2%).

Table 14. Network Overhead Comparison for Data Update Frequency.

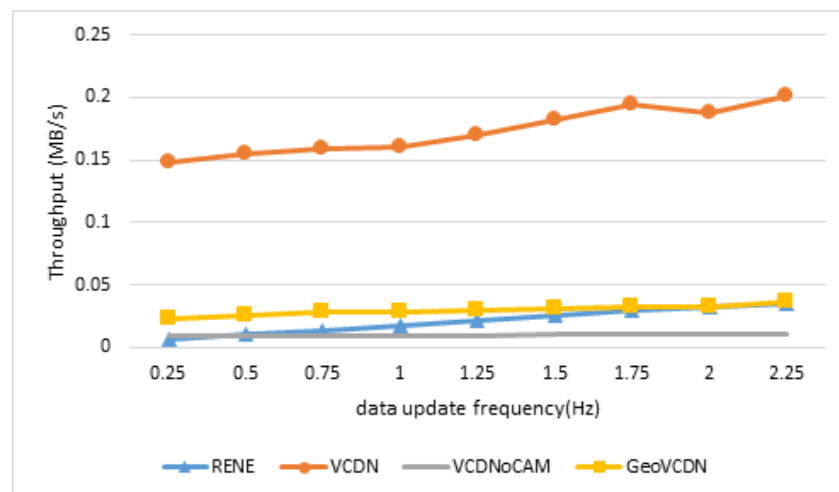| Approach	Expected Value	Observed Value	Difference
NDN	2,278,400	2,338,049.13	+2.6%
RENE	17,554.36	17,281.2667	−1.55%
VCDN	164,315.36	160,374.2	−2.39%
VCDNoCAM	9684.26	9462.53	−2.28%
GeoVCDN	28,900.20	28,399.6	−1.73%

Figure 23. Data Update Frequency.

5.1.3. Data Freshness Simulation Results

The data freshness is measured as the percentage of vehicles which receives the data before the latter expires. The data expire when updated data have been generated. In our environment, the data frequency (σ) fixes the validity duration. We will compare the behavior of each approach by varying some parameters associated with the vehicles (number and range), the RSUs (number and range), and the data (frequency).

Because the analytic model is pessimistic about the data freshness associated with GeoVCDN, the simulation will always give better results. The gap between the two results will be higher if there are more or fewer RSUs because the analytic model does not take into account the dissemination to the segments without RSUs.

On one hand, the communication range of the vehicles has no effect on the approach that does not handle the V2V exchanges, as we can see in Figure 24. However, for the GeoVCDN approach, where the vehicle's transmission range is high enough to communicate with the surrounding vehicles, the freshness increases until it quickly reaches 100%, if the hop limit allows it. For example, with a range of 150 m, with GeoVCDN, we reach a

freshness of 100% compared to 60% with the other approaches. With a range lower than the inter-distance, we still do better than the other approaches with almost 16% more vehicles receiving the data, thanks to the bidirectional traffic. The more we increase the range, the more we optimize the broadcast from client to client; finally reaching the optimum freshness if the other parameters allow it.
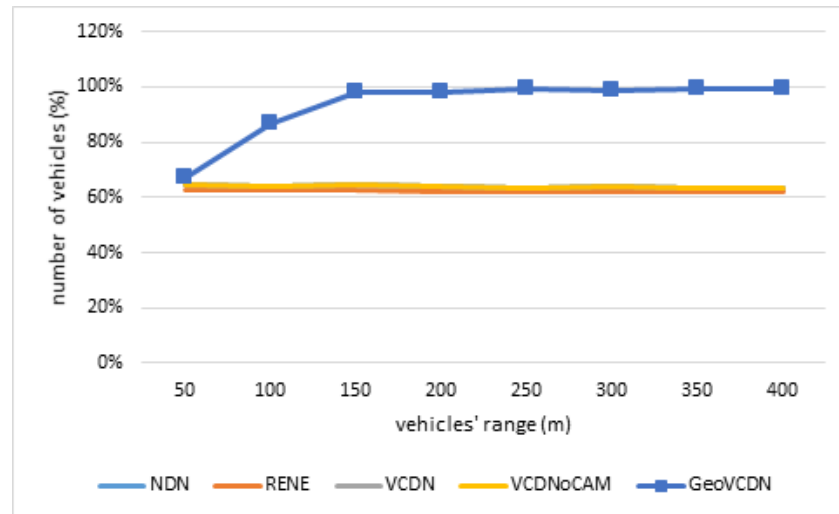

Figure 24. Vehicles' Range.

Table 15 presents a comparison between the expected values and the observed values for a vehicle range of 150 m.

Table 15. Data Freshness Comparison for Vehicles' Range.

Approach	Expected Value	Observed Value	Difference
NDN	64.13%	64%	−0.13 pt
RENE	62.50%	63%	+0.5 pt
VCDN	64.13%	64%	−0.13 pt
VCDNoCAM	64.13%	64%	−0.13 pt
GeoVCDN	95%	98%	+3 pt

The differences between the obtained values with the model and with the simulations are small (max three points).

As shown in Figure 25, where $\beta = 8$, increasing the number of vehicles does not have any effect on the approaches that do not handle the V2V exchanges. As expected, the percentage of vehicles in the coverage area and entering the coverage area before the expiration of the data remains the same, since the vehicle distribution is homogeneous and their speed is constant, so the freshness is also constant. Because the freshness is expressed as a percentage, the absolute amount of vehicles has importance only for the GeoVCDN approach. With this one, the increase in the number of vehicles has a positive effect on the freshness because we optimize our transmission range and our farthest neighbor is located farther away, so we cover a bigger area. Let us note that the freshness is limited by other settings such as the hop limit. With 170 vehicles, the GeoVCDN approach sees its freshness 50 points above the others. With 1000 vehicles, the performance is even better with 99% of the vehicles which obtain the information before it expires against 40% with the other approaches. The performance increase is logarithmically evolving according to the inter-vehicle distance.
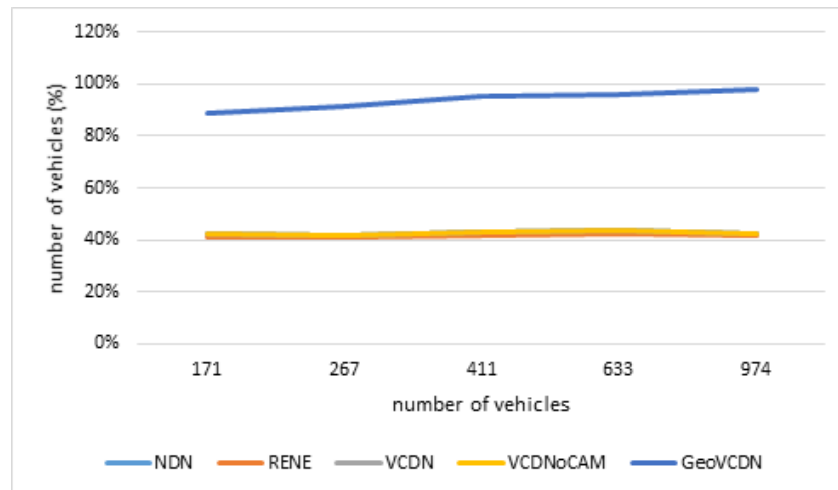
Figure 25. Number of Vehicles.

The comparison between the expected values and the observed values for 267 vehicles are listed in Table 16.

Table 16. Data Freshness Comparison for Number of Vehicles.

Approach	Expected Value	Observed Value	Difference
NDN	42.75%	42%	−0.75 pt
RENE	41.67%	41%	−0.67 pt
VCDN	42.75%	42%	−0.75 pt
VCDNoCAM	42.75%	42%	−0.75 pt
GeoVCDN	76.67%	91%	+24.33 pt

These results confirm the validity of the analytic model when we make the number of vehicles change. In this scenario, about a quarter of the segments are not covered by any RSU, which is why the analytic model gave such results.

Increasing the range of the RSUs means increasing the number of vehicles within their ranges. In Figure 26, we observe that the consequence is the same for all the approaches with better performance for GeoVCDN. The more we increase the range of the RSUs, the more we reduce the hops that the messages have to do using the V2V exchanges. We note that for the GeoVCDN approach, in the end, the freshness reaches 100%.
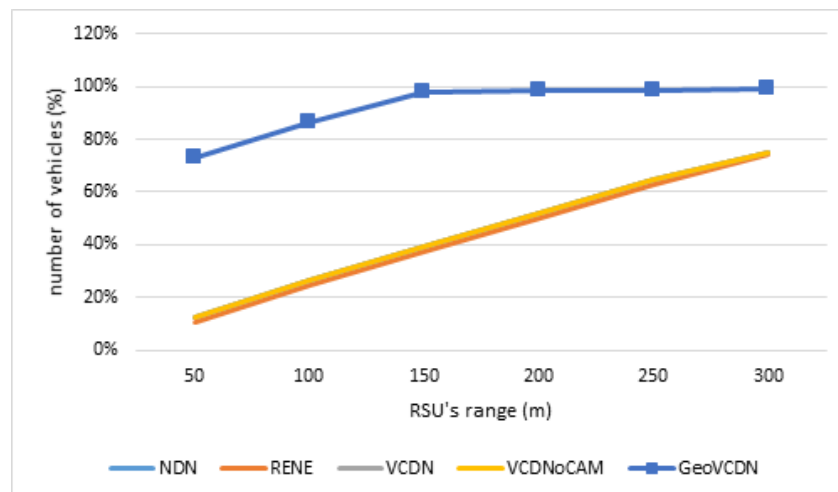

Figure 26. RSUs' Range.

A comparison between the expected values and the observed ones for an RSU transmission range of 100 m is presented in Table 17.

Table 17. Data Freshness Comparison for RSUs' Range.

Approach	Expected Value	Observed Value	Difference
NDN	26.63%	26%	−0.63 pt
RENE	25%	25%	0 pt
VCDN	26.63%	26%	−0.63 pt
VCDNoCAM	26.63%	39%	−0.63 pt
GeoVCDN	95%	98%	+3 pt

For the approaches NDN, RENE, VCDN, and VCDNoCAM, the difference in the results is almost null.

In Figure 27, we see that increasing the number of RSUs has a positive effect on the freshness of all the approaches. It increases the number of entering points of the coverage area, which increases the number of vehicles being in it when the data are generated. For GeoVCDN, increasing the number of RSUs means increasing the number of starting points for the propagation of the data in V2V, which explains that the curve increases quicker than the other approach For GeoVCDN, the more RSUs there are, the more vehicles are able to forward the data. This is why we have a logarithmic curve in Figure 27. A comparison between the expected values and the observed ones is given in Table 18.
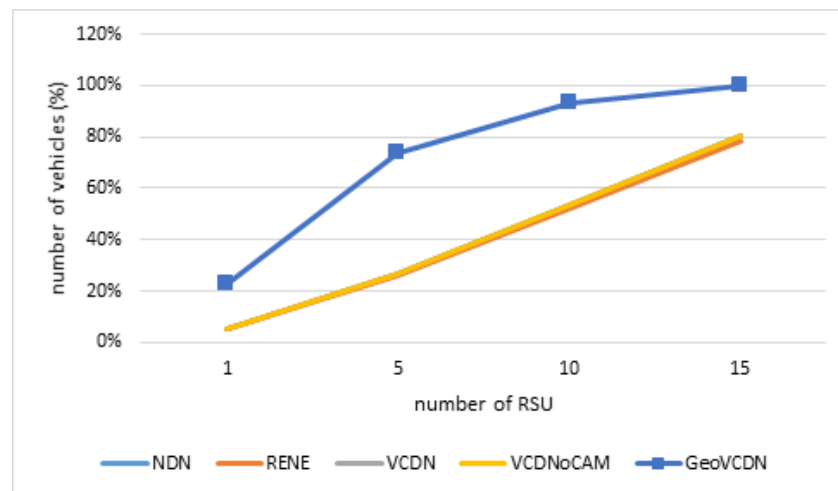

Figure 27. Number of RSUs.

Table 18. Data Freshness Comparison for Number of RSUs.

Approach	Expected Value	Observed Value	Difference
NDN	26.72%	27%	+0.28 pt
RENE	25.04%	26%	−0.4 pt
VCDN	26.72%	27%	−0.28 pt
VCDNoCAM	26.72%	27%	−0.28 pt
GeoVCDN	54.17%	74%	+19.83 pt

Again, since almost half of the segments do not have any RSU, the analytic model gave a worse result than the simulation for GeoVCDN.

By increasing the data generation frequency, we decrease the hop limit and so the number of vehicles, which can be affected by the V2V communications (GeoVCDN). For VCDN, NDN, and VCDNoCAM, we observe, in Figure 27, a slight loss, caused by the decrease in the number of vehicles entering the coverage between two generations. For

RENE, whatever the frequency is, the number of vehicles inside the coverage is constant, so the freshness also remains constant.

Finally, the more often data are sent, the less time the vehicles have to forward them. That is why, as we can see in Figure 28, the freshness decreases only with the GeoVCDN approach. For RENE, the frequency does not have any effect. For the others, the effect is almost null, there are fewer vehicles entering the coverage area before the expiration of the data and that is why we observe a slight loss. Despite this decrease in GeoVCDN, it outperforms all other approaches even with high data frequency updates thanks to the benefits of V2V communications.
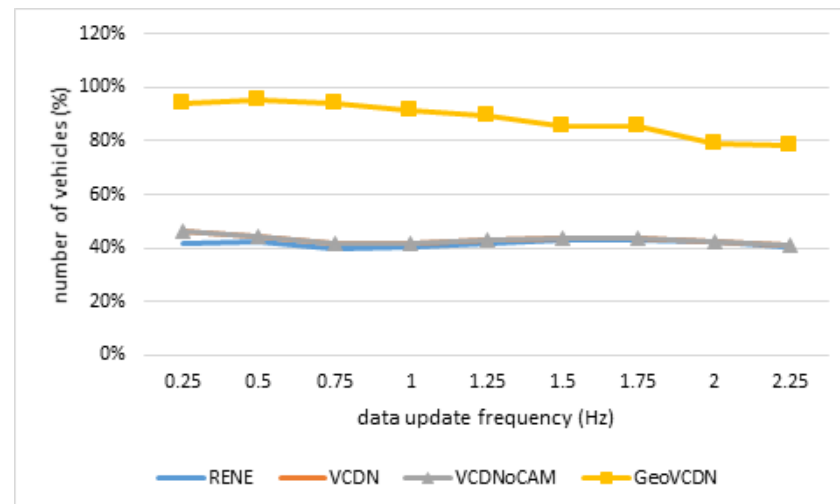

Figure 28. Update Data Frequency.

5.1.4. Discussion

This analytic evaluation highlights the efficiency of our approaches. VCDNoCAM provides the same service as already existing ones (NDN and RENE) for freshness, but with a lower cost on the network overhead. The NDN method gives disastrous results in an environment where the data are numerous and often change; resulting in the need to send interests very often for all data to ensure to keep them updated. NDN has been thought to make the network traffic lighter without taking into account the link to the client. Yet in vehicles, the G5 channel has a reduced bandwidth, and we have to save it as much as possible. So, if want to guarantee the authenticity and integrity of every message, it quickly becomes cost ineffective. We saw that we managed to save this network overhead related to the security layer by grouping a set of data in one set of metadata, which is heavier, but makes it possible to reduce the number of security data to only one occurrence. We also added a sequence number mechanism, which avoids sending the data to someone who already has them. Despite these two mechanisms, still, the results are not satisfying (VCDN). The RENE method is more adapted to such an environment because the data are only pushed when it changes. However, as with NDN, the final link towards the client is not taken into account and every vehicle in the same RSU's area is going to individually receive the information. By exploiting the existing technologies, we manage to widely reduce the cost of the data related to the ICN exchanges (VCDNoCAM), by integrating these exchanges in the standardized regularly broadcasted messages (CAMs). Thus, we use the CAM's security layer. In addition, the data are sent to every vehicle in range with only one message. The results highlight the gain in network overhead with this method. Finally, we also use CAMs to send the data farther than only RSU's range (GeoVCDN). This additional functionality improves the quality of service but has a cost on the network overhead. However, it stays lower than the existing ICN methods, and, as we saw, it is more scalable.

5.2. Use Case Simulation Study

In this part, we propose to apply our data exchange method, in a context cloud, to the pathfinding problem.

The used simulator is the same as the one presented in Figure 1: a city such as Manhattan and the vehicles are uniformly spread. First, we place traffic lights and RSU on five fixed junctions. A traffic light can be in two states: quick or slow. A quick state means that it will switch between red and green every 5 s and a slow state means that it will switch every 15 s. Then, we randomly select one or several junctions that will also be equipped with RSUs. At the beginning of the simulation, we insert three additional vehicles. One uses the NDN approach, one uses RENE, and the last vehicle uses GeoVCDN. In their initial state, each vehicle has the knowledge of the configuration topology and the state of all the traffic lights in the city and, then, they must obtain the same result for their pathfinding algorithm. After 2 s, every traffic light toggles its state, switching from quick to slow or from slow to quick which has for effect to change the shortest path. An ICN message is emitted to inform about this change. When the change occurs, if the vehicles are in the range of an RSU, they will receive the information and can adapt their paths. If they are not in the range, a vehicle with NDN will receive the information as soon as it will be in the range of an RSU and a vehicle with GeoVCDN will receive the information from other vehicles. However, a vehicle with RENE will not receive it since it is based on push mode.

Figure 29 is extracted from the simulator and represents a junction. An RSU is placed at the center and a traffic light is placed on each entry. The traffic lights on the line are synchronized. The red circle represents the transmission range of the RSU. The yellow square represents a vehicle with GeoVCDN.
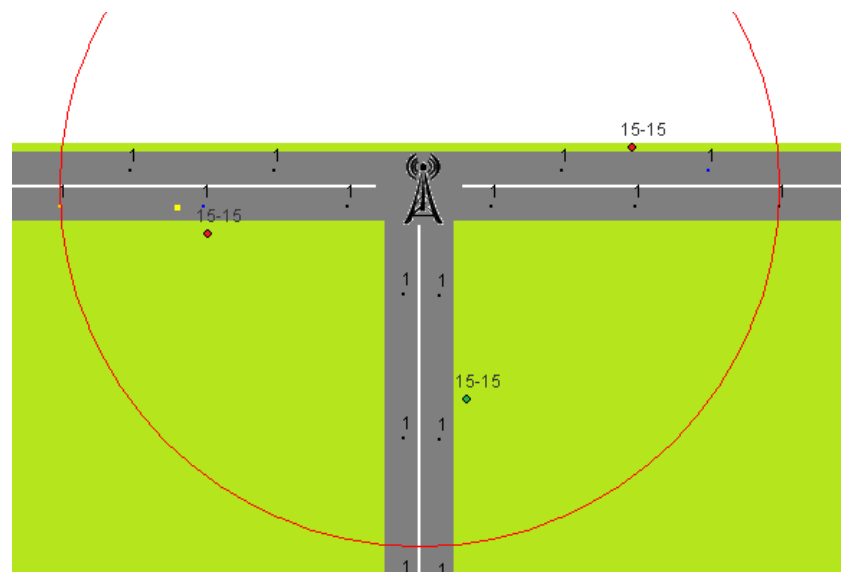

Figure 29. Junction configuration.

First, we randomly place RSUs in addition to the already fixed five. In the first configuration, only one RSU is randomly placed. In the second, two RSUs are randomly placed, and then up to eight for the third configuration. Finally, for the fourth one, an RSU is placed at every junction. For every configuration, we launched 100 simulations (the location of the RSU changes at each run). In the end, we measure the time needed for a vehicle to cross the city.

Figure 30 presents the results for the first configuration. The vehicle using GeoVCDN spends 2989 s on the road, whereas the other vehicles (with NDN or RENE) spent 3566 s, which is an improvement of 17% for the GeoVCDN approach.
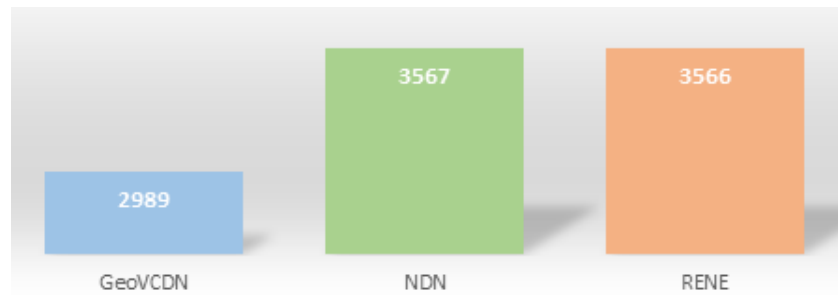
Figure 30. Path-finding with 1 random RSU.

Figure 31a–c show the results for other configurations. We observe that the more RSUs there are, the more the results are closer. With RSUs placed everywhere, there is only a 1 or 2% difference. With few RSUs, the initial pathfinding had the effect of dodging the traffic light to gain time. Ironically, by dodging it, the vehicles also made it impossible for them to receive the updated information. By increasing the number of RSUs, we also increase the probability for a vehicle with NDN or RENE to receive the information, because it becomes more and more complicated to dodge every traffic light.

(a)

(b)

(c)

Figure 31. Path finding for different numbers of RSUs. (**a**) 2 Random RSUs; (**b**) 8 Random RSUs; (**c**) 11 Random RSUs.

5.3. Discussion

In the analytic model, we made two assumptions: (1) for the network overhead, the dissemination of the messages was maximized and (2) for the freshness, it was minimized. Thus, it was expected that the simulation gives better results than the analytic model for our approach. The results we obtained match the expected ones. So, we have been able to validate our model thanks to simulation. The analytic expression of our approach could be perfected, yet we have been able to confirm that our approach is better than the others in the worst cases. Table 19 presents an overview of the behavior of each approach when it comes to large-scale situations, and it is not based on any threshold. It is based on the results

obtained with each approach, on both performance indicators that we studied, which are network overhead and data freshness.

Table 19. Approaches' Performances.

		Parameters	NDN	RENE	VCDN	VCDNoCAM	GeoVCDN
Network's	Overhead	Number of vehicles	Very High	Middle	High	Very Low	Low
		Segment's length	Very High	Low	High	Very Low	Middle
		RSUs' range	Very High	Middle	High	Very Low	Middle
		Coverage area	Very High	Middle	High	Very Low	Low
		Vehicles' range	Very High	Low	High	Very Low	Low
		Number of RSUs	Very High	Middle	High	Very Low	Very Low
		Data update frequency	Very High	Low	High	Very Low	Very Low
Data's	Freshness	Number of vehicles	Bad	Bad	Bad	Bad	Very Good
		RSUs' range	Middle	Middle	Middle	Middle	Very Good
		Vehicles' range	Bad	Bad	Bad	Bad	Very Good
		Number of RSUs	Middle	Middle	Middle	Middle	Good
		Data update frequency	Bad	Bad	Bad	Bad	Middle

Whether it is for the analytic models or the simulations, we improve the results when we increase the density or the number of sensors. The NDN approach is absolutely not worthy for the vehicular network, as VCDN does better and is already too high compared to the three others. For better freshness and lower cost on bandwidth, the VCDNoCAM is better than the RENE approach, as we can see in the table above. On the other hand, GeoVCDN gives a bit higher network overhead than VCDNoCAM but remains better than RENE. This additional cost compared to VCDNoCAM is widely compensated by the gain in freshness. By comparing VCDN and NDN, we note that the sequence number has a positive effect on the network overhead even if the data are bigger. By comparing VCDNoCAM and RENE, we note that the usage of CAMs to provide ICN exchanges also positively affects the network overhead. With VCDNoCAM, for freshness, it performs as well as NDN and better than RENE, all of that at a lower cost. Our approach is more expansive in terms of bandwidth consumption than RENE and NDN when there are few vehicles or few sensors, but in this case, it is less important to optimize the network overhead. Finally, the values obtained during the simulations are very close to the ones expected from the model and, therefore, validate them. Additionally, with GeoVCDN, we are limited by the CAM frequency, which affects the latency and could make our approach not adapted for some use cases. Yet, using our approach frees some space on the bandwidth for the use cases which need low latency. As we showed with the use case of pathfinding, we have both better quality of service and lower usage of the bandwidth.

6. Conclusions

In this paper, we have presented our approach based on a context cloud architecture formed by C-ITS stations. We have considered a Smart City environment for our study, but it can be easily adapted to any kind of environment: a county, a mall, a region, a part of a road, etc. We have used the ICN paradigm for data exchanges within the cloud. We have selected the ICN approach because it fits sensory vehicular networks (data-oriented, instead of host-oriented) and it eases mobility handling. We have included ICN interests and data within the standardized cooperative awareness messages (CAM), which means that our approach can be adopted in the currently deployed C-ITS without any modification of the standards. Additionally, we have used a C-ITS central station to be able to merge the data of different sensors if it is necessary and to add a sequence number, which allows avoiding sending non-updated data.

We have compared our approach to two existing solutions, one push-oriented method (RENE) and another pull-oriented method (NDN), in terms of the network overhead and data freshness metrics. For this purpose, we have proposed a probabilistic model, which takes into account the mobility of the vehicles, the number and the random location of roadside units (RSU), and the message dissemination method from vehicle to vehicle. In

order to confirm the results obtained from the analytic evaluation, we have developed a simulator that takes into account the standardized ETSI communication stack, the mobility model, and the ICN exchanges. The slight differences observed between the simulation and the numeric results (in general, less than 2%) can be explained by the time-discrete simulator limits and the assumptions made on the analytic model.

Our results are promising, as one can see in Table 19, both network throughput and data freshness are improved. Furthermore, we have shown that our approach is more scalable than the other approaches since it has better results for a large amount of data, a high number of vehicles, and a high number of RSUs (less network overhead and better data freshness).

Our future works will focus on testing our proposed approach in a real environment. Moreover, we intend to explore some machine learning methods to be implemented on the central station, which would be able to adapt its naming strategy in real time, based on different parameters, such as the number of sensors and the data update frequency.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

CAM	Cooperative Awareness Message
OBU	On-Board Unit
C-ITS	Cooperative Intelligent Transportation System
PC	Pseudonym Certificate
ETSI	European Telecommunications Standards Institute
PCA	Pseudonym Certificate Authority
IT	Information Technology
PKI	Public Key Infrastructure
ITSS	Intelligent Transportation System Station
ROV	Road Operator Vehicle
LTC	Long-Term Certificate
RSU	Roadside Unit
LTCA	Long-Term Certificate Authority
SUMO	Simulation of Urban Mobility
NS3	Network Simulator 3
VANET	Vehicular Ad Hoc Networks

References

1. Djama, A.; Djamaa, B.; Senouci, M.R. Information-Centric Networking solutions for the Internet of Things: A systematic mapping review. *Comput. Commun.* **2020**, *159*, 37–59. [CrossRef]
2. Musaddiq, A.; Ali, R.; Bajracharya, R.; Qadri, Y.A.; Al-Turjman, F.; Kim, S.W., Trends, Issues, and Challenges in the Domain of IoT-Based Vehicular Cloud Network. In *Unmanned Aerial Vehicles in Smart Cities*; Al-Turjman, F., Ed.; Springer International Publishing: Cham, Switzerland, 2020; pp. 49–64. [CrossRef]
3. El Hamdani, S.; Benamar, N.; Younis, M. Pedestrian Support in Intelligent Transportation Systems: Challenges, Solutions and Open issues. *Transp. Res. Part C Emerg. Technol.* **2020**, *121*, 102856. [CrossRef]
4. Festag, A. Cooperative intelligent transport systems standards in Europe. *IEEE Commun. Mag.* **2014**, *52*, 166–172. [CrossRef]
5. *EN 302 636-3 V1.2.1*; Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture. ETSI: Antibes, France, 2014.

6. *TS 102 724 (V1.1.1)*; Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems Operating in the 5 GHz Frequency Band. ETSI: Antibes, France, 2012.
7. *TS 102 687 (V1.2.1)*; Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems Operating in the 5 GHz Range; Access Layer Part. ETSI: Antibes, France, 2018.
8. *TS 103 175 (V1.1.1)*; Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for Operation in the ITS G5A and ITS G5B Medium. ETSI: Antibes, France, 2015.
9. *EN 302 663 (V1.3.1)*; Intelligent Transport Systems (ITS); ITS-G5 Access Layer Specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band. ETSI: Antibes, France, 2019.
10. *EN 302 637-2 (V1.4.1)*; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of CCooperative Awareness Basic Service. ETSI: Antibes, France, 2019.
11. *EN 302 637-3 (V1.3.1)*; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. ETSI: Antibes, France, 2019.
12. *TS 19091*; Intelligent Transport Systems—Cooperative ITS—Using V2I and I2V Communications for Applications Related to Signalized Intersections. ISO: Geneva, Switzerland, 2019.
13. *EN 103 097 (V1.3.1)*; Intelligent Transport Systems (ITS); Security; Security Header and Certificate Formats. ETSI: Antibes, France, 2017.
14. Petrolo, R.; Loscrì, V.; Mitton, N. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Trans. Emerg. Telecommun. Technol.* **2015**, *28*, e2931. [CrossRef]
15. Bitam, S.; Mellouk, A. ITS-cloud: Cloud computing for Intelligent transportation system. In Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 2054–2059. [CrossRef]
16. Zaslavsky, A.; Perera, C.; Georgakopoulos, D. Sensing as a Service and Big Data. *arXiv* **2013**. [CrossRef]
17. Ijemaru, G.K.; Ang, L.M.; Seng, K.P. Transformation from IoT to IoV for waste management in smart cities. *J. Netw. Comput. Appl.* **2022**, *204*, 103393. [CrossRef]
18. Rao, B.B.P.; Saluia, P.; Sharma, N.; Mittal, A.; Sharma, S.V. Cloud computing for Internet of Things amp; sensing based applications. In Proceedings of the 2012 Sixth International Conference on Sensing Technology (ICST), Kolkata, India, 18–21 December 2012; pp. 374–380. [CrossRef]
19. Suciu, G.; Vulpe, A.; Halunga, S.; Fratu, O.; Todoran, G.; Suciu, V. Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things. In Proceedings of the 2013 19th International Conference on Control Systems and Computer Science, Bucharest, Romania, 29–31 May 2013; pp. 513–518. [CrossRef]
20. Whaiduzzaman, M.; Sookhak, M.; Gani, A.; Buyya, R. A survey on vehicular cloud computing. *J. Netw. Comput. Appl.* **2014**, *40*, 325–344. [CrossRef]
21. Morello, R.; Mukhopadhyay, S.C.; Liu, Z.; Slomovitz, D.; Samantaray, S.R. Advances on Sensing Technologies for Smart Cities and Power Grids: A Review. *IEEE Sens. J.* **2017**, *17*, 7596–7610. [CrossRef]
22. Feng, C.; Han, P.; Zhang, X.; Yang, B.; Liu, Y.; Guo, L. Computation offloading in mobile edge computing networks: A survey. *J. Netw. Comput. Appl.* **2022**, *2022*, 103366. [CrossRef]
23. Gerla, M. Vehicular Cloud Computing. In Proceedings of the 2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Ayia Napa, Cyprus, 19–22 June 2012; pp. 152–155. [CrossRef]
24. Khan, W.Z.; Ahmed, E.; Hakak, S.; Yaqoob, I.; Ahmed, A. Edge computing: A survey. *Future Gener. Comput. Syst.* **2019**, *97*, 219–235. [CrossRef]
25. Ghafoor, K.; Abu Bakar, K.; Mohammed, M.; Lloret, J. *Vehicular Cloud Computing: Trends and Challenges*; IGI Global: Hershey, PA, USA, 2013; Volume 2. [CrossRef]
26. Qiu, T.; Liu, X.; Li, K.; Hu, Q.; Sangaiah, A.K.; Chen, N. Community-Aware Data Propagation with Small World Feature for Internet of Vehicles. *IEEE Commun. Mag.* **2018**, *56*, 86–91. [CrossRef]
27. Gomides, T.S.; Robson, E.; Meneguette, R.I.; De Souza, F.S.; Guidoni, D.L. Predictive Congestion Control based on Collaborative Information Sharing for Vehicular Ad hoc Networks. *Comput. Netw.* **2022**, *211*, 108955. [CrossRef]
28. Zheng, K.; Meng, H.; Chatzimisios, P.; Lei, L.; Shen, X. An SMDP-Based Resource Allocation in Vehicular Cloud Computing Systems. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7920–7928. [CrossRef]
29. Farooq, M.U.; Pasha, M.; Khan, K.U.R. Contextual communication patterns for vehicular cloud computing: Towards realizing Internet of Things. In Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016; pp. 1–6. [CrossRef]
30. Cheriton, D.; Gritter, M. TRIAD: A New Next-Generation Internet Architecture 2000. Available online: http://www-dsg.stanford.edu/triad/ (accessed on 9 March 2023).
31. Thomas, Y.; Fotiou, N.; Toumpis, S.; Polyzos, G. Improving mobile ad hoc networks using hybrid IP-Information Centric Networking. *Comput. Commun.* **2020**, *156*, 25–34. [CrossRef]
32. Bo Yu, C.X. Vehicular Ad-Hoc Networks:An Information-Centric Perspective. *ZTE Commun.* **2010**, *8*, 42.
33. Ahlgren, B.; Dannewitz, C.; Imbrenda, C.; Kutscher, D.; Ohlman, B. A survey of information-centric networking. *IEEE Commun. Mag.* **2012**, *50*, 26–36. [CrossRef]
34. Alhowaidi, M.; Nadig, D.; Hu, B.; Ramamurthy, B.; Bockelman, B. Cache management for large data transfers and multipath forwarding strategies in Named Data Networking. *Comput. Netw.* **2021**, *199*, 108437. [CrossRef]

35. Amadeo, M.; Ruggeri, G.; Campolo, C.; Molinaro, A. Diversity-improved caching of popular transient contents in Vehicular Named Data Networking. *Comput. Netw.* **2021**, *184*, 107625. [CrossRef]

36. Zahemszky, A.; Gajic, B.; Esteve Rothenberg, C.; Reason, C.; Trossen, D.; Lagutin, D.; Tuononen, J.; Katsaros, K. Experimentally-Driven Research in Publish/Subscribe Information-Centric InterNetworking. *Testbeds Res. Infrastruct. Dev. Netw. Commun.* **2010**, *46*, 469–485. [CrossRef]

37. Dannewitz, C. NetInf: An Information-Centric Design for the Future Internet. 2013. Available online: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=25216eed4449a2c08b11f86133ce5c543cc12701 (accessed on 9 March 2023).

38. Yi, B.; Wang, X.; Huang, M. Content delivery enhancement in Vehicular Social Network with better routing and caching mechanism. *J. Netw. Comput. Appl.* **2021**, *177*, 102952. [CrossRef]

39. Xylomenos, G.; Ververidis, C.N.; Siris, V.A.; Fotiou, N.; Tsilopoulos, C.; Vasilakos, X.; Katsaros, K.V.; Polyzos, G.C. A Survey of Information-Centric Networking Research. *IEEE Commun. Surv. Tutorials* **2014**, *16*, 1024–1049. [CrossRef]

40. Amadeo, M.; Campolo, C.; Molinaro, A. Information-centric networking for connected vehicles: A survey and future perspectives. *IEEE Commun. Mag.* **2016**, *54*, 98–104. [CrossRef]

41. Amadeo, M.; Campolo, C.; Quevedo, J.; Corujo, D.; Molinaro, A.; Iera, A.; Aguiar, R.L.; Vasilakos, A.V. Information-centric networking for the internet of things: Challenges and opportunities. *IEEE Netw.* **2016**, *30*, 92–100. [CrossRef]

42. Liu, X.; Li, Z.; Yang, P.; Dong, Y. Information-Centric Mobile Ad Hoc Networks and Content Routing. *Ad Hoc Netw.* **2017**, *58*, 255–268. [CrossRef]

43. Boukerche, A.; Coutinho, R.W.L.; Yu, X. LISIC: A Link Stability-Based Protocol for Vehicular Information-Centric Networks. In Proceedings of the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, USA, 22–25 October 2017; pp. 233–240. [CrossRef]

44. Tarroumi, M.; Jabri, I. EVNDN: Enhanced vehicular named data networking. In Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2017; pp. 1–6. [CrossRef]

45. Wang, L.; Afanasyev, A.; Kuntz, R.; Vuyyuru, R.; Wakikawa, R.; Zhang, L. Rapid Traffic Information Dissemination Using Named Data. In Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design—Architecture, Algorithms, and Applications, NoM '12, Hilton Head, SC, USA, 11 June 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 7–12. [CrossRef]

46. Grewe, D.; Wagner, M.; Frey, H. A Domain-Specific Comparison of Information-Centric Networking Architectures for Connected Vehicles. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2372–2388. [CrossRef]

47. Bouk, S.H.; Ahmed, S.H.; Kim, D.; Song, H. Named-Data-Networking-Based ITS for Smart Cities. *IEEE Commun. Mag.* **2017**, *55*, 105–111. [CrossRef]

48. Information Centric Networking 2020. Available online: www.icn2020.org/information-centric-networking/ (accessed on 9 March 2023).