electronics

MDPI

Article

CMAF: Context and Mobility-Aware Forwarding Model for V-NDN

Elídio Tomás da Silva [1,2,*], Joaquim Macedo [2] and António Costa [2]

[1] Department of Informatics Engineering, Faculty of Engineering, Lurio University, Pemba 3203, Mozambique
[2] Algoritmi Centre, University of Minho, 4710-057 Braga, Portugal; macedo@di.uminho.pt (J.M.); costa@di.uminho.pt (A.C.)
[*] Correspondence: id6644@alunos.uminho.pt

Abstract: Content dissemination in Vehicular Ad hoc Networks (VANET) is a challenging topic due to the high mobility of nodes, resulting in the difficulty of keeping routing tables updated. State-of-the-art proposals overcome this problem by avoiding the management of routing tables but resort to the so-called table of neighbors (NT) from which a next-hop is selected. However, NTs also require updating. For this purpose, some solutions resort to broadcasting beacons. We propose a Context- and Mobility-Aware Forwarding (CMAF) strategy that resorts to a Short-Term Mobility Prediction—STMP—algorithm, for keeping the NT updated. CMAF is based in Named Data Networking (NDN) and works in two modes, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). V2V CMAF leverages the overheard packets to extract mobility information used to manage NT and feed the STMP algorithm. V2I CMAF also uses a controlled and less frequent beaconing, initially from the Road-Side Units (RSUs), for a further refinement of the predictions from STMP. Results from extensive simulations show that CMAF presents superior performance when compared to the state of the art. In both modes, V2V and V2I (with one beacon broadcast every 10 s) present 5–10% higher Interest Satisfaction Ratio (ISR) than those of CCLF for the same overhead, at a cost of 1 s of increased Interest Satisfaction Delay (ISD). Moreover, the number of retransmissions of CMAF is also comparatively low for relatively the same number of hops. Compared to VNDN and Multicast, CMAF presents fewer retransmissions and 10% to 45% higher ISR with an increased overhead of about 20%.

Keywords: bloom filters; content delivery; forwarding; mobility prediction; Named Data Networking (NDN); Vehicular Ad hoc Network (VANET)

check for updates

Citation: Silva, E.T.d.; Macedo, J.; Costa, A. CMAF: Context and Mobility-Aware Forwarding Model for V-NDN. *Electronics* **2024**, *13*, 2394. https://doi.org/10.3390/electronics13122394

Academic Editors: Nikolay Hinov, Ognyan Nakov and Milena Lazarova

Received: 8 May 2024
Revised: 14 June 2024
Accepted: 16 June 2024
Published: 19 June 2024

1. Introduction

Intelligent Transportation Systems (ITS) [1,2], including other intelligent urban technologies, are the near future realities. Vehicular Ad hoc Networks (VANETs), or, more broadly, Vehicular Named Data Networking (V-NDN), have been seen as a potential network for the realization of the aforementioned technologies. The current development of these technologies, in terms of network communication, is based on the TCP/IP protocol stack, which was designed as a host-centered architecture. Current network applications are content-centric, and impose high demands in terms of network speed and bandwidth. Deploying these applications in a highly mobile network characterized by ephemeral and highly intermittent links between communicating nodes exacerbates the problems presented by a host-centric architecture. Aiming at solving the aforementioned problems, several Information-Centric Networking (ICN) based network architectures are currently under development. Named Data Networking (NDN), a specific realization of ICN, has been seen as a potential architecture to overcome some of the issues presented by the IP protocol stack (i.e., related to supporting mobility, routing and forwarding, network security, in-network caching, and application deployment).

Refs. [3,4] consider it infeasible to run a routing protocol in VANET. Having that in mind, several studies including the ones presented in Section 2 take advantage of the fact that the data plane in NDN is stateful, and propose forwarding strategies that are based on flooding instead of resorting to routing tables. Uncontrolled flooding may result in an undesirable problem, the broadcast storm. As it is described later, several mechanisms are being proposed to avoid the broadcast storm problem. One of the known mechanisms is to select a specific node from an existing neighbor list and unicast the packets to it. Only the selected node forwards the packets further. The management of the list of neighbors is another challenging issue. Several schemes resort to a beacon broadcast (a drawback) to discover new neighbors and for updating the list. This work proposes a Context- and Mobility-Aware Forwarding (CMAF) model, which, instead of broadcasting a specific beacon for neighbor discovering, leverages the overheard packets to extract mobility information that is used to manage the list of neighbors and feed a mobility prediction algorithm. CMAF is designed to reduce the transmission overhead while keeping the performance high. It works in two modes, a Vehicle-to-Vehicle mode, which does not send any beacon, and the Vehicle-to-Infrastructure mode, used for comparison, in which a controlled beacon is broadcast by the existing Road-Side Unit (RSU) in an urban environment.

The remainder of this section presents an overview of the technologies addressed by the present work, i.e., VANET and NDN. It includes a problem statement, which guides the development of the proposed solution.

1.1. Overview of Vehicular Ad hoc Networks (VANET)

VANET is a particular case of MANET, presenting specific characteristics that make it more challenging to develop. Other characteristics, however, are the advantages of VANET over the Mobile Ad hoc NETwork (MANET). For example, different from MANET, the mobility in VANET is predictable and constrained by the road layout and topology. Additionally, power supply, computational power, and storage are comparatively unrestrained requisites. Key differences of both can be found in works such as [5], and a summary of the main technical challenges that form the main areas for research in the VANET can be found in studies such as [6–9].

The VANET architecture has two types of nodes: (1) vehicles (equipped with On-Board Units—OBUs) and (2) the RSU. The RSU is a static infrastructure and has both communication interfaces, wired and wireless. It is mainly used to provide communications between vehicles and Internet Service Providers (ISPs) for Internet access [5]. The OBU is an in-car network device through which V2V or V2I communication takes place. Communications are usually grouped into the following network scenarios: (1) Pure ad hoc or V2V, an opportunistic and infrastructureless communication model, is effective for exchanging emergency content among vehicles. (2) V2I—vehicles take advantage of the existing static infrastructure (e.g., RSU or Base Station—BS) to communicate. The static infrastructures (i.e., RSU) are used not only for routing but also as cellular gateways and WLAN Access Points (APs) to provide Internet. (3) Vehicle-to-Everything (V2X) is a hybrid of V2V and V2I [10,11].

VANET is a potential network for the development of near future technology realities, such as ITS, including other intelligent urban technologies, e.g., autonomous vehicles, which are in high demand in this current reality, aiming at making daily life more secure and convenient [12]. As aforementioned, the current development of these technologies, in terms of network communication, is based on the 50-year-old host-centric TCP/IP protocol stack.

Current VANET applications, i.e., for safety and traffic information along with comfort and entertainment information, which is the main content that vehicles share in their interactions, are shifting their architecture to focus on the content (i.e., content-centric) and not on the host [13]. These applications impose high demands in terms of network speed and bandwidth for real-time content (e.g., high-resolution video streaming). When these applications are deployed in a highly mobile network characterized by ephemeral and highly intermittent links between communicating nodes, the characteristics of the aforementioned applications exacerbate the problems presented by a host-centric network architecture, such as the one based on the TCP/IP network stack. To overcome some of the issues of the TCP/IP protocol stack, emerging architectures based on the Information-Centric Networking (ICN) approach are being developed.

1.2. Information-Centric Networking (ICN)

ICN is an emerging network approach that defines a name as the central entity for forwarding, routing, and content identification. The ICN approach is designed to solve some of the issues still present in the TCP/IP-based networks. Some of the main characteristics that differentiate both architectures include the following: (1) In the ICN-based architectures, security is provided by design, not by extension, as in the TCP/IP-based networks. (2) In ICN-based architectures, communications are sessionless and natively support asynchronous data exchange between nodes. This is different from the TCP/IP networks, where communication between two nodes requires a pre-established session that must prevail during the content exchange period. (3) Interest aggregation in ICN-based architectures provides a native support for content multicast. (4) Differently from TCP/IP-based networks, in ICN-based architectures, content is self-consistent and independent from its location, providing the ability for in-network caching and sharing with whichever node that requests the content. Several ICN-based network architectures are under intensive development; among those, Named Data Networking (NDN) [14,15] has been under intense development and is seen as the most promising project [15,16].

Being a specific implementation of ICN, NDN architecture is also based on names for routing and content retrieval. The Interest and Data packets are the two messages on which the communications are based. NDN is a receiver-driven architecture, in which when a node (consumer) desires a specific content and sends an Interest packet that is forwarded through intermediate nodes towards the content holder. The content holder is either a producer or any other intermediate node that has previously received and cached the content, in its Content Store (CS). The Data packet is returned following the same routes the Interest packet used to reach the provider (i.e., breadcrumbs forwarding). NDN operates based on three main data structures: (1) FIB is for storing routing information. Routes stored in FIB are used to forward Interests towards content holders. The FIB population can be performed manually or by means of a name-prefix-based routing protocol. (2) PIT is for storing information about pending Interests which are on their way towards the content providers/holders. When an Interest is sent out for content retrieval, its identification and other related information is stored in PIT. When the corresponding Data packet is received, PIT is used to forward it to the node or nodes that requested the content. After this procedure, the corresponding PIT entry is removed. (3) In NDN, the content is decoupled from its producer/provider. This propriety provides the ability to ubiquitously store/cache and share contents with any node that should request it, and this way contribute to decreasing end-to-end delay and network load [17]. CS is the structure used for caching.

1.3. Realization of NDN-Based VANET

Several solutions for realizing a NDN-based VANET have been proposed. Studies such as those presented in [4,18–22], aimed at leveraging the characteristics of ICN/NDN for VANET. The study in [18] proposed a NDN-based inter-vehicular communication architecture aimed at improving content naming (supporting pull- and push-based communications), content addressing, Interest aggregation, and mobility in VANET. Arguing that the already proposed Content-Centric Network (CCN) [14] extensions for VANETs still present issues related to scalability and mobility, ref. [19] proposed a framework aiming at providing content-centric-based layer-3 and layer-4 operations over the IEEE 802.11p layers. The proposal leverages the CCN principles and takes further measures in order to better manage the node mobility and the broadcast-based wireless channel. The work in [20] highlighted the limitation of the NDN request–response communication pattern for the VANET event-based approach (e.g., the dissemination of traffic conditions for the better management of situations such as accidents) and augmented the NDN architecture capabilities with Publish/Subscribe extensions. Similarly, ref. [21] enhanced CCN with extensions in order to tackle the ad hoc propriety of VANET. Precisely, a special packet "event packet", which is a push-based message aimed at disseminating emergency information, is added to the architecture, and FIB is modified in order to be able to select Faces depending on the requirements of the applications using them. The study in [4] proposed a prototype implementing a NDN-based VANET, which caches all received data and uses geo-locations for content dissemination, with support for Delay-Tolerant Networks (DTNs). The authors in [22] proposed a hierarchical, cellular, and cluster-based architecture for NDN-based VANET, where nodes dubbed "barycenters" take the responsibility for communicating with a cellular layer in each cluster. For the interested reader, several other studies can be found in surveys such as [23–28], which include research challenges on implementing NDN-based VANET.

There are several proposed NDN-based VANET schemes for neighbor discovery and content dissemination. Section 2 presents some of these studies. One of the main drawbacks of those proposals is the broadcasting of a beacon message for neighbor discovery and content dissemination [6]. Broadcasting a specific packet into the network increases the traffic and can lead to network congestion. We propose CMAF, a protocol that, instead of broadcasting a beacon, leverages the overheard packets in the wireless channel to extract mobility information, later used to manage the list of neighbor nodes and to feed the mobility prediction algorithm. CMAF is designed to reduce the need for broadcasting a beacon, thus reducing the otherwise increased traffic in the network. The protocol is also designed to be context-aware, adapting itself according to the type of the received content and specific VANET environment (i.e., highway and urban), communication model (i.e., V2V and V2I), and network traffic (i.e., dense and sparse).

1.4. Main Contributions

This work proposes a context and mobility-aware forwarding model for VANET based on NDN, and is based on an initial proposal in [29]. It updates the logic presented in the aforementioned proposal and presents the results of simulations. The aforementioned work proposes a combo model providing a forwarding and a routing protocol. The solution presented in this work only deals with forwarding and does not manage FIB. Simulation and presentation of the results of the complete solution, including FIB management, which will also be based on a Long-Term Mobility Prediction (LTMP) algorithm for updating its entries, is planned for future work.

In addition to adopting NDN, the widespread use of localization systems (e.g., GPS, which we assume to be installed in all vehicles) has been seen as an important mechanism for improving routing in VANET. The use of localization systems for tracking the geographical position and trajectory of the vehicles, has opened opportunities for the development of various mechanisms for predicting the mobility of vehicles on the road. Resorting to these mechanisms, a designed forwarding model may be able to more efficiently track the node mobility and better select a relay node with the highest probability of forwarding a content to the destination. We assume a GPS always available and functional, although we are aware that indoors and extreme weather may affect it.

In this work, mobility awareness refers to the ability to gather information about the mobility of the vehicles and use the collected information to help in choosing the better node or path for forwarding. The vehicle's mobility information is appended to each packet and can be extracted from it by any node receiving the packet. The context awareness may include information such as current node speed and geographical location; time when the content is generated and its type (i.e., safety and non-safety); application to which the content belongs; content format; application popularity; and neighborhood conditions [30]. Context awareness in this work refers to the general network environment awareness. That is, the awareness of the type of application (i.e., safety, transport efficiency and information/entertainment) [7,31] communication model (i.e., V2V, V2I, and V2X as presented in Section 1.1) and network scenario (i.e., highway and urban environment).

In summary, the contributions are as follows:

- A novel forwarding scheme based on overheard packets, instead of broadcasting a beacon, for neighbor discovery is proposed. This scheme also uses mobility prediction for short-term updating of the neighbor list.
- A short-term mobility prediction algorithm based on the Extended Kalman Filter (EKF) is provided.
- A mechanism based on a Bloom Filter (BF) for content sharing is provided. For new content discovery, the RSU sends a periodic request for which the producer responds with a special Data packet, announcing its new content and attaching, by means of the BF, the contents stored in its CS.
- We provide and discuss the results of extensive simulations performed to assess the effectiveness and performance of the proposed model.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 presents the model design, which includes the presentation of the main modifications of the NDN structures, the proposed mechanism for neighbor and content discovery, the processing of the incoming Interest and Data, and finally the mobility prediction mechanism. Section 4 presents the simulation results and discussion. Section 6 presents the summary and final remarks, and Section 7 presents the future work for further improving the model.

2. Related Work

There are several solutions for content dissemination in NDN-based VANET. This work does not pretend to perform an exhaustive survey on the state of the art of forwarding schemes; existing surveys on this matter include [6,17,28,32] and the ones mentioned in Section 1.3.

Given the instability of routes in VANET, which is a direct result of the high mobility of nodes, network partitioning and short-lived connectivity, several studies (e.g., [4,13,24,33]) argue that maintaining an updated FIB is computationally costly and not feasible, and that a flooding-based approach is the one that better suits VANET. Flooding performs well and is very robust in finding routes to the content providers on sparse networks [34,35]; however, in a dense network, it brings a perverse problem that demands correction—the broadcast storm. In contention-based media, such as the case of wireless channels, flooding leads to a lot of conflicts in accessing the channel. The conflict results in packet loss, which in turn requires packet retransmissions. Packet retransmissions increase network traffic and worsen the channel's capability of disseminating packets, which results in deteriorated network performance. Several solutions have been proposed for restricting uncontrolled flooding. These solutions include (1) choosing a specific node, among others, which will forward the packet further; or (2) providing enough mobility information to the receiver node that it can use for deciding whether or not to forward the received packet further. For each of the aforementioned approaches, several other specific distinctions for better reducing the broadcast storm are proposed. For this, several parameters (e.g., geo-location, link stability or LET, and the distance to the provider or from the previous 1-hop sender) are usually employed. We highlight that this work strictly considers forwarding strategies as the solutions that do not directly manage FIB for routing/forwarding decisions. Studies that include FIB management for their forwarding decisions are considered to be routing protocols. This distinction is important and is extensively discussed in [28].

Based on the previously mentioned studies, including [17,36], and taking into consideration the node that is responsible for forwarding the Interest, we categorize the strategies into two categories: (1) sender-oriented approaches, and (2) receiver-oriented approaches.

2.1. Sender-Oriented Approaches—Neighborhood-Aware

In this category, the node sending the Interest upstream is responsible for selecting the next-hop node that will forward the Interest further. Protocols in this class are also dubbed neighbor-aware solutions, that is, in order to be able to select a better relay for forwarding the Interest, nodes need to keep a list of surrounding nodes (i.e., neighbors) from which the next forwarder is chosen.

State-of-the-art protocols in this category [36–41] resort to beacon broadcasting for discovering new neighbors and updating the positions of the already existing ones. Broadcasting an additional control packet into the network results in an increased transmission overhead. The frequency of the beacon broadcast should be chosen wisely, taking into consideration that, although broadcasting with high frequency is better for updating the list of neighbors, it also results in increased traffic, which may result in a broadcast storm, and consequently deteriorated performance. This problem is exacerbated in dense network scenarios. Conversely, sending less frequent beacons is better for keeping the traffic low and the channel reserved for the actual content dissemination. This, however, results in an outdated list of neighbors, which in turn affects negatively the efficiency of the protocol. Therefore, a compromise between these two factors (i.e., transmission overhead and the timely updated list of neighbors) should drive the selection of the frequency for beacon broadcasting. Instead of using beaconing for neighbor discovery, the proposal in [42] uses a pair of digital cameras for collecting the visual identification of vehicles in the range (line of sight) of the cameras. The need for a line of sight, and the complexity of processing the visual data (i.e., license plate, color, brand, and car type) and accurately distinguishing the neighbors may be challenging, mainly in a dense network. Another alternative to beacon broadcasting is the overhearing of the packets in the wireless channel and using them to build and update the list of neighbors. Moreover, as proposed in [28], incorporating packet overhearing and mobility prediction for updating the list of neighbors may avoid the need for beacon broadcasting and help to keep the transmission overhead low.

Protocols in this category are unicast by nature; as previously explained, they forward a packet to a specific selected node. In order to be able to identify a specific node, nodes should be unambiguously identifiable; therefore, usually, additional fields (i.e., MAC address of the node's Face) are added to the packets. Furthermore, the mobility information of the nodes also need to be added to the packets, either by including new fields in the Interest–Data packet or by piggybacking the information into NDN Link Protocol's headers. In sender-oriented approaches, the next-hop forwarder is selected considering parameters such as (a) their distance (mostly the Euclidean distance) from the 1-hop previous node; or (b) their distance (Euclidean distance or the number of hops) to the content provider, if its location is known. The following are the representative state-of-the-art solutions into this category.

In the Geographical Opportunistic Forwarding Protocol (GOFP) [37], a next-hop forwarder is chosen if it can satisfy the Interest itself (being a producer or holding the desired content in its CS), or if it can be the earliest to reach the Position-of-Interest (POI) where the desired content can be fetched. So if the node does not hold the desired content in its CS, the minimum distance to the POI is the parameter chosen for selecting the next-hop forwarder. If no vehicle meets the two previously mentioned conditions (i.e., Data in CS and existing node for calculating distance to POI, or the current node with minimum distance to POI compared to possible forwarders), the node holding the content does not forward it, and keeps it until a suitable node is located. For selecting next-hop forwarder for the Data, the nearest distance between the trajectories of the candidate next-hop and consumer before the Data's freshness expires is considered.

The Mobility-Predict-based Forwarding Strategy (MPFS) [38] uses two parameters for selecting the next-hop forwarder: (a) the Link Expired Time (LET), and (b) the distance along the road. The node with the highest LET and simultaneously farthest distance along the road, within the transmission range of the sender, is selected as the next-hop. The solution presented in the aforementioned study only consider highways, and two next-hop forwarders are selected, one for each direction. Moreover, besides the mobility prediction, the solution still resorts to broadcasting beacon (every 2 s) for updating the Neighborhood Table (NT). A similar solution, Predictive Forwarding Strategy (PRFS) [36], is presented by the same authors.

Similarly, two next-hops forwarders are selected in the Packet Forwarding strategy based on Optimal and Backup (PFOB) [39], one for each direction of the highway. This protocol uses distance, relative speed, link duration, signal strength, and node degree for selecting the next-hops. During the process of selecting a next-hop to forward the Interest upstream, simultaneously, a backup node is chosen for helping to forward the Data downstream. The chosen backup node is the one with the highest link stability. Beaconing for updating NT may increase the network overhead.

In RobUst Forwarder Selection (RUFS) [40], nodes exchange, via beacon broadcast, a so-called recent satisfied list which includes satisfied Interests for specific CN, number of hops for the satisfied Interests, and the Interest Satisfaction Ratio (ISR). The received information is used for updating a local list called the Neighbor Satisfaction List (NSL). A node from NSL that simultaneously presents the highest ISR and lowest number of hops to the content source, and that recently satisfied the requested content is selected as the next-hop forwarder. Beaconing for updating NSL may increase the network overhead.

As highlighted before in this section, the maintenance of the NT is a crucial factor for the efficiency of the protocols presented in this section. The NT is invaluable if not properly updated. Using an outdated NT for selecting the next-hop forwarder can result in an erroneous selection of a next-hop (i.e., the node may already be out of the transmission range of the sender), leading to the loss of the forwarded packet.

2.2. Receiver-Oriented Approaches

In this category, instead of selecting the next-hop forwarder, the sender broadcasts the Interest and leave the responsibility of deciding whether or not to forward the Interest further to the nodes that received the packet. Nodes calculate their relative forwarding priority, and nodes with higher priority retransmit the received interest. Nodes with lower priority only retransmit the received Interest if they do not overhear a retransmission of the same packet. Protocols in this category are basically timer-based forwarding approaches. When a node receives an Interest, it sets up a deferring timer and awaits to overhear a retransmission of the same Interest from other nodes. When the node overhears the same Interest retransmitted by other nodes, it cancels its own waiting timer. Nodes with higher priorities have a shorter waiting timer. Nodes calculate their priorities taking into consideration parameters such as (a) their distance (mostly the Euclidean distance) from the 1-hop previous node, and their distance (Euclidean distance or the number of hops) to the content provider if its location is known; (b) link stability; (c) Interest satisfaction rate; and (d) their geo-location and other centrality measures. Some of the state-of-the-art solutions in this category are presented next. Some of the protocols presented next calculate node priority based on more than one of the aforementioned criteria.

2.2.1. Position- and Distance-Based Approaches

Strategies in this category [4,43–55] select the farthest nodes from the 1-hop previous sender as the next-hop forwarders. The distance to the content provider, if available, is also used to further refine the next-hop selection.

Aiming at tackling the broadcast storm and network partitioning problem, Rapid Traffic Information Dissemination (RTID) [43] relies on four timers (i.e., collision avoidance timer, pushing timer, NDN-layer retransmission timer, and application retransmission timer) to achieve the aforementioned objective. The "collision avoidance timer" is used for delaying transmission among neighbors and, thus, avoid collisions. The "pushing timer" is used for defining a next-hop forwarder, which must be the farthest one from the actual sender. The "NDN-layer retransmission timer" is used for re-broadcasting and reception acknowledgment, and the "application retransmission timer" is used for Interest re-expressing from the applications. VANET via Named Data Networking (V-NDN) [4] uses only one timer for mitigating the broadcast storm. The waiting timer calculation is based on the locations of the 1-hop previous nodes and Data sources. A farthest neighbor from the previous 1-hop node that is concomitantly the closest node to the content provider has the highest priority for forwarding; thus, its waiting timer is the shortest. The proposed solution spreads the Interest in all directions. The Context-Aware Vehicular NDN (CA-VNDN) [44] uses the same idea but restricts the Interest forwarding to a specific area. Furthermore, ref. [44] includes an enhancement of Data breadcrumbs forwarding by unicasting the Data to its provider if its geo-location is present in PIT. Another proposal similar to [4] is Density-Aware Delay-Tolerant (DADT) [45]. The proposed solution adds the capability of supporting network partitioning by storing every Interest packet in a so-called pending retransmission queue, from which the Interest is only retransmitted when a capable neighbor can be selected for transmission. The protocol broadcasts a beacon every second, which may increase the transmission overhead. Additionally, the proposal requires the node to know the producer's geo-location, which may not be provided. Two timers (i.e., Forwarding-Priority timer and Collision-Avoidance timer) are used in their other work, dubbed Location-Based Deferred Broadcast (LBDB) [46]. The "Forwarding-Priority timer" is used to determine the priority and selection of the next-hop, and the "Collision-Avoidance timer" is used for preventing collision. Also, based on the distance from the 1-hop previous node and closeness to the content provider, Content Connectivity and Location-Aware Forwarding (CCLF) [47] also resort to a queue for storing Interests, which are then retransmitted when a neighbor is sensed. It sends periodic beacons for maintaining the NT when no packets are sensed for some time. A particular characteristic of this proposal that is different from [45] is that the neighborhood awareness is also used for suppressing Interest–Data, and thus efficiently manages the node's resources. Nodes in Opportunistic Interest Forwarding Protocol (OIFP) [48] only consider the distance between themselves and the 1-hop previous node to compute the priority and auto-select themselves as the next-hop forwarders. The closest node to the boundary of transmission range sets itself as the priority forwarder by setting its waiting timer lower. This proposal differentiates itself from the former by sending the Interest to different directions. In [48], nodes send the Interest even when there are no neighbors around, which results in an inefficient usage of the node's resources. In [54], prior to sending the content request, a beacon broadcast, for provider discovery, takes place. Only when the location of the content provider is known is the contention timer set and the content requested.

Besides the distance from the 1-hop previous node, the Content Discovery Protocol (CDP) [49] also uses the so-called sweet spot (i.e., an area where the vehicles are best positioned to forward packets to a higher number of neighbors) to further refine the process of prioritizing the next-hop forwarder selection. Thus, a node receiving an Interest sets its priority high, and consequently the waiting time for retransmission is lower if its distance from the previous node is relatively higher, considering the transmission range of the sender, and the node itself is within the sweet spot. The Interest–Data flow-tracking-based forwarding scheme (IDTracS) [50] uses the distance from the 1-hop previous node alongside the degree of centrality (i.e., counted Interest and Data packets) to decide which node should forward the received Interest. Differently from the study in [40], which uses ISR as one of the metrics for selecting the next-hop, ref. [50] uses the counted Interest and Data packets independently. The farthest nodes from the previous sender that simultaneously present a higher degree of centrality have higher priority and thus a lower waiting timer. The same authors propose in [51] a protocol that creates suppression areas for controlling and suppressing duplicated Interest forwarding. In the defined area, timers are used to prioritize forwarding. The road topology is used to specify the areas which are created in different directions, leading to a forwarding strategy that forwards Interests to each road direction. Context-Aware Content-Naming (CACN) [52] and Decentralized Receiver-based Link Stability-aware Forwarding (DRLSF) [55] also propose the creation of forwarding zones, which are based on the angular information of the nodes on the road. It catalogs the created zones (i.e., red and white zones) regarding their capacity for forwarding packets with fewer unnecessary retransmissions.

2.2.2. Link-Stability-Aware Approaches

Strategies in this category [53,56–59] select nodes with relatively more stable links as the next-hop forwarders. The idea is to ensure that the returning path for the Data packet is still available when the content is sent back downstream.

In Distributed Interest Forwarder Selection (DIFS) [56], two forwarders are selected as next-hops, for each of the two highway directions. The next-hop forwarder auto selects itself among neighbors if it cumulatively has a long link connectivity duration, it is the farthest node, and it has a relatively lower speed compared to the previous 1-hop node. Two lists (i.e., neighbor list and Decision List) are used in each node for deciding on the forwarding eligibility. The DL holds the status information of all neighbors and each node uses it for ranking itself among them. Beacons are used to share the mobility information among neighbors, which may increase the network overhead. Moreover, the information in DL may become outdated and affect the eligibility decision and consequently the efficiency of the protocol. In Link Stability-Based Interest Forwarding (LSIF) [57], when a node receives an Interest, it extracts the mobility information from the received Interest and estimates the link lifetime, i.e., the time the two nodes will be in their mutual communication range. It periodically predicts the node mobility and estimates the remaining lifetime of the link. If the estimated lifetime is above a defined threshold, the receiving node forwards the packet; otherwise, it drops it. The original sender node always sends the packet regardless of not having any node in its vicinity. Similarly, aiming at reducing redundant transmissions of Interest, Link Stability-Based Interest Forwarding for Content request (LISIC) [58] creates stable paths for Data delivery by selecting more stable links for forwarding Interests. A waiting timer is computed based on the estimated duration of the link lifetime. Furthermore, ref. [58] resorted to hop count for further limiting the propagation of Interests. The same authors proposed in Location-based and Information-Centric (LoICen) [59], a protocol that resort to the opportunistic sharing of CS content catalog among vehicles. A so-called Content Location Table (CLT) is used to store the location of nodes previously encountered. Interests are then forwarded towards the location where the vehicles holding the content are located. The aim of this proposal is also the avoidance of broadcast storm by redirecting the content for specific locations where a possible holder is located. Outdated content location may lead to packet loss and retransmissions. Without a possibility of updating or predicting the mobility of the nodes registered in CLT, information in this table may rapidly become outdated. The proposal in E-Fuzzy [53] includes also the distance between nodes and the signal strength, and resorts to fuzzy logic with the aforesaid metrics to identify the best nodes for forwarding.

2.2.3. Geo-Location-Aware Approaches

Protocols in this class [43,60] forward Interest to a specific region. Generally, the requested content is location dependent and only makes sense for that specific location. Any node, even outside the region in which the content is desirable, can forward the content should it possess the requested content in its CS. For location-independent content, the protocol is required to use additional measures to tackle the producer's mobility problem.

In GeoZone [60], Interests are forwarded using a geo-referenced naming scheme to request content from a desired POI or dissemination zone. When an Interest is issued, only nodes within the POI can flood it, and the flooding is limited to that given zone. A node receiving an Interest decides whether or not to forward it further based on its own geo-position. If within the dissemination zone, it can forward the Interest within the zone. For collision avoidance, the protocol adopts and uses three of the four timers proposed in [43].

2.2.4. Interest Satisfaction Rate-Based Approaches

In Neighborhood-Aware Interest Forwarding (NAIF) [61], the next-hop forwarder auto-selects itself considering its Data retrieval rate (i.e., ISR) and a forwarding rate (i.e., fraction of incoming Interests the node will forward) for a specific CN, and the distance to the consumer. Nodes in the proposed protocol can reduce the overhead by sharing the workload among their neighbors. Forwarding statistics are updated by periodically probing the 1-hop neighbors. The probing process introduces a new packet into the network and may result in an increased transmission overhead. As mentioned before, differently from this study, the work by [50] uses the counted Interest and Data packets independently, instead of computing the ISR.

2.2.5. Comparison of the Selected State-of-the-Art Solutions

The following section summarizes and compares the related work. Table 1 shows in column "Forwarding", the alternative solutions as presented in the previous sections. The options are sender-oriented, which are unicast-based solutions, and receiver-oriented, which are mainly broadcast-based forwarding schemes. In terms of new packets, the protocols that include new packets resort only to the beacon broadcasting. Our proposal is designed to work without resorting to beacons. However, it alternatively processes beacons from the RSU whenever they are received. It means that the solution adapts itself for working with beacons should they be received. As shown in Section 4.3, the alternative without processing beacons presents better performance. The "New struct." field indicates an additional structure besides the common ones from the native NDN (i.e., CS, FIB and PIT). The "New packets" indicate whether or not additional packets, besides Interest and Data, are used by the protocol. Generally, a beacon or additional Interest carrying mobility information or a possible catalog of cached content is used for this matter. CMAF have two distinct operational modes, one for V2V and another for V2I. In V2I mode, a beacon is broadcast by the RSU or vehicles; therefore, the field "New packets" should be considered differently for each operational mode. The "Neighbor" field indicates that the protocol manages a NT and only sends Interests when the lists are not empty—we highlight that some protocols do not regularly update their neighborhood list. The "Other" field indicates the use of other criteria, such as relative speed for calculating, for instance, which vehicle will reach the PoI first (e.g., study by [37,39]), node degree, or signal strength (e.g., in [39]), and suppression angle (e.g., in [44]).

Our proposal chooses the next-hop forwarder somehow similar to the proposal by [45]; the sender chooses from the NT a node that is (a) farthest from the previous 1-hop node, and (b) that is simultaneously the closest to the content provider, whenever this information is known. Moreover, the model uses hop count to further limit the Data packet dissemination. Differently from [62], the hop count is fixed based on the average number of neighbors for a previously predetermined time as presented in Section 3.4.2.

Table 1. Summary and comparison of representative state-of-the-art solutions.

Study	Forwarding	Criteria for Next-Hop Selection (Awareness)						Mob. Pred.	Scenario	New Packets	New Struct.	Comm.		Main Challenges/Limitations
	(-Oriented)	Dist.	Link -Stab.	Geo-Locat.	Packets (Rate)	Neighbor	Other					V2V	V2I	
V-NDN [4]	Receiver	–	–	✓	–	–	–	✓	Urban	–	–	✓	✓	Fixed HopLimit. Only location-dependent content.
PRFS [36]	Sender	✓	✓	–	–	✓	–	✓	Highway	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery. Geo-location provided but content source's location awareness not leveraged . NT is updated reactively and the listed neighbors in the NT may already be out of the transmission range of the sender.
GOFP [37]	Sender	✓	–	–	–	–	–	✓	Urban	✓	–	✓	–	Additional overhead from the opportunistic CS content digest announcements. Does not consider producer mobility.
MPFS [38]	Sender	✓	✓	–	–	✓	–	✓	Highway	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery. Mobility prediction only considers linear trajectory of vehicles.
PFOB [39]	Sender	✓	✓	–	–	–	✓	✓	Highway	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery. Geo-location provided but content source's location awareness not leveraged.
RUFS [40]	Sender	✓	–	–	✓	✓	✓	–	Urban	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery. Geo-location provided but content source's location awareness not leveraged.
RTID [43]	Receiver	✓	–	–	–	–	–	–	Highway	–	–	✓	–	Vehicles with a constant speed.
CA-VNDN [44]	Receiver	✓	–	–	–	–	✓	✓	Urban	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery.
DADT [45]	Receiver	✓	–	–	–	–	✓	–	Urban	✓	✓	✓	✓	Additional overhead from beaconing for neighbor discovery.
LBDB [46]	Receiver	✓	–	–	–	–	–	✓	Urban	–	✓	✓	✓	Producer mobility not considered.
CCLF [47]	Receiver	✓	–	–	✓	✓	–	–	Urban	✓	✓	✓	–	Additional overhead from beaconing for neighbor discovery.
OIFP [48]	Receiver	✓	–	–	–	–	–	–	Urban	–	–	✓	–	Geo-location provided but content source's location awareness not leveraged.
CDP [49]	Receiver	✓	–	–	–	–	–	✓	Urban	–	–	✓	–	Geo-location provided but content source's location awareness not leveraged.
IDTracS [50]	Receiver	✓	–	–	✓	–	–	✓	Urban	–	–	✓	–	Geo-location provided but content source's location awareness not leveraged.
GeoISA [51]	Receiver	✓	–	–	–	–	–	✓	Urban	–	–	✓	–	Requires knowledge of different topological road structure. Geo-location provided but content source's location awareness not leveraged.
CACN [52]	Receiver	✓	–	✓	–	–	–	✓	Highway	–	–	✓	✓	Considers an immovable producer. Only disseminates notification messages.
DIFS [56]	Receiver	✓	✓	–	–	✓	✓	✓	Highway	✓	✓	✓	–	Beaconing for sharing mobility information may increase overhead.
LSIF [57]	Receiver	✓	–	–	✓	–	–	–	Urban/Highway	–	✓	✓	–	Assumes nodes following a linear trajectory.
LISIC [58]	Receiver	–	✓	–	–	–	–	✓	Urban	–	✓	✓	–	A node with similar mobility pattern (i.e., velocity and direction) as the sender may not always be the better next-hop, and it may be better to keep the Interest with the current node. Low network density not considered.
GeoZone [60]	Receiver	–	–	✓	–	–	–	–	Urban	–	–	✓	✓	Geo-location of the content must be known.
NAIF [61]	Receiver	✓	–	–	✓	–	✓	–	Urban	✓	✓	✓	–	Additional overhead from beaconing for collecting forwarding statistics.
LoICen [59]	Receiver	–	✓	–	–	–	✓	–	Urban	–	✓	✓	–	Mobility of vehicles holding content in their CS is not considered. Outdated content location may lead to packet loss, and require retransmissions. Low network density not considered.
DRLSF [55]	Receiver	✓	✓	–	–	–	–	–	Urban/Highway	–	–	✓	–	
VIFVNDN [42]	Sender	–	–	–	–	✓	–	–	Urban	–	–	✓	–	Assumes the node transmission range equal to the range of cameras in order to ensure the reachability of the line of sight.
EGBIF [41]	Sender	–	–	–	–	✓	–	–	Urban	–	–	–	✓	NT is not updated.
E-Fuzzy [53]	Receiver	✓	✓	–	–	–	✓	–	Highway	–	–	✓	–	
DBVNDN [54]	Receiver	✓	–	–	–	–	–	–	Urban	–	✓	✓	–	
CMAF	Sender	✓	–	–	–	–	–	✓	Urban/Highway	✓	✓	✓	✓	Computation cost for proactively predicting vehicle's mobility to update the NT.

3. Design of the Proposed Model—CMAF

This section presents the design of the proposed forwarding model. First, we propose a context- and mobility-aware forwarding model, designed to take mobility prediction into consideration and forward packets to specific nodes whose trajectories are known. The proposal is designed to avoid broadcast whenever possible. In another configuration, we explore the in-network caching by means of an RSU-initiated process of sharing the list of cached content. The sharing procedure is only performed by nodes in the forwarding path from the RSU to the producer.

3.1. Novelty of the Proposed Model

State-of-the-art solutions resort to broadcasting a beacon for neighbors discovery. CMAF works in two distinct modes: (1) The beaconless mode (based on the V2V communication model) does not issue any specific Interest (beacon) for neighbor discovering. Instead, it leverages the propriety of the wireless channel, where packets share the same channel and all can be sensed by the nodes in the communication range of the sender. To attain this, all packets carry Node Mobility Status Information (NMSI), Section 3.2, of its sender. This way, the protocol avoids increasing the network load with additional packets for this specific purpose. (2) In the beacon-based mode (based on V2I communication model), beacons are periodically broadcast for content discovery. In this mode, an existing RSU sends a periodic Interest from which the producer responds with a special Data (sData) packet carrying a catalog of the contents of its CS by means of a Bloom Filter (BF). All intermediate nodes get to know which cached content the producer has and which cached content the previous nodes from the producer to the RSU have (this procedure is explained in Section 3.3). Secondly, a mobility prediction scheme proactively updates the neighbor's information in the NT. The mobility prediction scheme includes a proactive agent that is responsible for keeping the NT consistent (i.e., purging the inactive and out-of-transmission-range neighbors). The aforementioned agent periodically checks the status of the nodes in the NT. The neighbor purging of the NT takes place if any of the following conditions takes place: (i) when its NMSI is updated and the computed distance to the current node falls out of the transmission range of this node; (ii) when a time threshold ($t_{NTpurge}$) elapses without the current node hearing another packet from the neighbor in the NT, the corresponding neighbor's status changes to inactive. When the agent finds a node with the status "inactive", it means that the neighbor can no longer be selected as a relay; therefore, it is purged from the NT.

It is worth highlighting that, including additional information (i.e., NMSI) on the packets headers increases its size. An elongated packet is more likely to obtain transmission errors or collisions given that it takes more time to be transmitted. However, we anticipate that the advantages of including this additional information outweigh the beaconing alternative.

3.2. Main Modifications of the NDN Structures

The NDN architecture is originally designed for wired networks, where neighbors of a given node are unambiguously identified by the Face they are connected (wired) to. Sending or receiving a given packet to or from a specific node is achieved by selecting the Face that the node is connected to. PIT is extensively used to support this model, where an entry is created for each upstream and non-duplicated incoming Interest, associated with its ingress Face. However, different from the wired channel which is unicast, the wireless channel is broadcast-based by nature, i.e., each node within the communication range of the sender overhears the packets in the channel. Thus, for wireless channels, the Face-based communication model is not feasible. A manner to unambiguously identify a node in wireless-based networks, for unicast and anycast communications, is crucial [63]. A good candidate for node identification is the MAC address of the Faces, i.e., net devices, installed on these nodes. Some representative studies which used MAC addresses to identify nodes include [64–67].

The dissemination model of the wireless channel can be explored to reduce the need for a purposely initiated broadcast for discovering neighbors, and to learn about new content sources. In order to take advantage of the overheard packets, all NDN packets are extended to carry additional (optional) control information—the NMSI, which includes (a) the ID of the sender node; (b) its current speed; (c) its geographical position; and (d) the current timestamp. The NMSI is included in each packet, but its use is optional in the sense that whenever necessary, the model can fall back to the normal operational mode of NDN, flooding the network to discover new neighbors and new content sources. NMSI is added as a tag onto the Named Data Networking Link Protocol (NDNLP) packet header instead of modifying the network layer packet header. When a packet is received, its corresponding NMSI refers to the status of the last node from where the packet was received.

When the requested content is received, a PIT lookup takes place in order to find the Face to forward the content to. As aforementioned, for a wired network, a Face unambiguously identifies the next-hop. The Face-based communication mechanism does not work well for wireless channel, given that all packets are forwarded out from the same Face. To overcome this issue and be able to identify the possible next-hop nodes for Data delivery on downstream, the ID of the node from which the Interest was received (*fromId*) is included in the corresponding PIT entry alongside the Face. As mentioned, a good candidate for node identification is the MAC address of the Faces (i.e., net devices) installed on these nodes. The node identified by *fromId* is the first choice for Data forwarding. However, as the Data packet includes the ID of the original requester (as explained in Section 3.4), the current node knows where the packet should be sent. Thus, besides the node identified by *fromId*, whenever an intermediate good candidate for relay exists, it can be selected to forward the content. In this case, the Data next-hop relay, which is included in the packet, is updated accordingly.

Three additional tables are included: (1) the Cached Content Table (CCT)—used in each node to catalog the cached Data of all nodes with which the node has interacted before; (2) Neighborhood RSU Table (NTRsu)—used to store the list of neighbor RSUs, and nodes in this list are persistently stored; and (3) Neighborhood Table (NT)—used to store the list of mobile neighbor nodes. In fact, all RSUs are also added to this table but, differently from the NTRsu, when a predefined time elapses without without any contact from the RSU, its ID is removed from the NT.

Each CCT entry is a tuple composed of the ID of the node from which a packed has been received, and the corresponding BF that codifies the catalog of the contents of its CS.

The insertion of a new neighbor and the maintenance of the NT is described in the next Section 3.3.

3.3. Overview of Cached Content and Neighbor Discovery Process

This section presents the content and neighbor discovery process. The specific forwarding process, for Data and Interest dissemination, is explained in Sections 3.4 and 3.5, respectively.

As aforementioned, the proposed protocol has two operational modes: a beaconless (V2V-based) and a beacon-based (V2I-based) model, for neighbor and cached content discovery.

3.3.1. Beaconless Mode

In this mode, no cached content discovery process takes place. The neighbor discovery leverages the broadcast-based wireless channel. The nodes build the NT based on packet overhearing on the air. The model is configured to operate in the promiscuous mode and process all received packets. When a packet (i.e., Interest or Data, including the unsolicited) is received, it follows the normal processing from NDN architecture and, additionally, the packet is sent to the NDN strategy layer, where the protocol processes it further. That is, all received packets are processed by the strategy layer.

First, the strategy layer extracts the NMSI from the received packet and builds/updates the NT. Two lists (NTs) are provided in each node, one for storing the list of mobile neighbors (i.e., NT) and another (i.e., NTRsu) for storing the static neighbors (i.e., RSU).

Static nodes in NTRsu are persistently preserved in each node. The reasoning behind this choice is that given that the nodes are static, and knowing their geo-position, each time a node is in the vicinity of the geo-position of the RSU, it can always redirect the requests directly to that RSU, leveraging the broader knowledge of the network possessed by the RSUs. The NT management is provided by Algorithm 1. The NT management procedure is based on and resorts to (Line 19) the Short-Term Mobility Prediction (STMP) algorithm, Section 3.6, for predicting the next node's geo-location.

Algorithm 1: InsertOrUpdate the NT

```
    Input   :NMSI
 1  n_geoPosition ← ExtractNeighborPosition (NMSI);
 2  o_geoPosition ← GetOwnPosition ();
 3  d ← CalculateDistanceToNeighbor (n_geoPosition, o_geoPosition);
 4  neighbor ← setNeighborStatus (status_ACTIVE);
 5  neighbor ← setNeighborDistanceToActualNode (d);
 6  if (New nighbor is a RSU) then
 7      insertNodeToNTRsu (neighbor) // Definitive insertion
 8      insertNodeToNT (neighbor) // Possibly removed later when no packets are
            overheard from the RSU
 9  else
10      if (Node exists in the NT) then
11          if (Timer still running) then
12              if (Timer has less than 100 ms to finish) then
13                  neighbor ← resetTimerForUpdatingNeighborInfo ();
14                  updateNT (neighbor);
15              else
                    // Discards Update ()
16                  return;
17          else
18              if (status_ACTIVE) then
19                  n_geoPosition ← PredictActualGeoPosition ();
20                  d ← CalculateDistanceToNeighbor (n_geoPosition, o_geoPosition);
21                  if (d less than transmission range) then
22                      neighbor ← updateNeighborDistanceToActualNode (d);
23                      neighbor ← resetTimerForUpdatingNeighborInfo ();
24                      updateNT (neighbor);
25                  else
26                      setNeighborStatus (status_INACTIVE);
27              else
                    // Awaits For Removal From NT ()
28                  return;
29      else
30          neighbor ← setTimerForUpdatingNeighborInfo ();
31          insertNodeToNT (neighbor);
```

Every $t_{neighborUpdate} = 600$ ms, the STMP algorithm predicts the neighbors' location and updates the corresponding node's information in the NT. Whenever the mobility prediction takes place, the neighbor's status is updated. The status is set to *inactive* when the predicted geo-position is out of the current node's transmission range and kept *active* when it is within the current node's transmission range. Moreover, every $t_{inactivePurge} = (3 \times t_{neighborUpdate} + 1)$ ms, the NT is checked, and all neighbors with status *inactive* are removed from NT.

The proposed STMP algorithm is locally linear. As explained in Section 3.6, assuming the time interval from the last packet received from a neighbor and the possible need to select this neighbor as a considerably short relay, it allows us to take the traveled distance to also be short, and subsequently take the mobility as partially linear. The $t_{neighborUpdate} = 600$ ms is chosen in order to predict the mobility every distance $d_{traveled} = 20$ m/s × 0.6 s = 12 m. That is, a prediction is made at most 24 m of the traveled distance with a maximum speed of 20 m/s, for two nodes in opposite directions. As it is around this distance (20 m) from a junction, that nodes begin to be incapable of receiving transmitted messages [68].

After the described process has occurred, the processing of the received packet continues as described in Section 3.4 for Data packets, and in Section 3.5 for Interest packets.

3.3.2. Beacon-Based Mode

In the urban scenario, the RSU sends, by flooding, periodic beacons for content discovery. The frequency of beacon sending is f_{RSUbf}. When a producer receives a beacon, it responds with the sData packet. The returned sData packet carries a BF that codifies the catalog of the contents stored in the node's CS. Each node receiving this packet extracts this information and forwards the packet further to the RSU. When a Data packet is returned, it carries its next-hop in the "destId" tag. So, besides the extraction of the BF, the node receiving the sData verifies if it has been registered/selected as the next-hop. Being a selected next-hop, the node sends an unicast request to the previous node, from which the Data have been received. The response to this unicast request is another Data packet (dubbed *sharingBF*) carrying a BF that codifies the list of this intermediate node's CS. All 1-hop away nodes receiving the *sharingBF* extract the BF and then drop the packet. This procedure is explained in Figure 1. For instance, considering that figure, the intermediate nodes I_0, I_2, I_3 and I_4 all forward the beacon received from the RSU, and receive back the sData packet, which is processed to extract the BF it carries. The referred BF is a codified catalog of the content that the producer P has in its CS. When node I_3 receives the BF, it sends a request for a *sharingBF* Data packet to the node I_4. The requesting Interest is received by all nodes (e.g., I_2 and I_4) in its transmission range. Only node I_4 responds to this request, as it is the final destination of it. When node I_4 responds with a *sharingBF*, all 1-hop nodes in its transmission range (e.g., I_3 and P) receive this packet and extract the catalog that codifies node I_4's CS. This way, all 1-hop nodes from I_4 get to know the contents of its CS. The *sharingBF* Data packet is never requested to the producer, as it has already shared the sData packet, which includes the content catalog of its CS, by means of a BF.

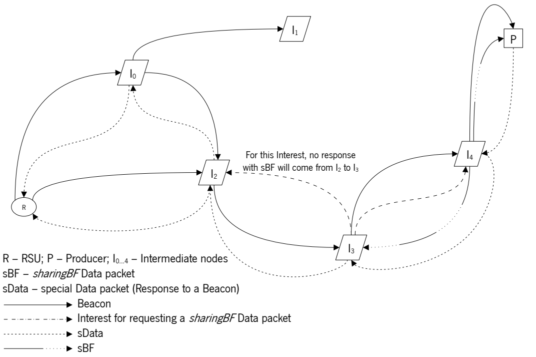

I1
P
I0
For this Interest, no response with sBF will come from I2 to I3
I4
I2
R
I3
R – RSU; P – Producer; I0...4 – Intermediate nodes
sBF – sharingBF Data packet
sData – special Data packet (Response to a Beacon)
Beacon
Interest for requesting a sharingBF Data packet
sData
sBF

Figure 1. Beacon processing in intermediate nodes. All intermediate nodes receive and process the sData. Only 1-hop nodes from the sender receive and process the *sharingBF*.

In the urban scenario, after a first beacon has been received from the RSU, the existing vehicles are responsible for sending a beacon themselves with a frequency of $f_{vehiclebf} = f_{RSUbf}/5$. That is, after five times the periodicity of RSU beaconing any vehicle is able to send the beacon, and whenever a beacon is received, all vehicles reset their timers. Vehicle beaconing is only triggered if at least one beacon has been received from the RSU.

In fact, the described process has five objectives: (1) query the content sources and their content; (2) allow the intermediate nodes to know the content sources; (3) allow the intermediate nodes to know the RSU; (4) allow the nodes to update their list of neighbor nodes; and (5) use the *sharingBF* Data packet to distribute the list of cached content among intermediate nodes.

With this solution, a given node need not periodically broadcast beacon messages to query the list of cached contents from its neighbors. It only gather the list from 1-hop nodes with which it interacts for forwarding the sData. The cached contents list is stored as a CCT, which is, in fact, a list of BF, each one corresponding to a specific node that has previously interacted with the actual node for forwarding the sData.

When a node receives a *sharingBF*, it extracts the received BF and the ID of the node from which the packet has been received. It looks up the received ID in the CCT. If the ID is not found in CCT, then no previous interaction with the node has occurred. Thus, a new entry is created, and the BF is stored. When the node ID is already registered in CCT, then a previous BF has been stored. If the previously stored BF is different from the received BF, then it is updated accordingly.

3.4. Processing of Incoming Data

Any intermediate node in the wireless network receives several packets, including the unsolicited Data. When a Data packet is received, CMAF processes it as presented in Algorithm 2. The first step is to check the application communication model, whether it is push- or pull-based (Lines 1 and 4). Then, the model looks in more detail to other application requirements (e.g., application type) as explained next.

Algorithm 2: Incoming Data processing

```
   Input  : Data
1  if (Push-based Data) then
2      if (Data still valid) then
3          Broadcast (Data) ;
4  else
5      if (Corresponding Interest exists in PIT) then
6          if (Efficiency-related Data) then
7              Broadcast (Data);
8          else
9              Add_To_CS (Data);
10             if (NT not empty) then
11                 threshold ← mean(NumberOfNeighbors_Last_3s());
12                 hopCount ← getHopCount(Data);
13                 if (hopCount greater than threshold) then
14                     Return;
15                 nextHop ← SelectFromNTaNextHop_WithShortestDistanceToConsumer ();
16                 SendData (Data, nextHop) ;
17     else
18         if (Long-lived Data) then
19             Add_To_CS (Data);
```

3.4.1. Awareness on Application Type

Some classes of applications (i.e., safety including warning messages) and efficiency classes require that a vehicle be able to proactively share this kind of information. This is the case of push-based communication. With the stringent time constraints, and the highly dynamic topology, these applications cannot afford the routing procedures delays. These applications have a short-lived time, which would force the constant modification of the main structures of NDN (i.e., PIT, FIB and CS). Furthermore, as referred by [63], besides the fact that delays of warning messages should be low, in the milliseconds range, it should be ensured that all vehicles within a given area receive the messages in a timely manner. Therefore, we consider that a broadcast instead of routing is a better choice for this class. Existing proposals use different mechanisms to deal with push-based Data. Some of these mechanisms and their characteristics include those presented in the next studies.

Although the work in [69] states that the proposed communication model is V2V, the proposed name-based scheme clearly works only where an infrastructure exists, and is used to delimit the region where broadcast should take place. This proposal relies on naming for selecting the push-based among all the unsolicited Data. Resorting to naming, the way the design is performed, as well as the subsequent prefix match, as proposed in this study, restricts somehow the dissemination of push-based Data in the V2V scenario. The proposal by [70] relies on beacon messages to initiate the processing of push-based Data. The beacon messages are used to guide consumers to prepare a PIT entry that should be used to accept the Data to be sent to the network. This scheme floods the network with a beacon, prior to sending push-based content. Resource usage is not optimized with this procedure. The authors in [71] propose three strategies for push-based Data in IoT: (1) Interest notification, appending small Data to Interest, modifies the semantics of the Interest packet. (2) With unsolicited, all Data packets are accepted and stored in CS as proposed in [4]. Then, like in the previous strategy, the consumer broadcasts an acknowledgment packet, confirming the reception of the Data. (3) Virtual Interest polling, based on long-lived Interest, locks a PIT entry for a long time. Using additional broadcast messages to prepare the node on the reception of an unsolicited Data as proposed in [70,71] leads to increased network traffic. The study by [72] proposes an Android-based application for named-Data emergency network services, with a push-based Data strategy. The mentioned strategy allows the forwarding plane to process a Data packet which does not have a corresponding PIT entry.

Push-Based Data

Although our proposal could use a scheme similar to [72], which clearly distinguishes the push-based content (which we consider for safety applications) and the other unsolicited Data packet that can arrive at a given node, we propose a generic name-based scheme to classify the contents, and treat differently contents falling into the safety class. CMAF proposes a scheme where all safety-related contents are based on the prefix presented in Table 2.

Table 2. Prefix for push-based content.

/push-based/info-type/sender-ID/sender-geo-coordinates/

The first component is used to identify the type of content as being push-based and destined for broadcast. We can have different types of push-based content. For instance, besides the active safety content, we have the content destined for road efficiency. The latter is longer lived than the former; thus, we can treat it differently from the former. This is the reasoning behind the distinction that will be provided by the second component.

The third component is the identification of the sender, and the last component is the geographical location of the sender. We consider it important to have the geographical coordinates of the sender, considering that safety content is location dependent. The broadcast information is only important a hundred meters away from the location where it has been sent.

If the received Data packet is push based, its validity is checked. If still valid, the Data are broadcast. As referred before, broadcasting is performed in order to shorten the delay that would exist if a routing procedure were to take place. If Data validity has elapsed, then the packet is discarded.

Only applications in the safety and efficiency classes can push Data into the network. Applications in the comfort and interactive-entertainment classes can provide Data only by request. Some applications in the efficiency class can also provide Data by request. This is the case of pull-based communication, for which the forwarding procedure is described next.

Pull-Based Data

Similar to the original procedure in NDN, when Data are received, a corresponding PIT entry is looked up. If a corresponding PIT entry does not exist, then the content is unsolicited. To take advantage of the less stringent characteristic of VANET in terms of CS size, unsolicited content is cached, similar to [4]. If a PIT entry exists, then the message was requested. The next step is to check if the Data fall into the efficiency class, in which case the Data are broadcast as explained before. As the Data of some efficiency-related applications may live longer, then, they are cached. The long-lived Data application includes, for instance, the audio/video downloaded.

If the received Data are classified into the comfort or interactive-entertainment classes, then the original NDN procedure takes place, i.e., its corresponding FIB entry is updated, and Data are added to the CS and then forwarded downstream. Data are forwarded downstream using breadcrumbs from PIT entries. As observed by [73], the RTT 95th percentile for an Interest–Data transaction is less than 300 ms. In the referred period, the vehicles do not move far between the time an Interest is issued and the time the corresponding Data packet is received, thus ensuring the validity of the Interest breadcrumbs in the PIT and the effective retrieval of Data packets. Therefore, CMAF assumes that even considering the topology is dynamic; the created paths from upstream forwarding will have enough stability for Data forwarding, mainly for urban scenarios where the speed of vehicles is relatively low.

Now, even considering the aforementioned assumption about the relative stability of routes, the link duration in VANET is relatively short, and the topology is continuously changing, which result in a frequent network disruption [74]. In the case of route disruption, the last node sending the packet gets, from the Interest, the geographical coordinates of the consumer and can predict the new consumer location, and a possible route to use for forwarding the Data to it. The updated NT is checked to determine the better relay node for Data delivery. Differently from other solutions, in which the Data packet is always forwarded whether there is or not a neighbor for forwarding it further, in CMAF, the node only sends the Data packet when there is at least one *active* neighbor. This way, the network resources are only used when forwarding to the next-hop is guaranteed.

3.4.2. Awareness on Network Density

Studies such as [4] limits the hop count for a fixed value. The number of hops a packet traverses from its origin to the destination depends on the network density. Therefore, limiting the packet forwarding for a fixed value for different network densities may not be a better solution. CMAF is designed to be aware of the network density. Differently from the aforementioned study, we propose to limit the traversed hops based on the current number of neighbors a given intermediate node has. Before the packet is sent out, downstream, the model computes the average number of neighbors (used as a *threshold*) the node has encountered for a previous period of $t_{hopTheeshold} = 3000$ ms (Algorithm 2, Line 11). This value is chosen from simulations. When the received packet has already traversed a number of hops superior to the *threshold*, the packet is discarded, and the consumer eventually sends the request again, which can hopefully be satisfied by any node in its vicinity that has received the packet before. In fact, from simulations, the *threshold* is corrected to Equation (1) when $threshold > \alpha$:

$$threshold = threshold - \beta \tag{1}$$

The chosen values are $\alpha = 14$ and $\beta = 2$. We highlight that only Data packets have their hops limited. This is due the fact that flooding Data are more disastrous than flooding Interests, given their size. Moreover, the model allows Interest flooding in order to leverage the caching system, that is, even if the requested Data do not reach the requester due to the hop limit, the next time the same Data are requested, they can be fetched from a nearby node that can have them already cached from the previous request.

3.5. Procedure for Incoming Interest

Algorithm 3 presents the processing of an incoming Interest. When a Interest is received, it is inserted into PIT. As the sender node is 1-hop away from the receiver, the node which is taken as a neighbor, gets its identification (including its NMSI) inserted (if new) or its NMSI updated in the corresponding NT entry as explained by Algorithm 1 in Section 3.3.1.

Algorithm 3: Incoming Interest processing

```
Input  : Interest
InsertOrUpdatePIT (Interest);
InsertOrUpdateNT (Interest);
if (Interest from 1-hop RSU) then
    InsertOrUpdateNTRsu (Interest);
if (Corresponding Data in CS) then
    ReturnToRequester (Data, destId);
    Return;
else
    if (Destined to RSU) then
        SendInterestToRSU (Interest);
        Return;
    else
        if (Efficiency or push-based) then
            BroadcastInterest (Interest);
        else
            if (Known nextHop) then
                if (NT not empty) then
                    if (nextHop is neighbor) then
                        SendInterest (Interest, nextHop);
                    else
                        neighbors ← getListOfNeighbors();
                        neighbor ← NeighborWithMinDistanceToNextHop (neighbors);
                        SendInterest (Interest, neighbor);
                else
                    Return;
            else
                if (Interest in CCT) then
                    nextHop ← NextHopFromCCT ();
                    n_geoPosition ← NextHopPosition (nextHop);
                    o_geoPosition ← GetOwnPosition ();
                    d ← CalculateDistanceToNeighbor (n_geoPosition, o_geoPosition);
                    if (NT not empty) then
                        if (nextHop is neighbor) then
                            SendInterest (Interest, nextHop);
                        else
                            neighbors ← getListOfNeighbors();
                            neighbor ← NeighborWithMinDistanceToNextHop (neighbors);
                            SendInterest (Interest, neighbor);
                else
                    if (NTRsu not empty) then
                        SendInterest (Interest, RSU);
                    else if (NT not empty) then
                        BroadcastInterest (Interest);
```

After updating the NT, it is verified whether the received Interest is duplicated or not. If it is duplicated, a procedure for looped Interest takes place. Otherwise, it is verified if it is a pending Interest. If pending, then a CS miss has occurred. If not pending, then a CS lookup takes place. In a case of CS hit, the corresponding Data packet is returned. In the case of CS miss, it is verified whether the Interest is or is not explicitly destined to a RSU. If it is indeed destined to a RSU, then it is sent to the selected RSU.

If the Interest is not marked as destined for a specific RSU, then it is checked if it comes from an efficiency-related application, Line 13. If it comes from an efficiency-related application, then it is immediately broadcast. The reason for the broadcast decision is explained in Section 3.4.1.

If the Interest cannot be satisfied by the present node, then its corresponding Data source still needs to be discovered. Existing solutions follow an almost similar procedure from this stage on, which is the immediate Interest broadcasting for content discovery. The study by [67], for instance, which uses FIB, follows a slightly different mechanism. The Interest is forwarded towards the RSU, which is supposed to have knowledge about other routes. If routes are not found even on RSU, the requester node delays its requests and awaits the creation of FIB entry, from a general beaconing process. CMAF deals with contents related to different applications classes differently. But, for comfort and interactive-entertainment applications, it follows a similar mechanism to that of the mentioned study. However, differently from the referred study, the proposed model does not await the creation of an FIB entry (CMAF does not use FIB). Instead, it broadcasts the Interest if there is any neighbor around. Before broadcasting the Interest, however, a last feature is explored, which is the use of known cached contents (i.e., see CCT, in Section 3.2) from other nodes.

Only if the content is not found in CCT, the broadcast procedure for content discovery takes place. If the corresponding Data are found in this table, then an updated route (i.e., geo-location) to the content is used, and the Interest if forwarded to the node holding the content. The broadcast for content discovery is performed in two different ways, depending on the current network scenario, as described in Section 3.5.2.

3.5.1. Forwarding Enhanced By Caching

To the best of our knowledge, the existing proposals for NDN-based forwarding for VANET do not usually explore caching for deciding where to forward packets. The sole exception found is the study by [75]. The aforementioned study explores caching by inquiring into the contents in the neighbor's cache. Vehicles know the neighbors' cached contents by periodically notifying them of their local cache information. This notification is performed via broadcasting beacon messages. All nodes broadcast the beacon.

Considering that periodic beacon broadcasting, by all nodes, increases network traffic and subsequently increases congestion and potential collision, CMAF proposes a different solution. In this solution, a producer keeps an updated list of its local cached content, and will append it to the sData packet via a BF, which is sent by intermediate nodes in the forwarding path from the content source to the RSU, in response to a beacon sent by the RSU (process explained in Section 3.3.2). The aforementioned process is only considered for urban scenarios where the network includes a RSU. Another use of cached content in the intermediate node is also explained in Section 3.3.2 and is related to the use of CCT and the *sharingBF* Data packet.

As referred before, the model proposed in this work leverages the less stringent characteristics of VANET, concerning the storage size, and caches all received Data. But, differently from other works (e.g., [4]), the present model caches unsolicited Data based on their classification in terms of application type. The short-lived content from the active–safety application class is not cached. Some long-lived content from the efficiency application class are cached along with the long-lived content from the interactive-entertainment- and comfort-related applications.

The reasoning behind the decision to avoid the caching of some short-lived content can be reinforced by the works in [76,77], which study and propose caching strategies based on the content lifetime. As stated in the referenced works, the CS space is not efficiently used when caching contents that are then shortly removed.

3.5.2. Awareness on Network Scenario

VANET is highly dynamic with frequent partitioning, having relatively short inter-contact times. To deal with this partitioning, it is necessary to build protocols that can efficiently adapt between highly disconnected, dynamically partitioning networks and non-partitioned networks [74]. Vehicles traveling between these scenarios need to adapt their behavior to the current network characteristics in order to provide high QoS and QoE to the users.

Having in mind the description presented in Section 3.3 for content discovery, when a consumer or any intermediate node does not find matching Data in its CCT, the node proceeds in two different modes according to the current scenario as detailed next.

For urban scenario where vehicles move at relatively lower speeds and there exists a higher density of vehicles and infrastructures, broadcasting has a more negative impact than in the rural scenario. Thus, as all nodes have a list of all existing RSUs in the network, and the RSUs have better knowledge of routes to the content sources (given their periodic beaconing to discover contents as described in Section 3.3), the proposed model prioritizes the V2I network communication model and redirects the Interest to the infrastructure (i.e., RSU). It is a matter of preference, not a limitation in only one communication model. If before reaching the RSU, a route to the content source is found in any of the intermediate nodes, then from this node, the request is unicasted using the new route. Otherwise, the Interest forwarding continues until it reaches the RSU, and from there, it is hopefully forwarded to a known content source. If no content source is known from the RSU (i.e., not even a CCT entry exists), then it is broadcast. The Interest is only broadcast as the last resort. The described procedure is performed in order to reduce the Interest broadcasting.

In a rural scenario, the preferred network communication model is V2V. Having in mind that this environment is less dense in terms of infrastructure and vehicles, thus, a broadcast is less problematic. Therefore, being the last solution left, a broadcast is used.

As presented in Section 2, state-of-the-art solutions adopt a timer- and contention-based mechanism for tackling the broadcast storm problem induced by allowing all nodes to forward the same received Interest. The time-based mechanism as presented by those solutions allows the nodes to send an Interest, one node at a time, and the other nodes suppress the forwarding when overhearing the same packet from another node. Some of these solutions disseminate the Interest in one direction and others in two directions, one for each road direction. CMAF proposes a different approach. In order to ensure the dissemination of the received Interest in all directions, all nodes receiving an Interest immediately forwards it and then stores it in a timer-based container with a duration of order of magnitude of the retransmission time (i.e., $t_{RTX} = 3$ s). The first Interest is always forwarded. If the same Interest (i.e., same sequence number) is received before the timer elapses, it is dropped, and the container timer is reset.

3.6. Short-Term Mobility Prediction

Aiming at reducing flooding for content and neighbor discovery (or to find the new position of the moved content source) and consequently reduce network congestion, delivery delay, and packet loss, CMAF resorts to mobility prediction for tracking the geographical position of the moving nodes. For this regard, new specific tables (see Section 3.2) are included for managing neighbor nodes, including their geographical positions, and the existing content in the neighbor's CS. These tables are periodically updated by the mobility prediction algorithms. Moreover, leveraging the in-network caching, the proposed solution takes advantage of vehicles moving between disconnected regions to opportunistically distribute contents between these regions as "content mules".

The key characteristics of VANET (i.e., highly dynamic topology, and intermittency of connectivity) make it difficult to build and maintain FIB [4,33]. Although these characteristics impose specific challenges for several applications, they can be exploited, for instance, for developing better movement management. In fact, due to the movement constraints imposed by the road topology and traffic conditions combined with the repetitive characteristic of users' movements, the future movements of the vehicles are usually predictable. As concluded by [78] on the predictability limits of vehicular mobility, there exists some strong regularity in the daily vehicular mobility in both the temporal and spatial dimensions. This regularity can be exploited to predict the vehicular mobility with a high degree of prediction accuracy.

Some representative applications that take advantage of prediction algorithms in VANET include routing, forwarding, traffic management, road safety, handover management, resource management, and location-based applications [79,80]. Based on their objectives, the prediction algorithms can be categorized into (1) link stability prediction; (2) location prediction; (3) trajectory prediction; (4) traveling time prediction; and (5) collision prediction. See [79] for details.

Mobility prediction is a component of mobility management [81] and can be viewed as the means by which a system proactively estimates the future location of mobile nodes. Mobility prediction algorithms can be categorized into various groups depending on the domain that is used as reference. For instance, the authors in [82] use the following categorization: (1) domain-independent algorithms, which are based on past node mobility history to extract context and predict the future location, usually requiring high computational capacity for algorithm training; and (2) domain-specific algorithms, which use the geometry of node motion and the semantics of the symbols in node's mobility history. The work by [83] considers (1) the deterministic model, which uses the vehicle displacement/kinetics to compute the future position; (2) the history-based model, where the model learns from past historic movement patterns to predict future position; and (3) stochastic models, where the focus is on correcting the prediction error using probabilities. With only different names, the categories of domain-independent and domain-specific algorithms from [82] are respectively equivalent to the history-based and deterministic models from [83].

State-of-the-art location and mobility prediction approaches include but are not limited to Dead Reckoning, Markov Chain, Hidden Markov Model, Artificial Neural Network, Data mining, and Filtering.

Dead Reckoning (DR)—this is an approach that is able to obtain the current node position and the distance traveled since the last known position [83,84]. It is generally used to overcome the non-existence of GPS signals, and can be used only for a short period in VANET since it can easily accumulate errors [85].

Markov Chain (MC)—this is a Markov model in which each state corresponds to an observable (physical) event [86,87]. Markov models can use the movement history in cellular/mobile or vehicular networks, in which user/node movements are *mined* from geographical positioning system (e.g., GPS and Galileo) traces.

Hidden Markov Model (HMM)—this is based on the Markov model, like the MC, and is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden) but can only be observed through another set of stochastic processes that produce the sequence of observations [86]. MC and HMM are probabilistic models that are built assuming that, given a sequence of states, the probability of a next state depends only on the current state and not on any other previous states (a Markov property). This approach defines a memoryless property for creating the model. The Markov property, however, can be extended by considering a more previous state (e.g., the work by [88]), which is dubbed the order of the model.

Artificial Neural Network (ANN)—this is a machine learning-based approach [89,90]. It is suitable for mobility prediction in VANET by being able to give solutions to complex problems due to the fact that they are non-linear processes, have the capacity for learning and generalization, and present efficient hardware implementation [91].

Data mining—this is the science of extracting useful knowledge from huge data repositories [92]. For mobility prediction, it is based on different knowledge, such as road topology information, user behavior, and movement parameters [80].

Filtering—this groups schemes such as (i) Particle Filters (PFs) [93], which are sequential Monte Carlo methods based on the weight representation of probability densities of any given state model [94]; (ii) KF [95,96], an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements; and (iii) its variation (i.e., Extended Kalman Filter—EKF [97], and Unscented Kalman Filter—UKF), proposed to deal with non-linear dynamics and non-linear measurement models.

The study in [94] presents a performance evaluation and subsequent comparison of the following algorithms: KF, EKF, UKF, Alpha–Beta–Gamma (ABG) filter [98], and PF. The study in [99] presents the comparison of ANN- and a KF-based algorithm. The study by [80] presents a comparative description of different algorithms (e.g., MC, HMM, Artificial Neural Network (ANN), Bayesian network, KF, and data mining based) in mobility prediction. The comparison considers (a) the main features and accuracy; (b) prediction output; (c) prediction metrics; (d) required information by the algorithm; and e) the main distinction of these algorithms.

The aforementioned studies conclude that ANN-based solutions present better performance for both linear and non-linear systems but at the cost of much more training, which requires much more computational power. Moreover, although ANN-based solutions can obtain good prediction results, they are generally difficult to converge and are prone to local minima [90]. PF-based solutions, designed for non-linear systems, present better performance than KF-based solutions but at the cost of more dedicated training. UKF-based solutions perform better than the solutions based on the original KF or EKF solution. The EKF solutions are designed for locally linear systems. Table 3 presents a brief comparison of the characteristics of filter-based solutions. MC- and HMM-based solution are designed to work with non-linear systems. They require training but are less computationally complex than the ANN-based solutions. HMM solutions can enable the mobile node to learn the environment and update the information itself, increasing their performance further. Therefore, the HMM-based solutions present better performance than the MC-based ones but at the cost of more dedicated training.

Table 3. Comparison of filter-based estimator approaches ([100]).

Algorithm (Estimator)	Model	Assumed Distribution	Computational Cost
KF	Linear	Gaussian	Low
EKF	Locally linear	Gaussian	Low (For analytically computed Jacobians) Medium (For numerically computed Jacobians)
UKF	Non-linear	Gaussian	Medium
PF	Non-linear	Non-Gaussian	High

The proposed model uses the STMP algorithm (i.e., only a future location prediction) for updating NT in all vehicles. The model is based on the EKF formulation presented in [86]. The use of STMP makes sense mainly in an urban scenario, where the density of the infrastructures and road topology constrains the duration of inter-vehicle link, which forces the need for frequently updating the nodes' NTs. For systems with higher non-linear proprieties, the alternatives to EKF can be UKF and PF, which, instead of approximating a non-linear function, approximate the probability distribution (sigma points for UKF, or particles for PF—for any distribution, not only Gaussian [100]). In CMAF, we adopt EKF assuming that the time interval from the last packet received from a neighbor, and the possible need to select this neighbor as a relay, will be considerably short. Our reasoning is that if the interval is short, we can also consider the traveled distance to be short, and subsequently assume the mobility to be partially linear or easily transformed to non-linear. This way, we keep the processing cost in vehicles low.

3.7. Research Methodology

The present work is part of a context-aware V-NDN architecture described in [29]. The mentioned architecture includes a routing protocol and the forwarding strategy presented in this work. Both the routing and forwarding strategy are enhanced by mobility prediction. Initially, in order to assess the existing research gaps from the state of the art, a Systematic Literature Review (SLR) on the realization of VANET by NDN was developed in [28].

This work refines the ideas presented in [29], for the forwarding strategy. It presents the actual implementation and evaluation. It also begins with a literature review, specifically for forwarding strategies. Although not exhaustive, it presents the state of the art and compares the existing solutions, highlighting the main differences from the proposal presented herein. Then, the proposed model is presented. The developed model underwent extensive simulations for its validation as presented in Section 4. The model was evaluated with and without the mobility prediction. Related works (e.g., CCLF, VNDN) were deployed in ndnSIM alongside the other existing strategies (i.e., Multicast).

All implementation code is written in C++ programming language. Scripts, for data extraction, graph plotting and statistics, are written in Python, and as described in Section 4.2, the simulation environment is based on ndnSIM.

4. Performance Evaluation

This section presents the simulation environment and the simulation results. We perform extensive simulations to assess the performance of the proposed protocol. The protocol is evaluated in three distinct modes: (i) V2V without a beacon broadcast; (ii) V2I with a beacon broadcast at a frequency of 1/5 beacons per second; and (iii) V2I with a beacon broadcast at a frequency of 1/10 beacons per second. The results of the simulations are compared with the following similar works: (a) VNDN protocol [4], without limiting the hop count to 5, which worsens the overall performance mainly for dense networks (e.g., network with 100, 120 nodes); (b) mutlicast [101]; and (c) CCLF [47]. Two network topologies (City of Porto and Manhattan-like network) are chosen for simulations as presented in Section 4.2.

4.1. Selected Metrics

We select six metrics for performance assessment and comparison with similar studies. These are the metrics usually used to assess the performance of similar studies.

- Interest Satisfaction Ratio (ISR)—Also dubbed the Packet Delivery Ratio (PDR), this refers to the ratio between the satisfied Interest (or the corresponding received Data) to the total Interest transmitted by the consumer(s). We compute this value at the consumer’s application Face.
- Interest Satisfaction Delay (ISD)—Refers to the amount of time taken from the Interest being sent by the consumer to the time the corresponding Data are received (i.e., total Round Trip Time—RTT). Two values can be analyzed: The last delay, which corresponds to the hiatus for the last Interest transmitted, and the full delay which refers to the delay, including retransmissions. We only consider the full delay.
- Jitter—For real-time applications, such as streaming a video file, delay between receiving a pair of video chunks should remain constant. Variations in the delays on streaming chunks of the the video can compromise the quality of the streaming. Jitter can be measured as the statistical variance in the latency/delay of received Data packets [102];
- Hop count—The requested content may be several nodes away from its holder. In a dynamic environment such as the case of VANET, it is even expected to receive content from a “data mule” node, bringing the content from an even further distance. The hop count refers to the average number of nodes Data packets traverse from the content holder (either a producer or any other intermediate node holding the Data) to the consumer.
- Number of retransmissions—When the Interest lifetime expires before its corresponding Data packets are received by the consumer, the consumer may issue another request for the same Data, should it still want the content. The number of retrans-

missions refers to the average number of transmissions needed to finally fetch the desired content.

- Transmission Overhead (TO)—The total number of all packets (i.e., Interest, Data, and control/management packets such as beacons) in the network. It measures the level of congestion in the network. This value is further normalized by the TO of the basic NDN flooding protocol.

4.2. Simulation Environment

For simulations, we use the network simulator for NDN (ndnSIM) [103] version 2.8, along with the Simulation of Urban MObility (SUMO) [104] version 1.16.0, used to generate the mobility of the nodes. ndnSIM is a modular simulator, written in C++ programming language, and runs under the Network Simulator (NS-3) [105], which is largely used for simulations in VANET [106]. It leverages the NS-3 to create the simulation topology and specify the respective parameters to simulate the link layer protocol model and the communication between nodes, and record the simulation events [103]. The simulations are conducted in a 6-core (12 threads) CPU laptop with 32 GB of RAM, running Gentoo Linux Base System release 2.13, with GCC-12.3.1 and kernel version 6.1.12-gentoo.

Two road network topologies are used for simulations: (1) A 4 × 3 Manhattan-like grid scenario (Figure 2) with an area of 1200 × 1600 m^2. Each street is bidirectional and has 2 lanes for each direction. Vehicles depart randomly and have varying speeds, from 0 m/s to 16.39 m/s. The probability of turning left or right on intersections if fixed as 50% straight, 25% right and 25% left; (2) City of Porto, Portugal (Figure 3). This scenario covers an area of 1000 × 1000 m^2, vehicles depart randomly and have varying speeds, from 0 m/s to 34.10 m/s. This scenario includes highways and the possibility of having disconnected sections. That is, the city of Porto scenario is more susceptible of having disconnected segments than the Manhattan's, therefore, the robustness of CMAF for partitioned network is well tested in that (Porto) scenario.

All scenarios have the indicated number of vehicles (i.e., 20, 40, 60, 80, 100, and 120) for the duration of each simulation.

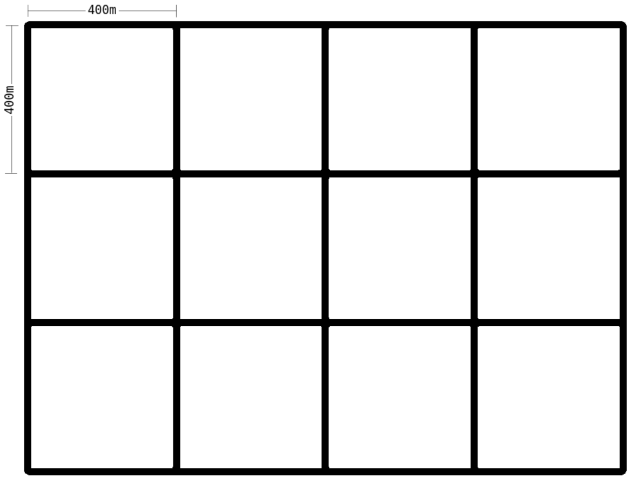


Figure 2. Manhattan-like scenario.

Table 4 presents the protocol parameters, and Table 5 the parameters configured for simulations. Understanding some of the parameters requires explanation, and thus we provide it next.

- Simulations time: Configured to run for 260 s. For data extraction, however, we limited the period sample to 250 s, ignoring the first 1 s used as the warm-up time.
- Network density per scenario: Although possible, we did not force vehicles to depart at *time* = 0.0. We calibrated the maximum number of vehicles in the scenario so that vehicles could enter and depart from the scenario at their specific time. However,

all vehicles (maximum number of vehicles per scenario, e.g., 20 for scenario with 20 nodes) are in the simulation after the first 10 s.

- $t_{hopThreshold}$: A previous value of $t_{hopThreshold} = 12{,}500$ ms was used considering the time for traveling 250 m (the defined transmission range), with a maximum speed of 20 m/s in the urban scenario. The results were not satisfactory, as we consider them with the new value. With higher $t_{hopThreshold}$, the average number of neighbors is more consistent.

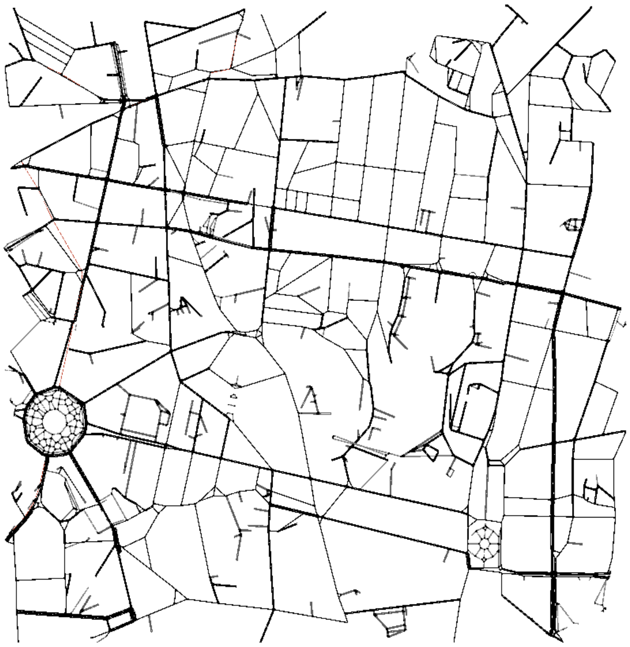


Figure 3. Scenario based on the City of Porto, Portugal.

Table 4. Parameters of the protocol.

Configured Parameters	Value
$t_{neighborUpdate}$	600 ms
$t_{NTpurge}$	$3 \times t_{neighborUpdate} + 100$ ms = 1900 ms
f_{RSUbf}	0.2 Beacons/s, 0.1 Beacons/s
$t_{hopThreshold}$	3000 ms

Table 5. Parameters for the simulation's environment.

Configured Parameters	Value
Network Topology/Scenario	City of Porto, Manhattan-like scenario
Number of producers	1
Number of consumers	1
Content request rate	5 packets/s
Forwarding strategy	CMAF
Propagation Loss Model	Two Ray Ground
Vehicle speed	Max 16.39 m/s Manhattan-like. Max 34.10 m/s Porto
Wireless Comm. Standard	IEEE802.11p
Data rate	12 Mbps
Channel bandwidth	10 MHz
Number of anntenas per node	1 (Omni directional)
TxPowerStart	12.0 dbm
TxPowerEnd	12.0 dbm

Table 5. *Cont.*

Configured Parameters	Value
TxGain	1.0 dbm
RxGain	1.0 dbm
Transmission Range	250 m
Network density (# nodes)	20, 40, 60, 80, 100, 120
Simulations time	260 s
Number of simulations	30
Interest lifetime	8 s
Retransmission timeout (t_{RTX})	3 s

4.3. Simulation Results

This section presents the results of simulations. We conducted extensive simulations, and some of the parameters (see Table 4) used to produce the final results presented in this section were defined after prior simulations. Section 4.3.1 presents the simulation results using the city of Porto's road network, and Section 4.3.2 presents the simulation results using the Manhattan-like grid. All results are extracted and compared with different numbers of nodes in the network (i.e., 20, 40, 60, 80, 100, and 120 nodes).

4.3.1. Simulations Using the Scenario from the City of Porto

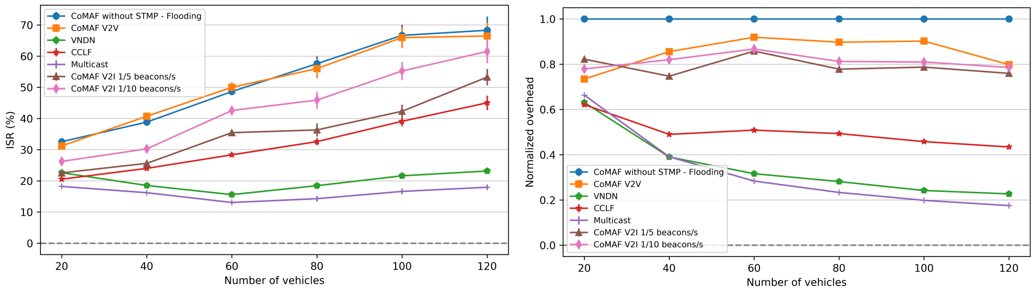
As already mentioned, due to the packet size, Data flooding is more problematic than Interest flooding. Thus, managing the Data hop count is crucial. In this section, we present the simulation results for three specific scenario configurations: (1) without limiting the Data hop count, (2) limiting the Data hop count as explained in Section 3.4.2, and (3) limiting the Interest and Data hop counts.

Simulation Results without Limiting Data Hop Count

Figure 4a shows the ISR for the selected strategies (i.e., CMAF without STMP—flooding, CMAF V2V, VNDN, CCLF, Multicast, CMAF V2I with 0.2 beacons/s, CMAF V2I with 0.1 beacons/s). Its is important to highlight that CMAF does not manage FIB. The multicast strategy resorts to FIB and sends Interest to all upstreams, based on it [101]. The results presented in Figure 4a should be analyzed with the results presented in Figure 4b that depict the level of TO produced by each protocol. CMAF without STMP performs pure flooding. CMAF V2V uses STMP for predicting the next node location as explained before. The presented results show that CMAF V2V has the same results as the CMAF without STMP with pure flooding but with a reduced TO, corresponding to a gain ranging from 7.5% to 22.5%. It means that with the mobility prediction, forwarding is more efficient and only the chosen nodes perform forwarding. Strategies that purely resort to FIB (e.g., multicast) perform badly in VANET as shown by the multicast result. CMAF V2V outperforms all other compared protocols, and the performance is consistently increasing when the number of nodes increases as expected. This is due to the fact that in a dense network, the content availability is greater than in a sparse network. The content availability can be either from the existence of a large number of CS or the high probability of completing the route to the content sources.

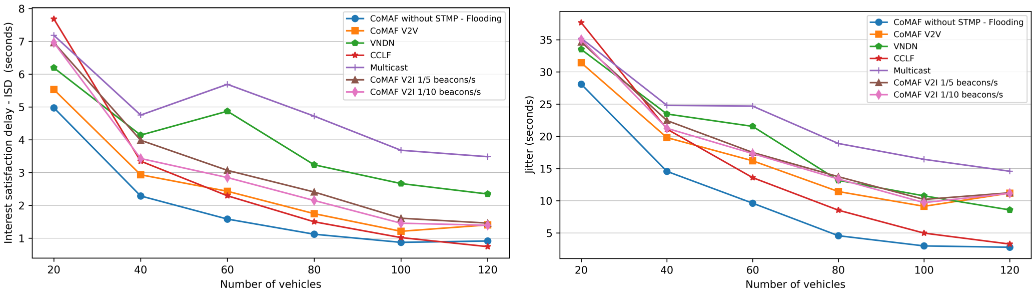
The ISR of CMAF V2I is worse than that of CMAF V2V. This is due to the additional packets (i.e., beacons) issued by the RSU. Increasing the frequency of beacon broadcast (i.e., 0.2 beacons/s) worsens the ISR for the same TO.

The high performance of CMAF V2V is gained at the cost of an increased ISD (maximum of 1 s from the flooding protocol) as presented in Figure 5a. In terms of ISD and jitter (Figure 5b), CCLF performs well mainly for dense networks.



(**a**) (**b**)

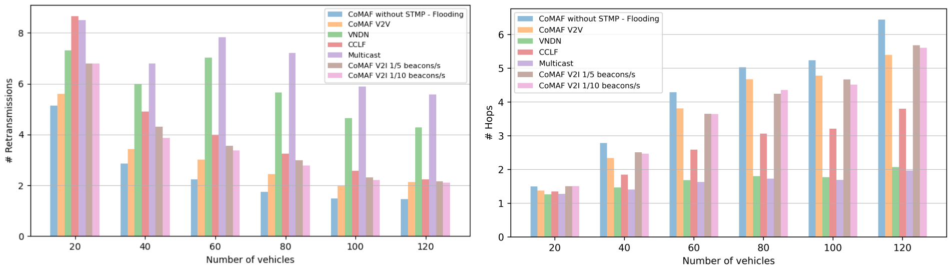
Figure 4. The Interest Satisfaction Ratio and the transmission overhead for the selected protocols without Data hop limit, with CMAF presenting the better performance. (**a**) ISR; (**b**) TO.



(**a**) (**b**)

Figure 5. Interest Satisfaction Delay and jitter for the selected protocols, for simulations without limiting Data hop count. (**a**) ISD; (**b**) Jitter.

The increased ISD can be justified by conjugating the level of packet retransmissions (Figure 6a) and the packet hop count (Figure 6b). In general, for CMAF, Interests traverse more nodes to fetch the requested content. Consistently, in a dense network, Interests traverse more nodes to fetch the requested content, and the corresponding content is easily returned to the consumer with fewer retransmissions. In a sparse network, nodes are dispersed, and the Interests lifetime may easily expire before they reach the content provider, forcing more retransmission from the consumer. Comparing the scenario with 120 nodes (a dense network), we note that for the same level of packet retransmissions for CMAF and CCLF, the latter presents fewer traversed nodes to fetch content. Therefore, for dense networks, the CCLF ISD is better than that of CMAF. As expected, VNDN and multicast present the highest ISD, which is justified by their results in terms of packet retransmissions and average hop count.

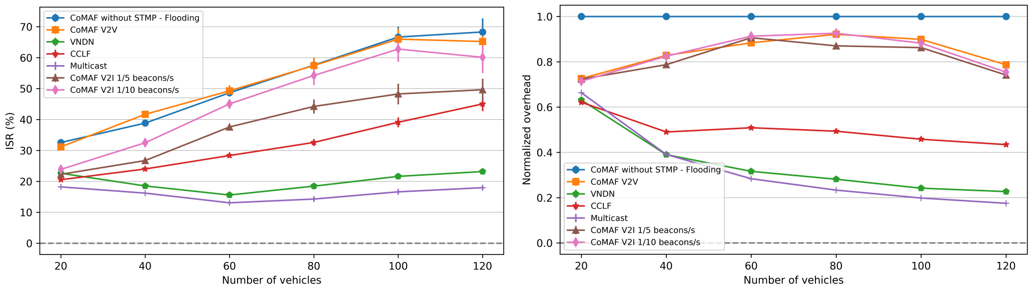


(**a**) (**b**)

Figure 6. Average packet retransmissions and hop count presented by the protocols. (**a**) Retransmissions; (**b**) number of traversed nodes (hop count).

Simulation Results Limiting Data Hop Count

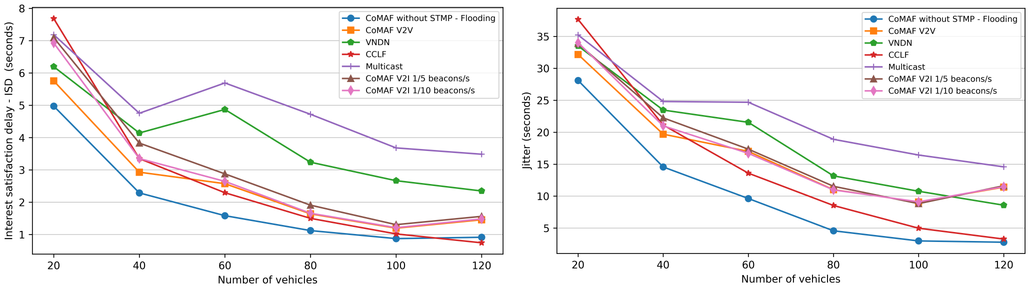
We performed a set of simulations varying the *threshold* used for limiting the Data hop count. If the hop count is limited to the threshold as presented in Section 3.4.2 with $\alpha = 14$ and $\beta = 2$, the ISR and TO are as depicted in Figure 7a,b, respectively. The results are consistent with the ones presented in the previous section. In terms of ISR, CMAF V2V outperforms all other compared protocols. The performance is consistently increasing when the number of nodes increases as expected. In the present case, however, the network with 120 nodes presents an exception. The ISR of CMAF V2V or V2I for the mentioned scenario (i.e., network with 120 nodes) is relatively lower than the ISR of CMAF—flooding, comparatively with the other scenarios with lesser nodes. It means that although the Interests are allowed and able to traverse many nodes to the corresponding content provider, for dense networks, the chosen *threshold* for Data forwarding is relatively low, causing Data to be discarded early before reaching their destination. This can be observed by comparing the average hop count from Figure 6 and the similar Figure from Section Simulation Results Limiting Interest and Data Hop Count. Only the average hop count of the network with 120 nodes is significantly decreased. For networks with 60, 80, and 100 nodes, for instance, the overall average hop count is even increased with this new configuration (i.e., with limited Data hop count).



(**a**) (**b**)

Figure 7. Interest satisfaction ratio and the transmission overhead, with limited Data forwarding hop count. (**a**) ISR; (**b**) TO.

From the results, we also highlight that when limiting the Data hop count, the network resources are better managed and, as a result, the scenario (i.e., CMAF V2I) that presents lower performance (see Figure 4) due to the additional packets (i.e. beacons) now presents improved performance at the cost of an increased ISD and jitter, Figure 8, and with a TO majored by the CMAF V2V.

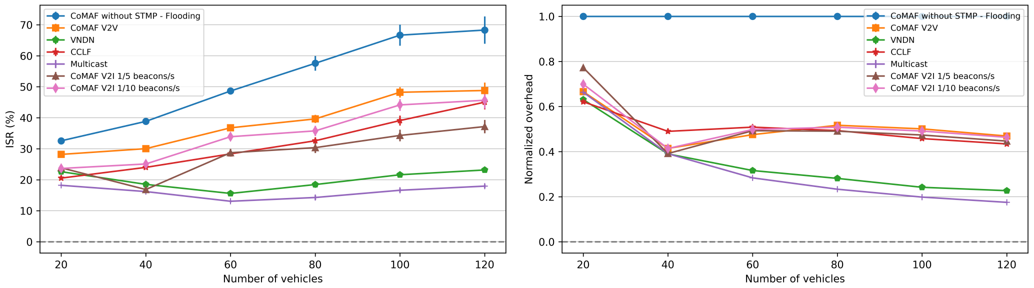


(**a**) (**b**)

Figure 8. Interest Satisfaction Delay and jitter for the selected protocols, for simulations with limited Data hop count. (**a**) ISD; (**b**) jitter.

Simulation Results Limiting Interest and Data Hop Count

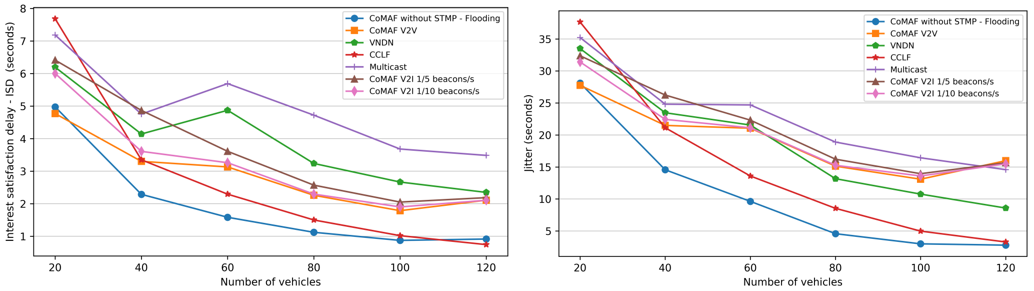
For the sake of comparison, we further investigate the performance of CMAF with TO as the same order of magnitude as the presented by CCLF. For this matter, both the Data and Interest hop counts are limited. For this simulation, $threshold = \{2, 2, 4, 5, 6, 7\}$ for both Data and Interest packets, respectively, for $20, 40, 60, 80, 100$ and 120 nodes. The chosen values are a result of tentative simulation configurations. The results are presented in Figure 9a,b for ISR and TO, respectively.



(**a**) (**b**)

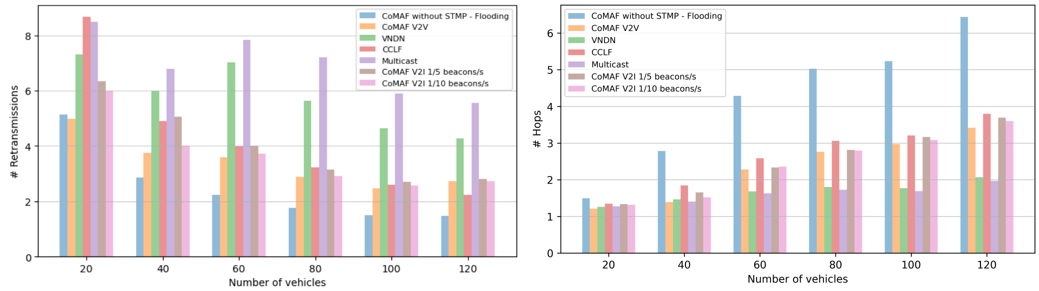
Figure 9. Interest Satisfaction Ratio and the transmission overhead, where Interest and Data forwarding are only allowed for limited hops, in order to have the CMAF TO with the same order of magnitude as that of CCLF. (**a**) ISR; (**b**) TO.

As depicted, CMAF V2V still presents the best performance (i.e., ISR vs. TO) at the cost of an increased ISD and jitter (Figure 10) as expected due to the fewer hops traversed by nodes in CMAF, Figure 11b. CMAF V2V and V2I also present fewer retransmissions (except the case of the scenario with 120 nodes) than CCLF, Figure 11a.



(**a**) (**b**)

Figure 10. Interest Satisfaction Delay and jitter for the selected protocols, for simulations with limited Interest and Data hop counts. (**a**) ISD; (**b**) jitter.



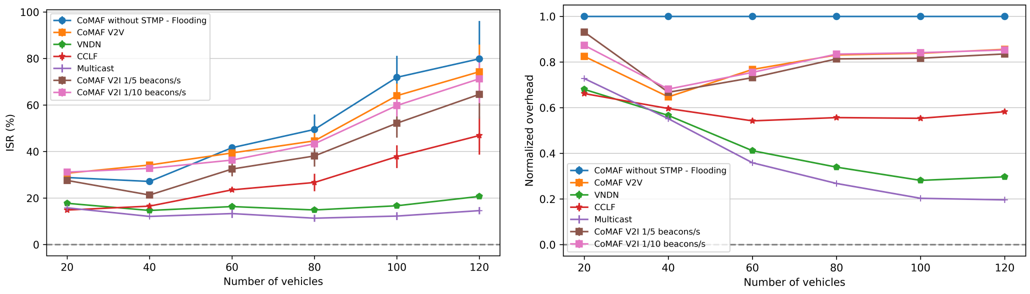
(**a**) (**b**)

Figure 11. Average packet retransmissions and hop count presented by CMAF, for scenario with TO with the same order of magnitude as that of the CCLF. (**a**) Retransmissions; (**b**) number of traversed nodes (hop count).

4.3.2. Simulations Using a Manhattan-like Road Network

In the Manhattan-like grid, the nodes are more likely to re-encounter each other during the simulations, than in the city of Porto's road network, where several roads are disconnected and content forwarding may significantly resort to the so-called "data mules". However, the overall results presented in this section are consistent with the ones presented in Section 4.3.1.

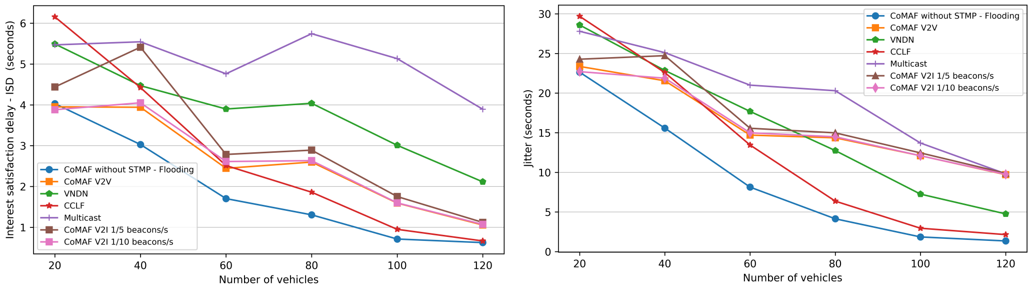
Figure 12a,b portray, respectively, the ISR and TO of the evaluated protocols. Again, and now with an average gain of 20% in terms of reduced TO compared to the CMAF—flooding, the proposed CMAF presents the best performance among the compared state-of-the-art protocols. The presented performance is relatively high than that presented for the city of Porto's road network due to the fact presented in the previous paragraph.



(**a**) (**b**)

Figure 12. Interest Satisfaction Delay and the transmission overhead. (**a**) ISR; (**b**) TO.

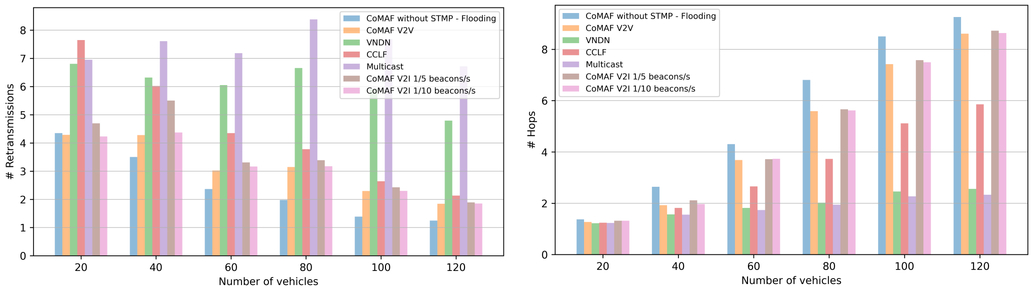
Figure 13a,b present the ISD and jitter, respectively. Again, as presented in the previous sections, in terms of the ISD and jitter, CCLF performs well mainly for dense networks.



(**a**) (**b**)

Figure 13. Interest Satisfaction Delay and jitter for the selected protocols. (**a**) ISD; (**b**) jitter.

CMAF presents fewer retransmissions (Figure 14a) as a result of more traversed nodes (Figure 14b) than the compared protocols (i.e., VNDN, CCLF, and multicast).



(**a**) (**b**)

Figure 14. Average retransmissions and hop count presented by the protocols. (**a**) Retransmissions; (**b**) number of traversed nodes (hop count).

5. Discussion

The proposed work aimed at developing a context-aware forwarding model capable of achieving high performance but keeping the overhead low. For such a case, instead of broadcasting a beacon for neighbor discovery, a node leverages the overheard packet from the nodes in its reception range. Instead of dropping unsolicited or looped packets, CMAF processes all received packets to manage the NT. Although this procedure increases the computational processing overhead, it also shows its potential to reduce the overhead, and this is observed when comparing the two modes of operation and the result when the beaconing frequency increases. Moreover, the model is also designed to resort to a mobility prediction scheme for better management of the NT. With a consistent NT, only valid and capable nodes could be chosen as next hops. This improves the performance further.

Furthermore, with CMAF V2I, we intended to investigate the impact on the performance when having further knowledge about the content in the neighbors' CS. Here, given that two beacon broadcast processes take place (i.e., one querying the content producer and the other for querying the 1-hop previous neighbor's CS catalog), the results show that the performance cannot be improved. In addition to the problem caused by the beaconing process, the received list of neighbors' CS cannot be used for a long time. Whenever a node is purged from the NT, tracking it and updating the information in the list with the current node position, for each name prefix, turns out to be infeasible. Moreover, the packet holding the neighbor's CS catalog is sent using BF. The packet carrying the BF Data ends up being an additional source of TO due to its size.

In summary, our main hypothesis was that by incorporating the mobility prediction and leveraging the characteristics of the wireless channel (broadcast by nature), we would be able to reduce the overhead induced by the beaconing process for neighbor discovery and by the retransmissions due to the inconsistency of the NT. And, as shown in Section 4.3, the model achieved the expected result. Moreover, the results show that additional packets in the network for neighbor discovery and neighbor's CS catalog dissemination only increase the TO, degrading the overall performance.

6. Conclusions

We developed CMAF, a context- and mobility-aware forwarding strategy, aiming at reducing transmission overflow while keeping high performance. The proposed protocol works in two modes, V2V (beaconless) and V2I (beacon-based), and in either mode, it outperforms compared state-of-the-art solutions, at the cost of an increased ISD. To further reduce the transmission overhead, a mechanism based on neighborhood density is used to limit the number of hops Data packets traverse from their provider to the consumer. The proposed mechanism still needs refinements to further adapt it with larger differences in node density and for different road topologies, as it shows a consistent effect on the results in both chosen topologies (i.e., the city of Porto's and the Manhattan-like road networks) but with different weights for different densities. In terms of improved performance vs. TO, the proposed model (CMAF V2V) achieves the same performance (i.e., ISR) as that of the flooding scenario (CMAF—flooding), with averages of 15% and 20% TO reductions for the city of Porto's and the Manhattan-like road topologies, respectively.

As previously stated, the model is designed to work in both V2V and V2I modes. CMAF V2V presents superior performance when compared to the state-of-the-art solutions; it presents 5–10% superior ISR than the CCLF for the same overhead, at the cost of 1 s of increased Interest Satisfaction Delay (ISD). Moreover, the number of retransmissions of CMAF is also reduced for relatively the same number of hops for the compared scenarios. Compared to VNDN and Multicast, CMAF presents fewer retransmissions and 10% to 45% superior ISR, with an increased overhead of about 20% in scenarios with 60 to 120 vehicles.

In V2I mode, a controlled beacon is broadcast. When the broadcast frequency increases, the performance degrades, as expected, due to the TO. For a lower beaconing frequency (i.e., one beacon every 10 s), the model still achieves better performance when compared with the state of the art.

7. Future Work

CMAF is designed to be aware of the network density and use this information to define the threshold for the hop count limit. With this logic, we noted that the defined threshold works well and presents a good overhead, except for the scenario with 120 nodes. For instance, increasing the defined threshold, the "performance vs. overhead" metric with a density of 120 nodes improves, but it worsens for the scenario with 80 nodes. For future work, we plan to study and devise a better formulation for defining the threshold that measures the density of vehicles and better limit the hop count. Additionally, we plan to extend the simulation for a denser network and compare with more related solutions. Furthermore, as FIB is not explicitly managed in the present work, we plan to include its management in future work.

Author Contributions: Conceptualization, E.T.d.S., J.M. and A.C.; methodology, E.T.d.S.; software, E.T.d.S.; validation, E.T.d.S., J.M. and A.C.; formal analysis, E.T.d.S.; investigation, E.T.d.S.; resources, E.T.d.S., J.M. and A.C.; data curation, E.T.d.S.; writing—original draft preparation, E.T.d.S.; writing—review and editing, E.T.d.S., J.M. and A.C.; visualization, E.T.d.S.; supervision, J.M. and A.C.; project administration, J.M. and A.C.; funding acquisition, E.T.d.S., J.M. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FCT—Fundação para a Ciência e a Tecnologia grant number PRT/BD/154352/2023.

Data Availability Statement: The data presented in this study are openly available in zenodo.org, at https://zenodo.org/doi/10.5281/zenodo.11135779 accessed on 8 May 2024.

Acknowledgments: This work has been supported by FCT—Fundação para a Ciência e a Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
AP	Access Point
BF	Bloom Filter
BN	Bayesian Networks
BS	Base Station
CCN	Content-Centric Network
CCT	Cached Content Table
CLT	Content Location Table
CMAF	Context and Mobility-Aware Forwarding Model For V-NDN
CN	Content Name
CS	Content Store
DADT	Density-Aware Delay-Tolerant
DIFS	Distributed Interest Forwarder Selection
DM	Data Mining
DR	Dead Reckoning
DRLSF	Decentralized Receiver-based Link Stability-aware Forwarding
EKF	Extended Kalman Filter
ETSI	European Telecommunications Standards Institute
FIB	Forwarding Information Base
GPS	Global Position System
HMM	Hidden Markov Model
ICN	Information-Centric Networking
IP	Internet Protocol
ISD	Interest Satisfaction Delay
ISR	Interest Satisfaction Ratio
ITS	Intelligent Transportation System

KF	Kalman Filter
LBDB	Location-Based Deferred Broadcast
LCE	Leave Copy Everywhere
LoICen	Location-based and Information-Centric
LTMP	Long-Term Mobility Prediction
MANET	Mobile Ad hoc NETwork
MC	Markov Chain
ML	Machine Learning
MM	Markov Model
NACK	Negative Acknowledgment
NDN	Named Data Networking
NDNLP	Named Data Networking Link Protocol
ndnSIM	NDN Simulator
NDO	Named Data Object
NFD	NDN Forwarder Daemon
NMSI	Node Mobility Status Information
NS-3	Network Simulator 3
NT	Neighborhood Table
OBU	On-Board Unit
PF	Particle Filter
PIT	Pending Interest Table
PoI	Point of Interest
P2P	Point-to-Point
QoE	Quality of Experience
QoS	Quality of Service
RoI	Region of Interest
RSU	Road-Side Unit
RTID	Rapid Traffic Information Dissemination
RTT	Round Trip Time
STMP	Short-Term Mobility Prediction
SUMO	Simulation of Urban MObility
SLR	Systematic Literature Review
TCP	Transmission Control Protocol
TO	Transmission Overhead
UKF	Unscented Kalman Filter
VANET	Vehicular Ad hoc NETwork
VCCN	Vehicular Content-Centric Networks
VNDN	Vehicular Named Data Networking
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

References

1. Fujise, M.; Kato, A.; Sato, K.; Harada, H. Intelligent Transport Systems. In *Wireless Communication Technologies: New Multimedia Systems*; Morinaga, N., Kohno, R., Sampei, S., Eds.; Springer: Boston, MA, USA, 2002; pp. 171–200. [CrossRef]
2. Perallos, A.; Hernandez-Jayo, U.; Onieva, E.; Zuazola, I.J.G. *Intelligent Transport Systems: Technologies and Applications*; Wiley: Hoboken, NJ, USA, 2015.
3. Grassi, G.; Pesavento, D.; Wang, L.; Pau, G.; Vuyyuru, R.; Wakikawa, R.; Zhang, L. ACM HotMobile 2013 poster: Vehicular inter-networking via named data. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2013**, *17*, 23–24. [CrossRef]
4. Grassi, G.; Pesavento, D.; Pau, G.; Vuyyuru, R.; Wakikawa, R.; Zhang, L. VANET via Named Data Networking. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 27 April–2 May 2014; pp. 410–415. [CrossRef]
5. Saini, M.; Alelaiwi, A.; Saddik, A.E. How Close Are We to Realizing a Pragmatic VANET Solution? A Meta-Survey. *ACM Comput. Surv.* **2015**, *48*, 1–40. [CrossRef]
6. Khelifi, H.; Luo, S.; Nour, B.; Moungla, H.; Faheem, Y.; Hussain, R.; Ksentini, A. Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 320–351. [CrossRef]
7. Hartenstein, H.; Laberteaux, L.P. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **2008**, *46*, 164–171. [CrossRef]

8. Karagiannis, G.; Altintas, O.; Ekici, E.; Heijenk, G.; Jarupan, B.; Lin, K.; Weil, T. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 584–616. [CrossRef]
9. Tomar, R.; Prateek, M.; Sastry, H. Vehicular Adhoc Network (VANET)—An introduction. *Int. J. Control Theory Appl.* **2016**, *9*, 8883–8888.
10. Misra, S.; Woungang, I.; Misra, S.C. *Guide to Wireless Ad Hoc Networks*, 2009th ed.; Springer Publishing Company: Berlin/Heidelberg, Germany, 2009.
11. Jeong, J. *IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases*; Internet-Draft Draft-Ietf-Ipwave-Vehicular-Networking-13; IETF Secretariat: Wilmington, DE, USA, 2020.
12. Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics* **2022**, *11*, 2162. [CrossRef]
13. Varvello, M.; Rimac, I.; Lee, U.; Greenwald, L.; Hilt, V. On the design of content-centric MANETs. In Proceedings of the 2011 Eighth International Conference on Wireless On-Demand Network Systems and Services, Bardonecchia, Italy, 26–28 January 2011; pp. 1–8. [CrossRef]
14. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking Named Content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 1–4 December 2009; pp. 1–12. [CrossRef]
15. Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.C.; Crowley, P.; Papadopoulos, C.; Wang, L.; Zhang, B. Named data networking. *Comput. Commun. Rev.* **2014**, *44*, 66–73. [CrossRef]
16. Vasilakos, A.; Li, Z.; Simon, G.; You, W. Information centric network: Research challenges and opportunities. *J. Netw. Comput. Appl.* **2015**, *52*, 1–10. [CrossRef]
17. Tariq, A.; Rehman, R.A.; Kim, B. Forwarding Strategies in NDN-Based Wireless Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 68–95. [CrossRef]
18. Yan, Z.; Zeadally, S.; Park, Y. A Novel Vehicular Information Network Architecture Based on Named Data Networking (NDN). *IEEE Internet Things J.* **2014**, *1*, 525–532. [CrossRef]
19. Amadeo, M.; Campolo, C.; Molinaro, A. CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks. *IEEE Commun. Lett.* **2012**, *16*, 1380–1383. [CrossRef]
20. Drira, W.; Filali, F. A Pub/Sub extension to NDN for efficient data collection and dissemination in V2X networks. In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, Sydney, Australia, 19 June 2014; pp. 1–7. [CrossRef]
21. Arnould, G.; Khadraoui, D.; Habbas, Z. A Self-Organizing Content Centric Network Model for Hybrid Vehicular Ad-Hoc Networks. In Proceedings of the First ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, New York, NY, USA, 4 November 2011; pp. 15–22. [CrossRef]
22. De Castro, C.; Raffaelli, C.; Andrisano, O. A dynamic hierarchical VANET architecture for Named Data Networking applications. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 3659–3665. [CrossRef]
23. Bouk, S.H.; Ahmed, S.H.; Kim, D. Vehicular Content Centric Network (VCCN): A Survey and Research Challenges. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, New York, NY, USA, 13–17 April 2015; pp. 695–700. [CrossRef]
24. 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]
25. Rainer, B.; Petscharnig, S. Challenges and Opportunities of Named Data Networking in Vehicle-To-Everything Communication: A Review. *Information* **2018**, *9*, 264. [CrossRef]
26. Kerrche, C.; Ahmad, F.; Elhoseny, M.; Adnane, A.; Ahmad, Z.; Nour, B. Internet of Vehicles over Named Data Networking: Current Status and Future Challenges. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks. Studies in Systems, Decision and Control*; Elhoseny, M., Hassanien, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2019. [CrossRef]
27. Al-Omaisi, H.; Sundararajan, E.A.; Alsaqour, R.; Abdullah, N.F.; Abdelhaq, M. A survey of data dissemination schemes in vehicular named data networking. *Veh. Commun.* **2021**, *30*, 100353. [CrossRef]
28. da Silva, E.T.; Costa, A.L.D.; de Macedo, J.M.H. On the realization of VANET using named data networking: On improvement of VANET using NDN-based routing, caching, and security. *Int. J. Commun. Syst.* **2022**, *35*, e5348. [CrossRef]
29. Silva, E.T.d.; Macedo, J.M.H.; Costa, A.L.D. Context-Aware Routing and Forwarding Model for NDN-Based VANET. In Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Sydney, Australia, 6–7 December 2022; Volume 428, pp. 18–32. [CrossRef]
30. Zafar, W.U.I.; Rehman, M.A.U.; Jabeen, F.; Ghouzali, S.; Rehman, Z.; Abdul, W. Context-Aware Pending Interest Table Management Scheme for NDN-Based VANETs. *Sensors* **2022**, *22*, 4189. [CrossRef]
31. ETSI. *TR 102 638*; ETSI ITS Specification TR 102 638 V1.1.1. Vehicular Communications; Basic Set of Applications; Definitions; Tech Report; European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2009.
32. Ahed, K.; Benamar, M.; Lahcen, A.A.; Ouazzani, R.E. Forwarding strategies in vehicular named data networks: A survey. *J. King Saud-Univ. Comput. Inf. Sci.* **2020**, *34*, 1819–1835. [CrossRef]

33. Talebifard, P.; Amadeo, M.; Campolo, C.; Molinaro, A. Information-centric networking for VANETs. In *Vehicular Ad Hoc Networks: Standards, Solutions, and Research*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 503–524. [CrossRef]
34. Villas, L.A.; Boukerche, A.F.M.; Maia, G.; Pazzi, R.W.N.; Loureiro, A.A.F. DRIVE: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. *Comput. Netw.* **2014**, *75*, 381–394. [CrossRef]
35. Amadeo, M.; Campolo, C.; Molinaro, A. Forwarding strategies in named data wireless ad hoc networks: Design and evaluation. *J. Netw. Comput. Appl.* **2015**, *50*, 148–158. [CrossRef]
36. Wang, J.; Luo, J.; Ran, Y.; Yang, J.; Liu, K.; Guo, S. Towards Predictive Forwarding Strategy in Vehicular Named Data Networking. *IEEE Trans. Veh. Technol.* **2023**, *72*, 3751–3763. [CrossRef]
37. Liu, X.; João Nicolau, M.; Costa, A.; Macedo, J.; Santos, A. A Geographic Opportunistic Forwarding Strategy for Vehicular Named Data Networking. In *Intelligent Distributed Computing IX*; Novais, P., Camacho, D., Analide, C., El Fallah Seghrouchni, A., Badica, C., Eds.; Springer: Cham, Switzerland, 2016; pp. 509–521.
38. Wang, J.; Luo, J.; Zhou, J.; Ran, Y. A Mobility-Predict-based Forwarding Strategy in Vehicular Named Data Networks. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
39. Li, D.; Song, T.; Yang, Y.; Ul, I.R. A reliable and efficient forwarding strategy in vehicular named data networking. *J. Commun. Netw.* **2020**, *22*, 348–357. [CrossRef]
40. Ahmed, S.H.; Bouk, S.H.; Kim, D. RUFS: RobUst Forwarder Selection in Vehicular Content-Centric Networks. *IEEE Commun. Lett.* **2015**, *19*, 1616–1619. [CrossRef]
41. Beri, S.S.; Dutta, N. Evolutionary game based interest forwarding in information centric vehicular networks. *Veh. Commun.* **2024**, *47*, 100779. [CrossRef]
42. Ngo, M.; Ohzahata, S.; Yamamoto, R.; Kato, T. A Visual-Identification Based Forwarding Strategy for Vehicular Named Data Networking. *IEICE Trans. Inf. Syst.* **2023**, *E106.D*, 204–217. [CrossRef]
43. 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, New York, NY, USA, 11 June 2012; pp. 7–12. [CrossRef]
44. Li, Y.; Shi, X.; Lindgren, A.; Hu, Z.; Zhang, P.; Jin, D.; Zhou, Y. Context-Aware Data Dissemination for ICN-Based Vehicular Ad Hoc Networks. *Information* **2018**, *9*, 263. [CrossRef]
45. Kuai, M.; Hong, X.; Yu, Q. Density-Aware Delay-Tolerant Interest Forwarding in Vehicular Named Data Networking. In Proceedings of the 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, 8–21 September 2016; pp. 1–5. [CrossRef]
46. Kuai, M.; Hong, X. Location-Based Deferred Broadcast for Ad-Hoc Named Data Networking. *Future Internet* **2019**, *11*, 139. [CrossRef]
47. Chowdhury, M.; Khan, J.A.; Wang, L. Leveraging Content Connectivity and Location Awareness for Adaptive Forwarding in NDN-Based Mobile Ad Hoc Networks. In Proceedings of the 7th ACM Conference on Information-Centric Networking, New York, NY, USA, 29 September–1 October 2020; pp. 59–69. [CrossRef]
48. Yu, X.; Coutinho, R.W.L.; Boukerche, A.; Loureiro, A.A.F. A distance-based interest forwarding protocol for vehicular information-centric networks. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–5. [CrossRef]
49. Rondon, L.B.; da Costa, J.B.; Filho, G.P.R.; Villas, L.A. A Distance and Position-based Caching Discovery Protocol for Vehicular Named-Data Networks. In Proceedings of the 2019 IEEE Latin-American Conference on Communications (LATINCOM), Salvador, Bahia, 11–13 November 2019; pp. 1–6. [CrossRef]
50. Al-Omaisi, H.; Sundararajan, E.; Alsaqour, R.; Abdulah, N.; Abu Bakar, K.A.; Abdelhaq, M. IDTracS: An Interest-Data-flow tracking-based forwarding scheme for vehicular named data networks. *J. Supercomput.* **2023**, *79*, 16580–16615. [CrossRef]
51. Al-Omaisi, H.; Sundararajan, E.A.; Alsaqour, R.; Abdullah, N.F.; Abu Bakar, K.A. GeoISA: A New Road-Topology-Assisted Geo-Based Content Discovery Scheme for Vehicular Named Data Networking. *Veh. Commun.* **2023**, *40*, 100573. [CrossRef]
52. Zafar, W.U.I.; Rehman, M.A.U.; Jabeen, F.; Kim, B.S.; Rehman, Z. Context-Aware Naming and Forwarding in NDN-Based VANETs. *Sensors* **2021**, *21*, 4629. [CrossRef]
53. Cunha, I.J., Jr.; Fernandez, M.; Patel, A.; Monteiro, M. VNDN-Fuzzy—A strategy to mitigate the forwarding interests broadcast storm problem in VNDN networks. In Proceedings of the 2023 International Conference on Information Networking (ICOIN), Bangkok, Thailand, 11–14 January 2023; pp. 263–270. [CrossRef]
54. Wang, F.; Hou, R. Delayed Broadcast-Based Data Forwarding Strategy for Vehicular Named Data Networks. In Proceedings of the Workshop on Computing, Networking and Communications (CNC), Big Island, HI, USA, 19–22 February 2024.
55. Zafar, W.U.I.; Rehman, M.A.U.; Jabeen, F.; Ullah, R.; Abbas, G.; Khan, A. Decentralized Receiver-based Link Stability-aware Forwarding Scheme for NDN-based VANETs. *Comput. Netw.* **2023**, *236*, 109996. [CrossRef]
56. Ahmed, S.H.; Bouk, S.H.; Yaqub, M.A.; Kim, D.; Song, H. DIFS: Distributed Interest Forwarder Selection in Vehicular Named Data Networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3076–3080. [CrossRef]
57. de Sousa, A.M.; Araújo, F.R.; Sampaio, L.N. A Link-Stability-Based Interest-Forwarding Strategy For Vehicular Named Data Networks. *IEEE Internet Comput.* **2018**, *22*, 16–26. [CrossRef]

58. 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]
59. Boukerche, A.; Coutinho, R.W. LoICen: A novel location-based and information-centric architecture for content distribution in vehicular networks. *Ad Hoc Netw.* **2019**, *93*, 101899. [CrossRef]
60. Prates, A.A.; Bastos, I.V.; Moraes, I.M. GeoZone: An interest-packet forwarding mechanism based on dissemination zone for content-centric vehicular networks. *Comput. Electr. Eng.* **2019**, *73*, 155–166. [CrossRef]
61. Yu, Y.T.; Dilmaghani, R.B.; Calo, S.; Sanadidi, M.Y.; Gerla, M. Interest propagation in named data manets. In Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, USA, 28–31 January 2013; pp. 1118–1122. [CrossRef]
62. Ahmed, S.H.; Bouk, S.H.; Yaqub, M.A.; Kim, D.; Gerla, M. CONET: Controlled data packets propagation in vehicular Named Data Networks. In Proceedings of the 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2016; pp. 620–625. [CrossRef]
63. Braun, T.; Careglio, D.; Matta, I. Vehicular Networking in the Recursive InterNetwork Architecture. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5. [CrossRef]
64. Park, C.M.; Rehman, R.A.; Tran-Dinh, H.; Kim, B.S. Enhanced Protocol for Wireless Content-Centric Network. In Proceedings of the International Conference on Mobile and Wireless Networks (MoWiN), Cairo, Egypt, 11–13 April 2016; Volume 6. [CrossRef]
65. Kato, T.; Minh, N.Q.; Yamamoto, R.; Ohzahata, S. How to Implement NDN MANET over ndnSIM Simulator. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 7–10 December 2018; pp. 451–456. [CrossRef]
66. Kalogeiton, E.; Kolonko, T.; Braun, T. A topology-oblivious routing protocol for NDN-VANETs. *Ann. Telecommun.* **2017**, *73*, 577–587. [CrossRef]
67. Kalogeiton, E.; Braun, T. Infrastructure-Assisted Communication for NDN-VANETs. In Proceedings of the 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Chania, Greece, 12–15 June 2018; pp. 1–10. [CrossRef]
68. Martinez, F.J.; Fogue, M.; Toh, C.K.; Cano, J.C.; Calafate, C.T.; Manzoni, P. Computer Simulations of VANETs Using Realistic City Topologies. *Wirel. Pers. Commun.* **2013**, *69*, 639–663. [CrossRef]
69. Ullah, R.; Atif Ur Rehman, M.; Kim, B.S. Hierarchical Name-Based Mechanism for Push-Data Broadcast Control in Information-Centric Multihop Wireless Networks. *Sensors* **2019**, *19*, 3034. [CrossRef]
70. Majeed, M.F.; Ahmed, S.H.; Dailey, M.N. Enabling Push-Based Critical Data Forwarding in Vehicular Named Data Networks. *IEEE Commun. Lett.* **2017**, *21*, 873–876. [CrossRef]
71. Amadeo, M.; Campolo, C.; Molinaro, A.; Ruggeri, G. Content-centric wireless networking: A survey. *Comput. Netw.* **2014**, *72*, 1–13. [CrossRef]
72. Tavares, M.; Aponte, O.; Mendes, P. Named-Data Emergency Network Services. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, New York, NY, USA, 10–15 June 2018; p. 516. [CrossRef]
73. Grassi, G.; Pesavento, D.; Pau, G.; Zhang, L.; Fdida, S. Navigo: Interest forwarding by geolocations in vehicular Named Data Networking. In Proceedings of the 2015 IEEE 16th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Los Alamitos, CA, USA, 14–17 June 2015; pp. 1–10. [CrossRef]
74. Rowstron, A.; Pau, G. *Characteristics of a Vehicular Network*; Technical Report; Computer Science Department, The University of California Los Angeles: Los Angeles, CA, USA, 2009.
75. Duan, M.; Zhang, C.; Li, Y.; Xu, W.; Ji, X.; Liu, B. Neighbor Cache Explore Routing Protocol for VANET based on Trajectory Prediction. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 October 2018; pp. 771–776.
76. Amadeo, M.; Campolo, C.; Ruggeri, G.; Lia, G.; Molinaro, A. Caching Transient Contents in Vehicular Named Data Networking: A Performance Analysis. *Sensors* **2020**, *20*, 1985. [CrossRef]
77. 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]
78. Li, Y.; Jin, D.; Hui, P.; Wang, Z.; Chen, S. Limits of Predictability for Large-Scale Urban Vehicular Mobility. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2671–2682. [CrossRef]
79. Abdel-Halim, I.; Fahmy, H. Prediction-based Protocols for Vehicular Ad Hoc Networks: Survey and Taxonomy. *Comput. Netw.* **2017**, *130*, 34–50. [CrossRef]
80. Zhang, H.; Dai, L. Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications. *IEEE Access* **2019**, *7*, 802–822. [CrossRef]
81. Denko, M.K. Mobility Prediction Schemes in Wireless Ad Hoc Networks. In *Advanced Wired and Wireless Networks*; Wysocki, T.A., Dadej, A., Wysocki, B.J., Eds.; Springer: Boston, MA, USA, 2005; pp. 171–186. [CrossRef]
82. Cheng, C.; Jain, R.; van den Berg, E. Location Prediction Algorithms for Mobile Wireless Systems. In *Wireless Internet Handbook: Technologies, Standards, and Application*; CRC Press, Inc.: Boca Raton, FL, USA, 2003; pp. 245–263.
83. Haerri, J.; Bonnet, C.; Filali, F. *The Challenges of Predicting Mobility*; Technical Report EURECOM+2240; Eurecom: Biot, France, 2006.

84. Agarwal, A.; Das, S.R. Dead reckoning in mobile ad hoc networks. In Proceedings of the 2003 IEEE Wireless Communications and Networking, 2003 WCNC 2003, New Orleans, LA, USA, 16–20 March 2003; Volume 3, pp. 1838–1843. [CrossRef]
85. Balico, L.N.; Oliveira, H.A.B.F.; Souza, E.L.; Pazzi, R.W.; Nakamura, E.F. On the performance of localization prediction methods for vehicular Ad Hoc Networks. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 359–364. [CrossRef]
86. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
87. Meyn, S.; Tweedie, R. *Markov Chains and Stochastic Stability*; Cambridge University Press: Cambridge, UK, 2009. [CrossRef]
88. Gambs, S.; Killijian, M.O.; del Prado Cortez, M.N.N. Next Place Prediction Using Mobility Markov Chains. In Proceedings of the First Workshop on Measurement, Privacy, and Mobility, Bern, Switzerland, 10–13 April 2012. [CrossRef]
89. Balico, L.N.; Loureiro, A.A.F.; Nakamura, E.F.; Barreto, R.S.; Pazzi, R.W.; Oliveira, H.A.B.F. Localization Prediction in Vehicular Ad Hoc Networks. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2784–2803. [CrossRef]
90. Zhang, H.; Hua, Y.; Wang, C.; Li, R.; Zhao, Z. Deep Learning Based Traffic and Mobility Prediction. In *Machine Learning for Future Wireless Communications*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2020; Chapter 7, pp. 119–136. [CrossRef]
91. Ibnkahla, M. Applications of neural networks to digital communications-survey. *Signal Process.* **2000**, *80*, 1185–1215. [CrossRef]
92. Chakrabarti, S.; Ester, M.; Fayyad, U.; Gehrke, J.; Han, J.; Morishita, S.; Piatetsky-Shapiro, G.; Wang, W. *Data Mining Curriculum: A Proposal (Version 1.0)*; Technical Report, Intensive Working Group of ACM SIGKDD Curriculum Committee; ACM: New York, NY, USA, 2006.
93. Sayed, A.H.; Djurić, P.M.; Hlawatsch, F. Chapter 6—Distributed Kalman and Particle Filtering. In *Cooperative and Graph Signal Processing*; Djurić, P.M., Richard, C., Eds.; Academic Press: Cambridge, MA, USA, 2018; pp. 169–207. [CrossRef]
94. Aljeri, N.; Boukerche, A. Performance evaluation of movement prediction techniques for vehicular networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]
95. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
96. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina: Chapel Hill, NC, USA, 1995.
97. Julier, S.; Uhlmann, J. *New Extension of the Kalman Filter to Nonlinear Systems*; SPIE: Bellingham, WA, USA, 1997.
98. Zaki, S.M.; Ngadi, M.; Razak, S.; Kamat, M.; Mohamad Sharif, J. A Location Based Prediction Service Protocol for VANET City Environment. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 6811–6836.
99. Feng, H.; Liu, C.; Shu, Y.; Yang, O. Location Prediction of Vehicles in VANETs Using A Kalman Filter. *Wirel. Pers. Commun.* **2014**, *80*, 543–559. [CrossRef]
100. Understanding Kalman Filters, 02. Available online: https://www.youtube.com/playlist?list=PLn8PRpmsu08pzi6EMiYnR-076Mh-q3tWr (accessed on 15 March 2023).
101. Afanasyev, A.; Shi, J.; Zhang, B.; Zhang, L.; Moiseenko, I.; Yu, Y.; Shang, W.; Li, Y.; Mastorakis, S.; Huang, Y.; et al. *NFD Developer's Guide*; Technical Report; Named Data Networking (NDN): Los Angeles, CA, USA, 2018. [CrossRef]
102. Frederick, R.; Casner, S.L.; Jacobson, V.; Schulzrinne, H. *RTP: A Transport Protocol for Real-Time Applications*; RFC 1889; Internet Society: Reston, VA, USA, 1996. [CrossRef]
103. Mastorakis, S.; Afanasyev, A.; Zhang, L. On the Evolution of ndnSIM: An Open-Source Simulator for NDN Experimentation. *ACM Comput. Commun. Rev.* **2017**, *47*, 19–33. [CrossRef]
104. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic Traffic Simulation using SUMO. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2575–2582.
105. NS-3. NS-3: Network Simulator. 2022. Available online: https://ndnsim.net/2.7/intro.html (accessed on 31 October 2022).
106. Campanile, L.; Gribaudo, M.; Iacono, M.; Marulli, F.; Mastroianni, M. Computer Network Simulation with ns-3: A Systematic Literature Review. *Electronics* **2020**, *9*, 272. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.