A Simulation Framework for Supporting Vehicular Knowledge Networks

Ali Nadar, Jérôme Härri
EURECOM, 450 route des Chappes
06904 Sophia-Antipolis, France
{ali.nadar, haerri}@eurecom.fr

Mohammad Irfan Khan, Seyhan Uçar, Onur Altintas
InfoTech Labs, Toyota Motor North America R&D
Mountain View, CA, USA
{mohammad.irfan.khan, seyhan.ucar, onur.altintas}@toyota.com

***Abstract*—Vehicular Knowledge Networking (VKN) is a paradigm where vehicles exchange knowledge instead of data. Named Data Networking (NDN) is a resilient architecture for sharing data between vehicles without information about the hosting vehicle. NDN might not be adapted for sharing knowledge, as the peculiar complexity of knowledge requires knowledge-driven NDN functions as well as a specific simulation environment enabling joint knowledge perception, inference and reasoning. In this paper, we present an open-source co-simulation framework connecting NDN, traffic, dynamic control and perception simulators for knowledge perception & inference, and with an AI-as-a-Service (AIaaS) platform for knowledge reasoning. This paper notably describes new interfaces and functions between the NDN daemon and the AIaaS micro-services for handling interest naming, caching and forwarding between knowledge producers and consumers. We demonstrate the benefit of the proposed framework and knowledge-specific extensions through an AI-driven vehicular intersection management.**

***Index Terms*—Vehicular Knowledge Networks, Named Data Networking (NDN), AI models, simulation, ns-3, SUMO, CARLA.**

I. Introduction

The rapid advancement of Cooperative Connected Automated mobility (CCAM) and Cooperative Intelligent Transportation Systems (C-ITS) has spurred interest in efficient vehicular networking paradigms. Vehicular Ad-hoc Networks (VANETs) — self-organizing wireless networks between vehicles - have been developed to share data between vehicles in stringent vehicle-to-vehicle (V2V) environments. VANETs are optimized to share data between known endpoints (address or geographic location). Named Data Networking (NDN), as a content-centric approach, provides a promising alternative by shifting the focus from endpoint to content names. By identifying data by its name rather than by its location, this paradigm is particularly well suited for vehicular data exchange in highly dynamic environments.

In parallel, the rise of Artificial Intelligence-as-a-Service (AIaaS) platforms has democratized access to advanced machine learning and AI capabilities, offering scalable and cost-effective solutions for deploying and managing AI models. As described in Nadar et al. [1], AIaaS platforms leverage cloud infrastructure and sophisticated semantic reasoning mechanisms to provide on-demand access to AI services, harnessing the power of AI without the need for extensive infrastructure.

With increasing AI capabilities, vehicles are increasingly capable of generating knowledge e.g. AI models. Vehicular Knowledge Networks (VKN) [2] are therefore a novel paradigm where vehicles exchange knowledge rather than data. Several studies demonstrated the pertinence of VKN for CCAM or C-ITS, such as vehicular risk assessment [3], content placement for vehicular micro-clouds [4], or decentralized machine learning orchestration [5]. Harnessing AIaaS functionalities to NDN protocols could be an enabler to fulfill the promises of VKN.

In the context of CCAM or C-ITS, AI models play a critical role in enabling adaptive decision-making processes, such as collision avoidance, dynamic route planning, and autonomous vehicle control. However, the exchange of large AI models between vehicles in a VKN poses unique challenges. The traditional NDN architecture is not optimized for handling large data objects, such as AI models, due to limitations in caching and forwarding strategies. Existing NDN research indeed primarily focuses on the exchange of smaller, sensor-based data or traffic-related information [6], leaving a gap in the efficient management of larger, knowledge-driven content such as AI models.

To address these challenges, we propose NDN4VKN, a novel framework designed to support the efficient exchange of AI models in VKN. Our approach builds on NDN and integrates a distributed AIaaS platform, originally conceived as a centralized system, into a decentralized vehicular environment. This integration enhances the functionality of core NDN components such as the Pending Interest Table (PIT), Content Store (CS), and Forwarding Strategy, optimizing them for knowledge-centric applications in dynamic vehicular scenarios.

In this work, we also introduce a multi-layered simulation environment that synchronizes ndnSim[1] for NDN protocols and network-level simulations, SUMO[2] for vehicular mobility modeling, and CARLA[3] for high-fidelity AI training and perception tasks. The proposed framework not only facilitates the exchange of AI models between vehicles but also dynamically adapts to changing vehicular contexts, improving the overall efficiency of vehicular knowledge networking.

Our contributions are threefold: First, we describe the architectural integration between the NDN stack and the AIaaS platform creating the NDN4VKN platform; second we introduce

[1]https://ndnsim.net
[2]https://eclipse.dev/sumo/
[3]https://carla.readthedocs.io/en/stable/

our methodology for building our co-simulation framework facilitated by TraCI (Traffic Control Interface). Finally, we demonstrate the feasibility of AI/ML model exchanges within NDN4VKN. The platform is available as open-source on EURECOM Gitlab: *https://gitlab.eurecom.fr/cats/ndn4vkn* and is developed as a multi-Docker container platform.

The rest of this paper is organized as follows: Section II discusses related work in NDN. Section III outlines the proposed NDN4VKN architecture. Section IV details the knowledge-driven NDN extensions. In Section V, we illustrates a proof-of-concept of the NDN4VKN framework. In Section VI we conclude and shed light on future direction in NDN and VKN.

II. Literature Survey

NDN is defined as a future internet architecture by the IETF Internet Research Task Force (IRTF) and is investigated by a dynamic community. The principles and challenges of NDN are well described by Saxena *et al.* [7], with specific details on NDN forwarding challenges by Farhan *et al.* [8] or NDN naming conventions by Nurhayati *et al.* [9]. Vehicular networking being a major use case for NDN, significant work has been pursued towards Vehicular NDN as surveyed by Khelifi *et al.* [6].

In particular, at the forwarding table level, *Jaebeom et al.* [10] proposed a TOPology aware CCN protocol (TOP-CCN) that relies on Multiple Point Relay (MPR) based packet flooding, where Publisher MPR (PMPR) merge content announcements, and flooding is restricted by hop counts and MPR nodes to reduce multiple content announcement. *Kato, et al.* [11] depicted implementation details of an NDN MANET over the ndnSIM simulator as well as its performance in ad-hoc networking. At the PIT management level, *Zafar, et al.* [12] introduced a dynamic and context-aware strategy for managing the PIT in Vehicular NDN architectures. The primary contribution of the paper lies in addressing the scalability and efficiency challenges associated with PIT management in highly dynamic vehicular environments. The paper however did not consider the impact of complex NDN naming typically observed in VKN.

Campolo *et al.* [13] were among the first to evaluate the feasibility and implication of NDN for AI orchestration, followed by a more recent study by Hail *et al.* [14]. However, they remained at a conceptual level and did not address the impact of semantic reasoning and knowledge graph description of modern AI models.

From a simulation environment perspective, the ndnSim[4] platform connects and adapts an NDN software library (NDN Fowarding, CS and PIT) with ns-3, which emulates nodes and wireless links. ndnSim has therefore been a valuable evaluation platform for NDN research. When investigating Vehicular NDN, realistic mobility is missing. *Arajo et al.* [15] therefore presented *NDN for Inter-Vehicle Communication (NDN4IVC)*, interconnecting ndnSim and the traffic simulator SUMO. When applying NDN for CCAM or VKN, additional features are required, such as realistic sensor perception, vehicular control and knowledge reasoning.

[4]https://ndnsim.net/current/

In [1] *Nadar et al.* proposed a centralized AI-as-a-Service architecture (AIaaS) called *Infrastructure-Assisted Knowledge Management (IAKM)*, highlighting the potential of delivering AI models as services through a cloud-based system. This centralized design enabled efficient deployment, management, and access to AI models, ensuring scalability and ease of maintenance within a single infrastructure. The authors demonstrated how AI models could be effectively semantically named, and shared between multiple AI actors.

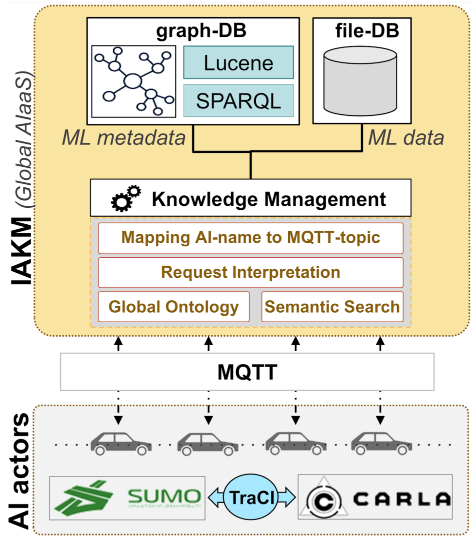

Figure 1: IAKM - Centralized aspect

However, as depicted in Fig. 1, the centralized nature of this architecture introduces certain limitations, especially in vehicular domains which demand low latency, high throughput, and the ability to scale in a more dynamic, geographically distributed manner. Accordingly, leveraging edge AI and NDN to bring networking and computing decisions closer to vehicles drives the development of *distributed* AIaaS platforms, enhancing both the performance and scalability of decentralized AI services..

III. NDN4VKN

In addition to NDN protocols and vehicular mobility, VKN requires sensor perception, vehicular control as well as knowledge reasoning functions. Table I presents VKN functional requirements compared with two leading simulation platforms (ndnSIM and NDN4IVC [15]) as well as the proposed framework.

	VKN Functions	**ndnSIM**	**NDN4IVC**	**NDN4VKN**
Modules	Network simulator:	ns-3	ns-3	ns-3
	Networking	NDN	NDN	NDN
	Mobility simulator	N/A	SUMO	SUMO + **CARLA**
	Driving Control	N/A	N/A	**ROS + PID**
	Perception	N/A	N/A	**radar, LIDAR, camera**
	Map	N/A	SD-Map	**HD-Map**
NDN	Interest naming	uri	uri	**SPAQL**
	Interest matching	Exact	Exact	**Knowledge Reasoning**
	CS Storage policy:	Rand,LRU LFU, FIFO	Rand,LRU LFU, FIFO	**Knowledge Graph**
	Content type:	String object	String object	**AI/ML model**
	Knowledge	N/A	N/A	**Share, Train, Infer**

Table I: NDN4VKN vs Current Simulators

NDN4VKN interfaces key modules and simulators to support VKN as depicted in Fig. 2. The first key module is ndnSim, which interfaces the network simulator ns-3 with ndn-cxx, which provides key NDN primitives (NFD and application layer APIs). NFD is a key module supporting NDN core functions, including the Content Store (CS), Forwarding Information Base (FIB), and Pending Interest Table (PIT). Within ndnSim, ns-3 enables large-scale NDN simulations, but ndn-cxx itself is a software library supporting experimental studies. The second key module, is the traffic simulator SUMO modeling vehicular mobility and providing realistic traffic scenarios. ndnSim and SUMO are interconnected via the TraCI interface.

While the first two described modules are similar to NDN4IVC, NDN4VKN adds two critical modules. The first is CARLA, an open-source simulator for autonomous driving perception and control. It supports flexible specification of sensor suites and realistic environmental conditions through HD-Maps. CARLA is connected to SUMO and ndnSim via TraCI to share perception and environmental data to the NDN application layer for knowledge training. Knowledge is used by CARLA for inference on sensor, environmental or control functions. Accordingly, a dedicated interface has been designed to connect CARLA to the NDN application layer for sharing knowledge or inferring knowledge from CARLA data.

The second module is the AIaaS platform IAKM. As ndn-cxx, IAKM is not a simulator but a software library providing key functionalities for knowledge reasoning. Within NDN4VKN, the IAKM is directly connected to the NVF, with dedicated APIs to the NDN CS, PIT and FIB. Within ndnSim, the IAKM contains one instance of each ns-3 node in order to be available at each NDN entity (producer, consumer and router).

Through the described NDN4VKM framework, AI agents (as application layer entities) do not need to request data via NDN to train a model, but can share AI models via NDN with selected AI agents closest to the data required for training. AI agents may also query other AI agents via NDN for knowledge related to a particular unknown environment. And through CARLA and the IAKM libraries, NDN4VKN allows evaluating the true knowledge accuracy in a particular context. NDN4VKN therefore provides primitives and a flexible simulation environment for VKN.

Figure 3 shows such multi-layer and multi-hop features of VKN and the role played by the various NDN4VKN modules or NDN functions.

IV. Knowledge Reasoning for NDN

In this section, we describe the key NDN extensions for Knowledge reasoning provided by the interconnection with the IAKM.

A. From Knowledge to Interest Naming

As described in [2], knowledge naming is critical to VKN. Knowledge names correspond to metadata describing their meaning and usability context. In [1], knowledge is described and identified by a Knowledge Graph (KG), such as an RDF graph. This graph structure is better suited than domain-specific hierarchical NDN naming as it allows flexibility in the naming structure and support inter-domain reasoning. While KG may be described by JSON-LD objects, mapping must be provided to respect the structure of NDN interest packets.

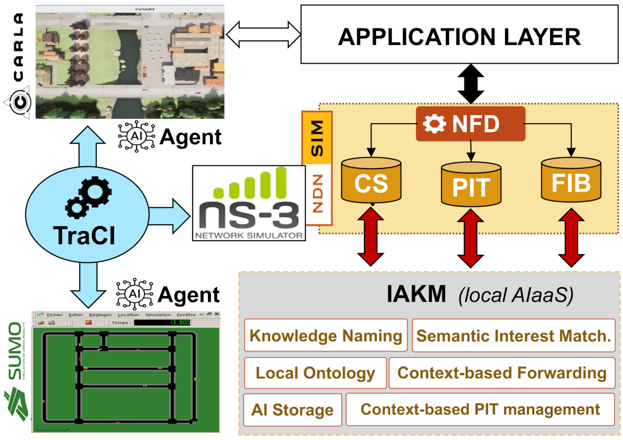

Figure 2: NDN4VKN: Framework overview

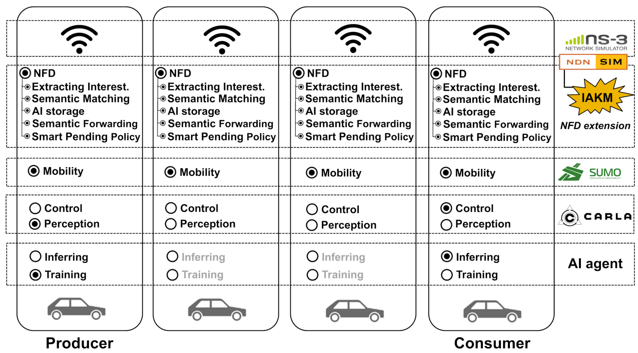

Figure 3: Multi-layers VKN framework in NDN

Nadar et al. proposed in [1] a method to convert JSON-LD described KG into MQTT topics by extracting full key paths from the JSON structure of the KW. While this method is straightforward, it significantly increases the name size, as depicted in Fig.4ⓐ, where the name size reaches 245 bytes. We introduce an alternative mechanism that reduces the AI name size in NDN Interests. The proposed method, which is implemented by an IAKM *knowledge translation* microservice and illustrated in Fig.4ⓑ, produces a more compact name of 182 bytes by using *_S_* and *_E_* to denote the beginning and end of nested objects, respectively. This optimization improves efficiency in resource-constrained vehicular knowledge networking environments.

NDN4VKN provides a flexible API between AI agents in the application layer and the IAKM *knowledge translation* microservice to translate a knowledge name into a NDN interest. This IAKM microservice is also used to identify a given knowledge by its NDN interest.

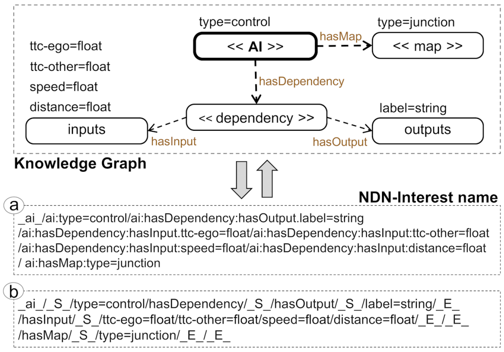

Figure 4: Bidirectional mapping of AI metadata in the Knowledge Graph with NDN Interest (Serialization/ Deserialization)

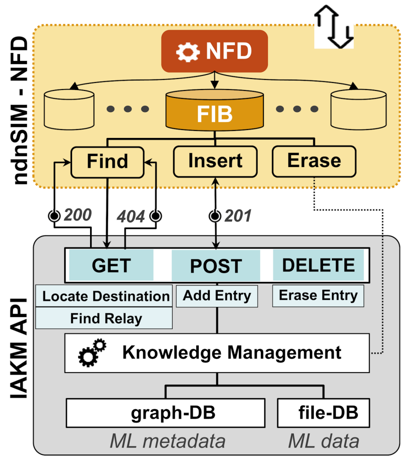

Figure 5: Extending NDN FIB with IAKM

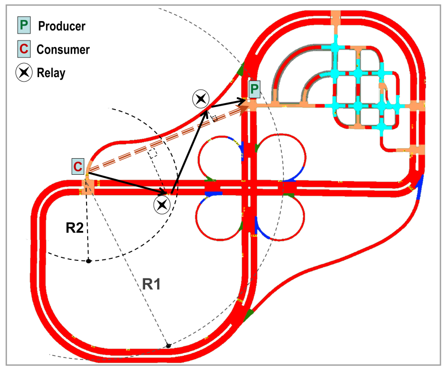

Figure 6: GeoRouting implementation in ndnSIM

B. Forwarding Strategy: Semantic Geo-Routing

In Named Data Networking (NDN), forwarding strategies are pivotal for determining how Interest packets are routed through the network to retrieve the desired data packets. These strategies directly affect the network's performance, including its scalability, efficiency, and ability to manage resources effectively. However, the existing forwarding strategies have limitations in handling dynamic vehicular networks, where mobility and context-awareness are crucial. In [10] *Jaebeom, et al.* proposed a Topology aware CCN protocol (Top-CCN) to reduce multiple content announcement overhead and broadcast storming problem. This proposal is technically a proactive state-full ad-hoc routing protocol, which is not efficient in highly dynamic vehicular networks. Stateless approaches, such as geo-routing are better suited to select optimal relays assuming the presence of a location service to obtain the geolocation of the destination node. NDN4VKN provides both mechanisms through the interaction between the NDN FIB and the IAKM as depicted in Fig. 5. First, a *semantic location service* leverage KG descriptions to match a knowledge usability context and HD-Map topology. As an example, knowledge suited to T-intersections should be located there and accordingly through HD-Map reasoning, the *semantic location service* will extract the GPS coordinates of the most suitable T-intersection in the area. Second, default a greedy forwarding mechanisms is defined at the FIB to select the relay node providing the best geographical progress toward the destination.

Fig. 6 illustrates a traffic scenario where a car identified as *C* is approaching a complex T-intersection and requires knowledge to handle it. Vehicle C queries its HD-Map for semantically T-shape intersection within a *R1* range similar to the required knowledge. Once obtained by the IAKM, it selects the most optimal geo-relay within a *R2* range providing the maximum geographical progress. Knowledge reasoning plays here a critical role, as not any T-shape intersection match the usability context of the required knowledge. Through knowledge reasoning, the IAKM is able to identify the mostly likely one.

C. Semantic-Based Interest Matching

Given the strict NDN naming structure, NDN interest matching is limited to quasi-exact matches. Knowledge bearing more subtle features requires a semantic proximity metric to describe how semantically close one piece of knowledge is to a target interest. NDN4VKN introduces an enhanced matching mechanism through our semantic API platform. Traditional NDN relies on exact matches between Interests and Data, which can be restrictive due to the variability in how Interests are expressed across different nodes. A semantic analysis that transcends exact component matching is presented in [1], which we interconnect with NDN in NDN4VKN . By understanding the underlying meaning within Interests, the IAKM Context-Alignment-Function (CAF) microservice facilitates a more flexible and context-aware retrieval process that uses Natural Language Processing (NLP) to align the large language expressions with IAKM ontology. This approach promotes more efficient and relevant data retrieval, aligning with the semantic intent of Interests rather than their precise specifications.

For example, consider a scenario where a vehicle A sends an Interest regarding a roundabout with the specifics "4 legs" and "30 meters radius", expressed as /ai/model/map/roundabout/legs/4/radius/30. If the Content Store (CS) of Vehicle B holds data with a name like /ai/model/map/round-intersection/exits/4/width/30, which semantically corresponds to the same information, the traditional exact match mechanism would fail to retrieve the data because of the mismatch in Interest naming. In contrast, the NDN4VKN provides API between the IAKM and an AI agent at the application layer to perform a semantic-based reasoning between the knowledge name and the Interest description.

D. Towards Context-Aware Content Store Management

The Content Store (CS) acts as a cache that temporarily holds data packets to facilitate NDN content retrieval without querying the original producer. CS management policies in NDN like Least-Recently-Used (LRU), First-In-First-Out (FIFO) or Random, rely on static parameters like access frequency and recency. When considering knowledge, it becomes more challenging. For example, let's consider three pieces of the same knowledge (e.g. same IA model): one trained in a context corresponding to an T-intersection A, and one trained in a context corresponding to a T-intersection B bearing 80% similarity with T-intersection A. Storing both models might significantly impact the capacity of the CS and reasoning would be required to assess the interoperability of knowledge for T-intersections A and B, keeping only one in CS. Similarly, when receiving a third undertrained knowledge with 60% similarity with the T-intersection A, the CS might decide to cache it in addition to the existing knowledge due to its higher accuracy but lower proximity. In the end, the CS ends up with three copies of the same knowledge, whereas traditional NDN would only keep one.

In NDN4VKN, the IAKM optimizes the CS by relying on semantic reasoning on the KG representation of the Interest and KG describing knowledge stored on the CS. The NDN4VKN dedicated APIs supporting knowledge reasoning are depicted on Fig. 7 and briefly described next.

- *CS Find function* - a GET method triggering semantic reasoning on availability of a given knowledge in the CS and returning knowledge semantically closest to the Interest.
- *CS Insert function* - a POST pushing knowledge into the CS and triggering IAKM semantic reasoning on duplicate content.
- *CS Delete function* - triggers IAKM semantic reasoning on the pertinence of a given knowledge in the CS.

NDN4VKN provides a default implementation of CS knowledge reasoning for VKN. NDN4VKN is yet a toolbox to extend CS management for knowledge networking.

E. Context-aware PIT management

The Pending Information Table (PIT) is a key player in finding content in Named Data Networking (NDN). Due to the limited opportunities offered by current memory technologies, PIT size is a bottleneck. The stored PIT Entry (PITE) is removed either when the PIT Entry Lifetime (PEL) expires or the vehicle with PITE receives the required Data packet. When considering knowledge, the situation is more complicated. Taking the same example as before, assuming that an existing PITE corresponds to knowledge training for T-intersection A and that a new Interest packet is received for knowledge corresponding to a T-intersection B, bearing 60% similarities with A. Given the context similarities, should the vehicle

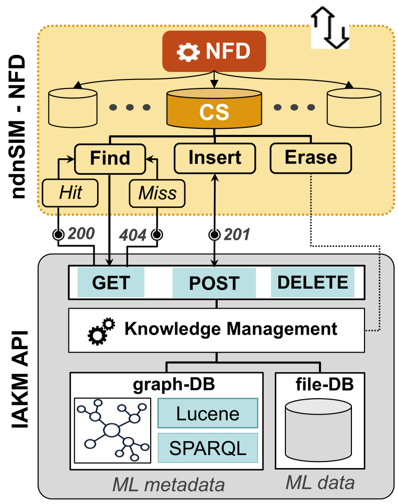

Figure 7: Extending NDN CS with IAKM

assume that the current PIT entry is sufficient, or should it store both entries, potentially overloading the PIT table? A second challenge occurs when the vehicle receives knowledge and needs to assess whether to remove the corresponding PIT entry. Assuming a vehicle receives knowledge corresponding to T-intersection C bearing 80% similarity with A and trained at 40%, would a PITE corresponding to the T-intersection A be satisfied and therefore deleted, or should the vehicle hope to get better knowledge later and keep the PITE? A third challenge is PIT priority avoiding PIT overflow. In case one PITE needs to be removed, how should the vehicle choose between the three different PITE entries previously described, and how to compare with PITE for different knowledge?

NDN4VKN interfaces the NFD PIT entity with the IAKM via three APIs described in Fig. 8 and briefly described below.

- *PIT Proximity Match* - a GET method triggering semantic reasoning on the semantic proximity between an Interest packet and available PITE.
- *PIT Add Entry* - a POST method creating a new PITE and triggering IAKM semantic reasoning on duplicate PITE entries.
- *PIT Erase function* - triggers IAKM semantic reasoning on PITE prioritization in the PIT.

Related to the GET method, NDN4VKN additionally proposes a *ProximityMatch* function as an alternative to the existing ndnSim *ExactMatch* method, as depicted in Fig. 9. The NDF PIT entity provides a *find* function to match an Interest packet with a PIT entry and a *lookup* function to remove PIT entries upon corresponding knowledge reception. NDN4VKN connects the NFD PIT entity to the IAKM via a GET method and a *ProximityMatch* function. The latter function applies knowledge reasoning on PIT entries, Interests and Knowledge by merging their KG description and computing a proximity metric. A threshold is defined to decide if the PIT find/lookup is successful or not. NDN4VKN finally also defines a multi-parameter PIT prioritization, where the *ProximityMatch* is one of them, but also extracts from the PIT KG description the related application description. For instance, a safety-related PITE would bear a higher priority compared to a traffic efficiency PITE, and within a similar class, a PITE entry with a higher proximity metric would bear a higher priority.

NDN4VKN provides a default KG proximity metric, priority threshold and multi-parameter PIT priority function. NDN4VKN also provides a toolbox to extend PIT management for VKN with different values or functions.

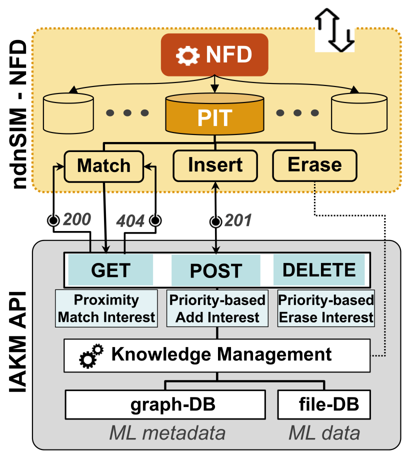

Figure 8: Extending NDN PIT with IAKM

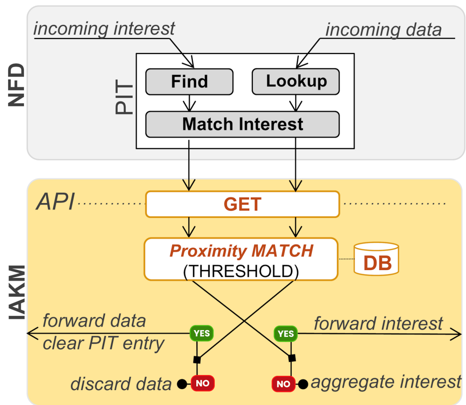

Figure 9: Proximity-based matching in PIT

V. Experimentation: Synchronization among Platforms

In this section, we illustrate how NDN4VKN can be beneficial to support VKN. In this proof-of-concept, we aim at improving the efficiency of knowledge caching in the Content Store (CS) and enabling PIT dynamic management to optimize communication in a vehicular environment. To achieve a comprehensive toolset for Vehicular Knowledge Networking (VKN), we simulate vehicular mobility using the SUMO simulator, while AI model training and inference are performed using the CARLA simulator. Although CARLA is utilized for more granular perception and AI-based vehicle control in this study, the proposed toolset is not restricted to CARLA; AI models can be trained or inferred using data from other simulators, such as SUMO or any external application. All modules described in Fig. 2 are interconnected and controlled via a ndnSim scenario, and a close to reality vehicular scenario involving mobility, control and perception is reproduced. Without loss of generality, Knowledge takes the shape of an AI model.

A. *Experiment setup*

The simulation environment is composed of several components, each designed to replicate a specific aspect of vehicular networking, mobility, information perception and AI processing. Below are the key components and configuration:

- **SUMO -** it is configured to model an urban block including a T-Intersection.
- **CARLA -** it is configured to closely model the T-shaped intersection as an HD-Map as well as all vehicles approaching. Vehicles are synchronized with SUMO vehicles. Each CARLA vehicle is equipped with a sensor suite (cameras, LIDAR). Vehicular sensors as well as the HD-Map enables NDN4VKN to model vehicular control based on AI inference or AI training.
- **ndnSIM -** used for NDN network layer, it models the wireless link and communication channel between vehicles, and provides NDN primities connected to the IAKM.
- **IAKM -** used for any reasoning over knowledge at NDN functions (CS, PIT, FIB) required during the simulation.

The simulation was conducted in a Linux environment, utilizing Ubuntu 20.04.6 as the operating system. The following simulator versions were employed: ndnSIM v2.9, SUMO 1.19, and CARLA 0.9.12. Table II details the specific setup and configurations used in ndnSIM components to facilitate the exchange of AI models in network packets. Without loss of generality, this proof-of-concept is based on WiFi V2X, as 5G NR SL is not available in the current version of ndnSIM. Additionally, modifications were made to the MSDU (MAC Service Data Unit) aggregation settings to accommodate the substantial size of ML models. We leveraged the Aggregated-MSDU (A-MSDU) feature, which packs multiple MSDUs into a single MPDU (MAC Protocol Data Unit), reducing protocol overhead and enhancing throughput.

Table II: Platform Configuration

Component	**Configuration**
Network Device	IEEE 802.11p
Nb. of Producer/Consumer	1/1
Propagation Loss Model	Range propagation loss model
WiFi max-queue-size	50k p
Vehicle Radio Range $(_{max})$	200 m
Wifi max-queue-delay	50 s
MTU size	8096
LP- MaxFragments	10k
Reassembly Timeout	5 s
Language (CXX)	C++ (17)
Routing Protocol	OLSR, Semantic-georouting
Content Store Size	optimized-IAKM (no-limit)
Simulation time	120 s

B. Coupling Network and Mobility Simulators

To realistically capture the dynamic nature of vehicular networks, we couple SUMO with ndnSIM by synchronizing the mobility events from SUMO with the network operations simulated in ndnSIM. Every vehicle in SUMO corresponds to a node in ndnSIM, where vehicles generate Interest packets for AI models based on their surrounding context (e.g., safety alerts, traffic conditions, navigation). The vehicle's location, speed, and trajectory in SUMO directly influence the propagation of Interest packets within ndnSIM, allowing to evaluate the impact of vehicular mobility on NDN performance. In the designed scenario, a vehicle approaching a roundabout in SUMO generates an Interest for AI model data related to managing that intersection, which is then disseminated through the NDN-based vehicular network. This coupling is also available in the NDN4IVC [15] platform.

C. Synchronizing SUMO with CARLA

To realistically model driving control and sensor perception, we connect the simulator CARLA to SUMO via TraCI, so that vehicles in CARLA are synchronized with SUMO. Sharing TraCI, ndnSIM may therefore also interact with CARLA to train or infer AI models exchanged via NDN leveraging CARLA HD-Map and vehicular sensor suites. The synchronization between CARLA and SUMO provides hierarchical granularity of the simulation, where a large-scale urban environment is simulated by SUMO, whereas only a few vehicles in a target intersection are simulated by CARLA. The large-scale SUMO scenario allows, in turn large-scale NDN networking to observe the impact of AI model dissemination on NDN CS, PIT and FIB entities.

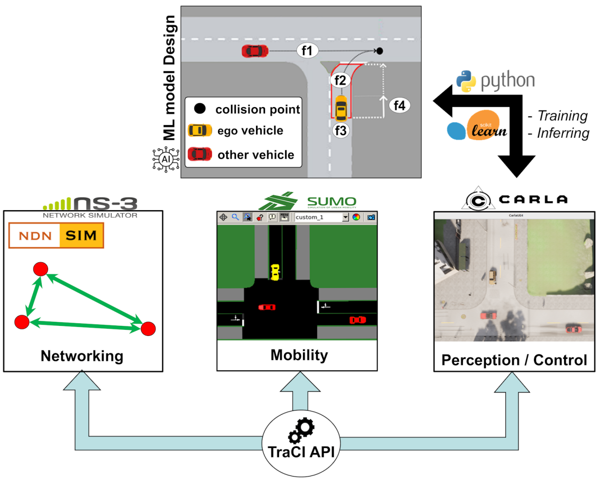

Figure 10: TraCI-Based Co-Synchronization

D. Exchanging AI models over VKN

Without loss of generality, we defined an urban environment with two vehicles, one producer and consumer. The producer trains an AI model based on its driving dynamics on a target intersection. The consumer requests such AI model for AI inference on vehicular control safely driving through the intersection. We finally included 23 additional vehicles on SUMO in order to simulate traffic in the urban area. As AI model, we used a Random Forest Classifier (RFC) due to its robust ensemble learning capabilities, and which is designed to decide if a vehicle should enter a T-intersection based on its perception of the dynamics of other vehicles. The producer has an empty AI model and drives through the T-intersection using CARLA auto-pilot, names the model according to the intersection context and start training the model. Once completed, the AI model is saved in the producer CS store. From SUMO trajectory planning, the consumer identifies that it needs to cross through the target T-intersection. Accordingly, it extracts from SUMO the semantic description of the T-intersection and builds a AI Intersect packet, which is disseminated via NDN. Upon reception of the AI model, the consumer may proceed through the intersection. If it does not receive the model on time, it will simply stop at the intersection.

The proposed AI model [1] aims to avoid collision. Accordingly, as shown in Fig. 10, we identify a Collision-Spot in the road waypoints where the paths of the ego and the target vehicles intersect, and then we define a control area (in red) where the ego vehicle must exercise caution, specifically by according priority to the target vehicle. Building upon this identification, we can precisely define four key features that form the foundation of the AI model:

- **f1**: Time-To-Collision of '*target*' vehicle. Using speed and distance metrics, we determine the time interval before the ego vehicle reaches the identified collision point.
- **f2**: Time-To-Collision for '*ego*' vehicle. Using speed and distance metrics, we determine the time interval before the ego vehicle reaches the identified collision point.
- **f3**: speed of the ego vehicle
- **f4**: normalized float distance in [0,1] interval for '*ego*' vehicle within the *red* control area.

While the ego vehicle is in the control area, data collection is performed at each time step to capture the 4 input features and the control tag. As output label for RFC model, we pick up the label according to the mapping table defined in Table III using the pair of (throttle & brake) of actual dynamic control. Once the ego vehicle leaves the control area, the ego vehicle trains its ML model using the collected dataset where the combination of features helps the model learn patterns and make accurate predictions.

The outcome of the proof-of-concept is that the consumer managed to pass through the T-intersection, demonstrating the following aspects of NDN4VKN: (i) the producer managed to train a AI model; (ii) the AI model was correctly named and reasoned by the IAKM to correctly deliver it to the consumer; (iii) the consumer received the AI model before reaching the intersection, demonstrating efficient NDN dissemination; (iv) the consumer's driving dynamics were correctly applied by CARLA to travel through the intersection.

VI. Conclusion and Future Work

In this paper, we present a knowledge-driven simulation framework for vehicular knowledge networking. We first introduced an architectural integration between the key components

Dynamic Control		
Throttle	**Break**	**Label (decision)**
$0.0 < t \leq 0.1$	0	t1
$0.1 <$ t ≤ 0.2	0	t2
.....	.....	.....
$0.6 <$ t ≤ 0.7	0	t7
0	0	n
0	$0.0 <$ b ≤ 0.1	b1
0	$0.1 <$ b ≤ 0.2	b2
.....	.....	.....
0	$0.6 <$ b ≤ 0.7	b7

*RFC model has 15 labels: 7/7 levels for throttle/break and neutral

Table III: Mapping control (throttle,break) to decision

of the Named-Data-Networking (NDN) stack and an AI-as-a-Service (AIaaS) platform to support knowledge reasoning in NDN. We then presented a co-simulation framework interfacing our knowledge-driven NDN architecture with state-of-the-art simulators modeling vehicular mobility, sensor, and control for vehicular knowledge perception and inference. We finally applied the proposed knowledge-driven simulation framework in a proof-of-concept, successfully exchanging AI models between vehicles - AI trainer (Producer) and AI inferring (Consumer) - through ad-hoc communication within an NDN-based network.

Released in open-source to the community, the knowledge-driven simulation framework is expected to be a key enabler to support research in Vehicular Knowledge Networking (VKN), providing missing bricks from current state-of-the-art NDN simulation platforms, such as knowledge reasoning or integrated knowledge perception and inference.

In our future work, we intend to integrate and evaluate the impact of advanced knowledge-driven mechanisms in NDN core components in large-scale scenarios. We also plan to explore the integration of additional AI capabilities and perform real-world validations.

VII. Acknowledgements

This work has benefited from a government grant managed by the "*Agence Nationale de la Recherche*" under the France 2030 program with NF-FITNESS, grant "ANR-22-PEFT-0007".

References

[1] A. Nadar and J. Härri, "An ai-as-a-service platform for an artificial intelligence of things (aiot)," in *VTC 2024, IEEE Vehicular Technology Conference, 24-27 June 2024, Singapore*, 2024.

[2] D. Deveaux, T. Higuchi, S. Uçar, J. Härri, and O. Altintas, "A definition and framework for vehicular knowledge networking: An application of knowledge-centric networking," *IEEE Vehicular Technology Magazine*, vol. 16, no. 2, pp. 57–67, 2021.

[3] ——, "A knowledge networking approach for ai-driven roundabout risk assessment," in *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2022, pp. 1–8.

[4] S. Uçar, T. Higuchi, C.-H. Wang, D. Deveaux, O. Altintas, and J. Härri, "Vehicular knowledge networking and mobility-aware smart knowledge placement," in *CCNC 2022, 19th Anual IEEE Consumer Communications & Networking Conference, 8-11 January 2022, Las Vegas, USA (Virtual Conference)*, IEEE, Ed., Las Vegas, 2022.

[5] D. J. Deveaux, T. Higuchi, S. Ucar, C.-H. Wang, J. Härri, and O. Altintas, "On the orchestration of federated learning through vehicular knowledge networking," in *VNC 2020, IEEE Vehicular Networking Conference, 16-18 December 2020, Virtual Conference*, IEEE, Ed., New-York, 2020.

[6] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *Commun. Surveys Tuts.*, vol. 22, no. 1, p. 320–351, Jan. 2020.

[7] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey," *Computer Science Review*, vol. 19, pp. 15–55, 2016.

[8] F. Ahmed, A. Aman, R. Hassan, K. Nisar, and M. Uddin, "Named data networking: A survey on routing strategies," *IEEE Access*, pp. 1–9, 01 2022.

[9] A. Nurhayati, R. Mayasari, S. Ahdan, R. M. Negara, G. N. Nurkahfi, and N. R. Syambas, "Naming scheme on named data networking: A survey," in *2022 8th International Conference on Wireless and Telematics (ICWT)*, 2022, pp. 1–5.

[10] J. Kim, D. Shin, and Y.-B. Ko, "Top-ccn: Topology aware content centric networking for mobile ad hoc networks," in *2013 19th IEEE International Conference on Networks (ICON)*, 2013, pp. 1–6.

[11] T. Kato, N. Q. Minh, R. Yamamoto, and S. Ohzahata, "How to implement ndn manet over ndnsim simulator," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 451–456.

[12] W. Zafar, M. Atif Ur Rehman, F. Jabeen, S. Ghouzali, Z. Rehman, and W. Abdul, "Context-aware pending interest table management scheme for ndn-based vanets," *Sensors*, vol. 22, p. 4189, 05 2022.

[13] C. Campolo, G. Lia, M. Amadeo, G. Ruggeri, A. Iera, and A. Molinaro, "Towards named ai networking: Unveiling the potential of ndn for edge ai," in *Ad-Hoc, Mobile, and Wireless Networks: 19th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2020, Bari, Italy, October 19–21, 2020, Proceedings*. Springer-Verlag, 2020, p. 16–22.

[14] M. Hail, A. Bin Salem, and W. Munassar, "Ai for iot-ndn: Enhancing iot with named data networking and artificial intelligence," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems*, 01 2024, pp. 1020–1026.

[15] G. B. Araújo, M. L. M. Peixoto, and L. N. Sampaio, "Ndn4ivc: A framework for simulating and testing of applications in vehicular named data networking," *Comput. Networks*, vol. 235, p. 109949, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235727615