A Definition and Framework for Vehicular Knowledge Networking

Duncan Deveaux†, Takamasa Higuchi‡, Seyhan Uçar‡, Jérôme Härri†, Onur Altintas‡

†EURECOM, Campus SophiaTech, 450 route des Chappes, 06904 Sophia-Antipolis, France
E-mail: {deveaux,haerri}@eurecom.fr
‡InfoTech Labs, Toyota Motor North America R&D, Mountain View CA, USA
E-mail: {takamasa.higuchi,seyhan.ucar,onur.altintas}@toyota.com

***Abstract*—To operate intelligent vehicular applications such as automated driving, machine learning, artificial intelligence and other mechanisms are used to abstract from information what is commonly referred to as knowledge. Defined as a state of understanding obtained through experience and analysis of collected information, knowledge is promising for vehicular applications. However, it lacks a unified framework to be cooperatively created and shared to achieve its full potential. This paper investigates on the meaning and scope of knowledge applied to vehicular networks and defines a structure for vehicular knowledge description, storage, and sharing. Through the example of passenger comfort-based automated driving, it exposes the potential benefits for network load and delay of such knowledge structuring.**

***Index Terms*—Framework, knowledge, networking, vehicular.**

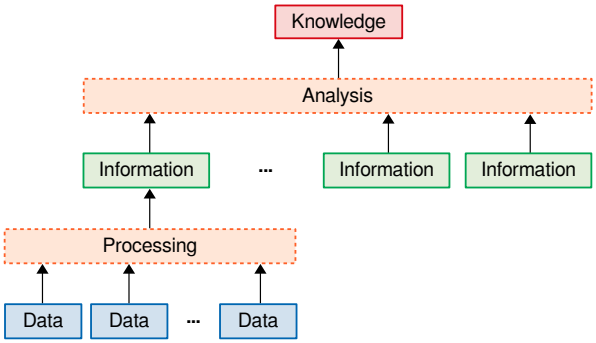

Fig. 1: The Relationship between Data, Information and Knowledge

I. Introduction

OVER the last decade, we have witnessed the evolution of vehicular networking from 'Vehicular Ad-Hoc Networks' (VANETs), enabling spontaneous direct communications between vehicles, to 'Connected Vehicles' generalizing information exchange among vehicles and infrastructure. Vehicular networking has been developed as an enabler of innovative applications intended to improve traffic safety, reduce congestion, and even provide infotainment on-board. Early applications were designed to only provide information to drivers, delegating any decision making to them. However, in recent ambitious applications, such as automated driving or platooning, simple information treatment and forwarding mechanisms are not sufficient anymore. Instead, decision-making is based on models of the environment built from much more sizable sets of input information. Models are designed to learn from experience rather than react to static input signals. In this context, models have the potential to reduce the load and delay in vehicular networks, as key content is extracted from larger sets of input information. What is more, by favoring the distribution of models over static content, information privacy is improved. However, unlike with static content sharing, mechanisms to name, localize, and network/offload knowledge creation capacities of models in vehicular networks are lacking.

The existing literature in the information science domain covers conceptual definitions of data, information, and knowledge [1]. In this paper, for the sake of clarity, we make similar distinctions among these categories. As shown in Figure 1, the most fundamental element is data, that we define as an atomic value with a unit, e.g., (30kph). Next, information is built by aggregating pieces of data that describe a situation, e.g., (17:00, 30kph) a vehicle's speed at a given time. On top of information lies knowledge, which describes general patterns and relationships obtained through the analysis of sets of information. For example, clustering or classification algorithms can be used to extract hidden relationships within a set. As an example, (17:00, 30kph) $\Rightarrow$ `SCHOOL_RUSH_HOUR` is knowledge associating a time and speed information with a context, i.e., the end of the school day.

Various techniques, such as Artificial Intelligence (AI), Machine Learning (ML), or Formal Language (FL) have been used to extract knowledge in vehicular contexts through the analysis of various sources of information. For example, Ruta *et al.* [2] composed *knowledge* to recognize a high level context of driving by using sensor information from multiple cars in a common geographical area. Qi, Wang *et al.* [3] applied both learning algorithms and edge computing units offloading to provide optimal caching of high level connected driving services to vehicles, including image auto annotation or locally relevant recommendations yielding. Khan *et al.* [4] applied deep learning to learn the transmission patterns of neighboring vehicles and paved the way for fewer packet collisions.

Regardless of the technique, extracting *knowledge* from information is a complex and expensive process, and the generated *knowledge* may be beneficial to other vehicles.

So far each vehicle remains autonomous for its *knowledge* building, which requires highly specialized algorithms and a large amount of input information, potentially sourced from multiple different vehicles. This can be seen as a significant overhead considering that *knowledge* can be shared and not individually recreated. As a reaction, research has recently been focused on defining a knowledge-centric approach to networking, where information would not be the main focus anymore. Instead, knowledge would be created by nodes in the network, and directly stored and shared among them. Wu *et al.* [5] described the concept of a knowledge-centric networking framework, separated into three building blocks: knowledge creation, composition and distribution. A literature survey on means of creating and distributing knowledge is performed. However, the concept of knowledge remains abstract and its implementation or format is left for future work.

The contributions of this paper are as follows: Vehicular Knowledge Networking (VKN), a knowledge-centric networking framework applied to vehicular networks, is presented. It defines a common architecture for knowledge description, needed for subsequent storage, composition, and exchange with other connected vehicles. As such, VKN is a framework that makes performance improvements in other applications possible. In a passenger comfort-based rerouting application, we evaluate the load impact of VKN knowledge distribution in vehicular networks compared with ICN-based approaches, and show an overhead reduction of 14 to 40% depending on the network topology, through cooperative knowledge building and sharing.

The rest of this article is organized as follows: Section II introduces information treatment standards and defines the scope of *knowledge* in vehicular networks. Based on this understanding, Section III describes a structure for knowledge description, storage and distribution. In Section IV, an application of the concept shows potential load and delay improvement for the network. Section V finally points out the potential research applicability behind VKN, while Section VI summarizes the article.

II. Vehicular Information and Vehicular Knowledge

In this section, we first describe the current forms of information in vehicular networks, as well as various standards for information storing and sharing. Then, we build on this understanding to define a format for vehicular knowledge representation.

A. Information in Vehicular Networks

Nodes of the vehicular network may exchange diverse types of information, including but not limited to:

- Safety notifications, e.g., accidents or road condition.
- Vehicle state information and sensor measurements.
- Navigation information, e.g., maps, road or parking data.
- Information on topics such as weather or traffic flow.
- Road-related information, e.g., gas station opening times.
- Multimedia contents for user infotainment.

In ETSI standards, the storage of information in connected vehicles is performed inside the *Local Dynamic Map (LDM)* information base, which is divided into four layers:

1. Permanent static data, i.e., map data.
2. Temporary static data, i.e., roadside infrastructure.
3. Temporary dynamic data, e.g., roadblock, signal phase.
4. Highly Dynamic data, e.g., vehicles, pedestrians.

The LDM provides a standard approach for storing information, but not for naming it. Generally, any information may be stored in the LDM as long as it is labeled with a space-time area of relevance. This can lead to a lack of interoperability between the contents generated by different providers. To tackle this issue, semantic standards have been developed to provide nodes with a "common language" and avoid redundancy. For example, the Vehicle Signal Specification (VSS) and ontology [6], provides a standard way to address the state of vehicle components, e.g., steering wheels or window opening. Moreover, standard safety messages such as the Cooperative Awareness Message (CAM) and the Decentralized Environmental Notification Message (DENM) in Europe or the Basic Safety Message (BSM) in the US have been defined to describe various types of information and events.

Finally, after it has been sensed and stored, information is spread within the vehicular network. In most vehicular applications, as it describes road events, the information itself is more valuable than its source. Routing algorithms have thus been developed that focus on the information being shared rather than its host. Information-Centric Networking (ICN) is a networking paradigm that may be suitable for some vehicular applications [7]. Rather than sending a request to a specific host, a request is disseminated to fetch a specific information identified by a unique content name.

B. Knowledge Networking in Vehicular Networks

Compared with information, knowledge is condensed while maintaining reusability across different contexts. As such, we envision a shift away from the information-based architecture to the benefit of knowledge in future vehicular networks.

We summarize related works on vehicular knowledge networking. The concept of *Knowledge-Centric Networking* (KCN) was introduced in [5]. As with information, mechanisms must be developed to create, compose, store and distribute knowledge. Applications have been studied which use the concepts of KCN. In [8], knowledge is created as a ML model predicting the video playback pattern of users. The knowledge is sent to edge servers to optimize video caching. In [9], statistical knowledge about network topology is used to optimize routing in unmanned aerial vehicle fleets.

While the concepts of KCN have been described in [5] and applied for specific applications in works such as [8, 9], no formal structures for knowledge representation or protocols for knowledge distribution have been described. The lack of a generic framework to describe, store, and share knowledge prevents interoperable applications of KCN and hinders the potential of vehicular knowledge networking.

In [10], knowledge is expressed and composed as deep learning models. In this paper, we formalize a KCN framework

and present structures to define and describe a generic form of knowledge, not specialized in deep learning models. Based on knowledge description, we introduce protocols to cooperatively create, exchange and localize knowledge following the dynamically evolving needs of vehicles.

C. Vehicular Knowledge Representation

We understand knowledge as an abstract content obtained from the analysis of larger sets of information [1]. Knowledge can be extracted from information using ML algorithms, divided into three classes. Supervised learning applies to classification or regression. A model is trained based on a number of samples of the form: (information, class) for classification – or (information, value) for regression. Knowledge is extracted as the relationship between the information and its associated class, i.e., the function that takes information as an input and returns its estimated class. Unsupervised learning extracts clusters of similar items in a set of information. It creates knowledge by exposing relationships among information items and sorting them into different clusters. Reinforcement learning, finally, can be used by an agent to learn the optimal behavior to adopt in a context of interaction with an environment to maximize a user-defined reward.

A trained ML model is a piece of knowledge, able to return synthetic knowledge from input information. The knowledge that is extracted through learning techniques can be further leveraged through knowledge composition methods where existing knowledge is further analyzed/collated to produce new knowledge. For instance, if a user wants to avoid traffic congestion, the system needs to first detect the congested zones and decompose the necessary factors including current location, destination, and estimated route/arrival time based on the current traffic. In this case, in addition to knowledge creation, the knowledge composition also collates some information and/or other knowledge, such as closed roads and/or construction zones, so as not to exacerbate the congestion.

Thus, the word *knowledge* can refer to both: (i) algorithms able to synthesize sets of information into pieces of knowledge, that we refer to as *knowledge models*, and (ii) abstracted information obtained by applying models, that we refer to as *knowledge samples*. Both aspects should be considered as we describe a formalization of knowledge definition, storage and distribution in vehicular networks.

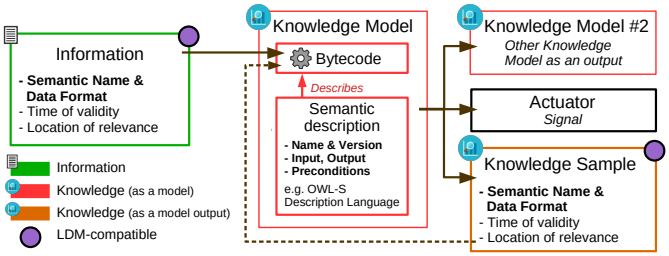

Fig. 2: The Vehicular Knowledge Ecosystem

Figure 2 illustrates when and how knowledge is handled in vehicular networks for safety and driving-related applications. On the left of the figure, information is meant to be stored in the LDM and has a time and area of validity. For the sake of interoperability with other vehicles, we consider it to be named and structured following well-known constraints, e.g., following the VSS specification.

A *knowledge model*, typically implemented as a trained ML algorithm, takes information as input and produces output knowledge. In Figure 2, we distinguish two aspects of a knowledge model: semantic description and bytecode. The semantic description is used to describe the unique name, version, necessary input, produced output and the potential preconditions necessary to apply the model. We define the *bytecode* of a model as the executable file that produces an output from a well-formed set of input information. We identify three possible outputs to a model. The model may output another knowledge model, parameterized by the inputs of the original model. Alternatively, a knowledge model may output an actuation signal, or a *knowledge sample* that, like the information used to produce it, has a time and area of validity.

While it is abstract and obtained through analysis, a *knowledge sample* is structured similarly to information. As a consequence, it can be fed as input to another knowledge model, generating new composed knowledge.

Figure 3 shows an application of this definition of knowledge related to the estimation of passenger comfort onboard a Highly-Automated Vehicle (HAV). The top of the figure describes a knowledge model `model.env_comfort` able to infer a value of passenger comfort from three road-related inputs: traffic conditions, visibility and two-wheelers concentration. Then, in the bottom of the Figure, `model.fetch_driving_behavior` takes generic contextual information as input, namely the obtained comfort level and the given town of application of the model. Based on this input, it tailors the parameters of a personalized output knowledge model to provide real-time driving assistance to the ego vehicle, optimized depending on the requested comfort level and town. We will come back to the illustrated models in Section IV, as we describe an application of VKN.

III. Aspects of Vehicular Knowledge Networking

Technologies such as vehicular clouds, fog or edge computing provide support for knowledge storage and distribution within vehicular networks [11]. Nevertheless, a challenge to achieve knowledge networking is related to identification, naming and localization of knowledge. For example, upon reaching a new city, e.g., in a foreign country where human drivers behave differently, a HAV might need knowledge about how to drive there. It is a complex task as:

- The requested knowledge should be identified and named. For example, should the requested knowledge involve urban driving, intersections only, handling two-wheelers?
- Then, it should be located: It is not straightforward to determine who owns the knowledge. It could be located within a vehicular cloud or edge unit. Due to the dynamic aspect of vehicular networks as well as optimizations in knowledge caching [12], a knowledge discovery or subscription mechanism is required.

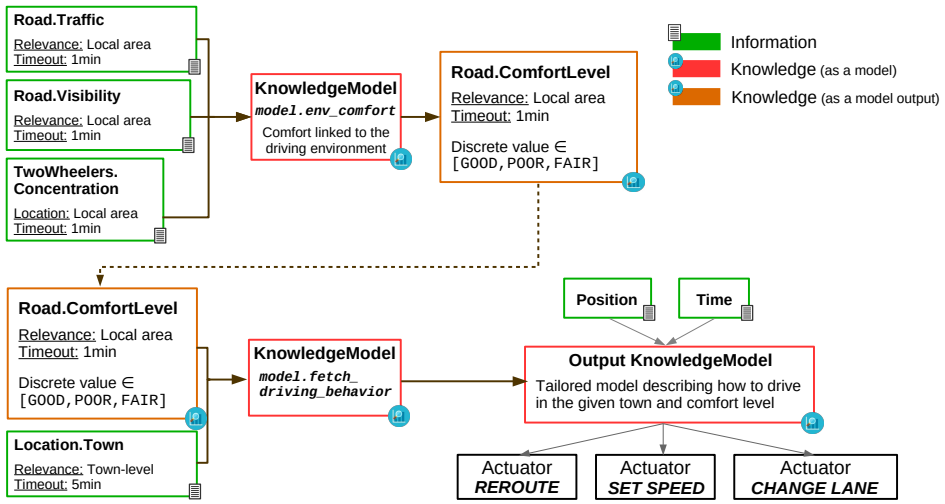

Fig. 3: Knowledge Models for Passenger Comfort Handling

Vehicular clouds and edge computing units are supporting technologies that enable efficient knowledge sharing and storage. VKN as a framework supports knowledge identification, cooperative creation and localization mechanisms. As such, it can be implemented on top of such architectures to enable efficient knowledge networking in vehicular networks.

A. Knowledge Description & Storage

The creation of knowledge in vehicular networks takes place at two levels:

- Using ML algorithms, automakers and organizations train and provide models capable of generating knowledge from a set of input (information or *knowledge samples*).
- The models are applied to real input, returning new *knowledge samples* or *models*.

In some cases, keeping models proprietary can be intentional by automakers to protect their competitiveness in the market. However, in other cases, as considered in [10], a lack of cooperation in knowledge model building may lead to an inefficient use of resources. Similar knowledge models are likely to be independently trained by competing entities, leading to redundant computations. Nevertheless, no common format to describe the input, output, and preconditions of a model are provided by training entities, preventing the cooperative use of knowledge models by a larger number of nodes.

To tackle these issues, we separate the semantic description of a model from its actual execution place, as shown in Figure 2. A knowledge model description formally states (i) a unique name and version code for the model, and (ii) the input, output and preconditions to its application. It is a lightweight content, shareable with multiple nodes. The knowledge model's bytecode is the executable file performing the knowledge samples creation. Even if a vehicle is not in possession of a model's bytecode, it may request knowledge creation from another node following the constraints detailed in the model description. In addition to using unique names and version codes for each model, knowledge synchronization

```
<AtomicProcess ID="model.env_comfort:1.1">
  <hasInput resource="#traffic" />
  <hasInput resource="#visibility" />
  <hasInput resource="#twoWheelers" />
</AtomicProcess>

<Input ID="traffic">
  <parameterType resource="#Road.Traffic" />
</Input>
<Input ID="visibility">
  <parameterType resource="#Road.Visibility" />
</Input>
<Input ID="twoWheelers">
  <parameterType resource="#TwoWheelers.
    Concentration" />
</Input>
<Output ID="comfort">
  <parameterType resource="#Road.ComfortLevel" />
</Output>
```

Fig. 4: Comfort Model Description Structure in OWL-S

mechanisms should be defined to ensure that two remote nodes have the same understanding of a piece of knowledge. This aspect is further detailed in Section III-B3.

As a requirement for model description, the inputs and outputs of a model should be named according to standard semantics specifications such as VSS [6]. By consulting the specification for the name associated with each input or output, nodes are able to deduce the format of information and *knowledge samples* required to apply the model. Moreover, preconditions to the model application may be set, e.g., limited to a given town.

A candidate model description language matching these requirements is OWL-S [13]. It was originally developed for automatic Web Services discovery, composition and invocation. The *process model* standard of OWL-S provides a means of description for the set of input, output and preconditions of a model. Figure 4 provides an example of a OWL-S description of the `model.env_comfort` model introduced in Figure 3.

As part of VKN, we separate the storage of models' descriptions and bytecodes. Figure 5 illustrates the on-board unit of a connected vehicle. The facilities layer contains the LDM, able

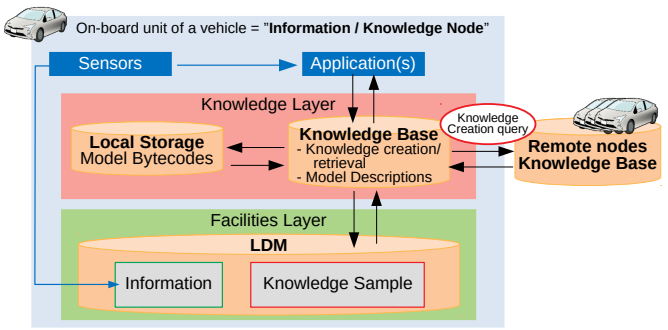

Fig. 5: On-Board Storage of Knowledge and Information

to store information as well as knowledge samples obtained from abstraction. As part of VKN, we add a knowledge layer as an interface between applications and information storage:

- In a Knowledge Base (KB), a list of known knowledge model descriptions is stored.
- A local storage in the knowledge layer may store knowledge model bytecodes. The stored bytecodes are independent from the model descriptions stored in the KB.

As illustrated by Figure 5, the KB is connected both with the ego vehicle's on-board local model storage and remotely with the KBs of other vehicles. As such, it is responsible for the orchestration of knowledge creation in vehicular networks. To create knowledge, access to both a model's bytecode and input is needed. If relevant input is stored locally in the LDM, the KB can obtain the model's bytecode with the right version code through local storage if available, or by requesting a remote KB. Another option to perform knowledge creation, especially if no relevant input is locally available, is to forward a request of remote knowledge creation to another vehicle which possesses relevant input. The remote vehicle can then in turn issue a request for the required model, if not locally stored. The modalities of such knowledge sharing requests are described in Section III-B. This allows for a flexible framework, able to expand or limit the creation and distribution of knowledge within a certain group of vehicles as needed. It is also responsible for knowledge synchronization between nodes.

B. Knowledge Distribution

As we separate knowledge models' bytecodes and their description, a structural need appears for the distribution of knowledge. Nodes of the vehicular network are interconnected and knowledge creation may be the product of a cooperation between multiple nodes.

We consider two contexts for knowledge distribution within vehicular networks:

1. The distribution of knowledge models, able to produce new knowledge samples.
2. The distribution of knowledge creation capacities, as a means of outsourcing knowledge sample creation to remote vehicles.

As an example, we consider a vehicle v about to drive in an unknown environment, e.g., to cross a new city. VKN should allow respectively two types of knowledge sharing for v. It may: (i) request knowledge directly as a model advising a driving behavior based on the local context, or (ii) request the generation of a *knowledge sample* by a remote vehicle, e.g., the generation of the estimated driving comfort based on the local context.

1) Knowledge Models Request: We define protocols to request and retrieve knowledge models in vehicular networks. Models may be requested based on: (i) their names, if known by the requester, or (ii) their input and output parameters, to discover unknown knowledge.

The issued request takes the following pseudocode form:

1. `REQUEST MODEL name: [model_name]`
   `CONTEXT version >= 1.0,`
   `last_update: [time]`
2. `REQUEST MODEL output: Road.ComfortLevel`
   `CONTEXT last_update: [time],`
   `spatial_relevance: [location]`

The request is distributed in the vehicular network following a process that we describe next. A response to the issued request by a remote vehicle could take the form:

1. `REQUESTED MODEL name: [model_name]`
   `CONTEXT version >= 1.0,`
   `last_update: [time]`
   `RESPONSE bytecode: [bytecode]`
2. `REQUESTED MODEL`
   `output: Road.ComfortLevel`
   `CONTEXT last_update: [time],`
   `spatial_relevance: [location]`
   `RESPONSE name:[model.env_comfort:1.1],`
   `model_description:[input, outputs...],`
   `bytecode:[bytecode]`

2) Knowledge Application Request: Then, vehicles may request the remote application of a knowledge model within a given context to retrieve knowledge samples without performing local input data collection and computation. We give an example request and response for the remote creation of knowledge about the driving comfort level in a distant area:

- `APPLY model.env_comfort:1.1`
  `IN [location]`
- `COMPUTED model.end_comfort:1.1`
  `IN [location] BY [node_address]`
  `RESULT Road.ComfortLevel: FAIR`

3) Knowledge Routing and Synchronization: To route knowledge requests in vehicular networks, we extend existing Content Centric Networking (CCN) interests-based routing mechanisms. CCN is an implementation of the ICN paradigm.

When a vehicle v issues a knowledge creation request:

1. The request is transmitted to an initial selection of remote nodes.
2. When receiving a knowledge creation request, a remote node checks (i) whether it owns a model matching the request and (ii) whether the context faced by the remote node matches the request.
3. If the conditions are matched, the remote node computes and returns the requested knowledge to v. Otherwise, the request is further transmitted to another neighboring node.

Finally, content synchronization mechanisms implemented as part of ICN, as surveyed in [14] can be adapted for knowledge. Models could be divided between (i) ready-trained

models, whose synchronization should follow described mechanisms, and (ii) models being trained, where no fixed version of the model is distributed. In that case, synchronization could take the age of the last contribution to the model into account. Similarly, transmission delay constraints are challenging and should be addressed through caching mechanisms, e.g., using vehicular clouds [11].

IV. Application: Passenger Comfort-based Driving

As an application of VKN and to show potential benefits for vehicular networks, we investigate on the use case of *passenger comfort-based automated driving*. We describe a simple model using rule-based semantics as an example. However, the VKN framework supports a greater complexity on knowledge definition and its composition as, e.g., ML algorithms. Then, we evaluate the overhead associated with comfort knowledge distribution for both ICN-based and VKN approaches.

A. Comfort Knowledge Models

As an example, we define a simplified knowledge model `model.env_comfort` to determine the level of *passenger comfort* in a given area, as introduced in the top of Figure 3. The model reads a set of area-related input with semantically defined names:

1. The current traffic conditions, $tr \in$ `Road.Traffic` $\equiv$ [`FLUID`, `CONGESTED`].
2. The visibility in the area, $v \in$ `Road.Visibility` $\equiv$ [`CLEAR`, `OBSTRUCTED`].
3. The concentration of two-wheelers in the surroundings, $c_{tw} \in$ `TwoWheelers.Concentration` $\equiv$ [`HIGH`, `MEDIUM`, `LOW`].

The model outputs a discrete qualification of the level of comfort associated with driving in the area, as $cft \in$ `Road.ComfortLevel` $\equiv$ [`GOOD`, `FAIR`, `POOR`]. While Algorithm 1 provides a simple pseudocode implementation of the model as an example, the framework also supports complex models, e.g., ML for realistic applications.

Algorithm 1 Simplified algorithm to compute comfort from environmental parameters

Input
$c_{tw} \in$ [`HIGH`, `MEDIUM`, `LOW`]
$v \in$ [`CLEAR`, `OBSTRUCTED`]
$tr \in$ [`FLUID`, `CONGESTED`]
Output
$cft \in$ [`GOOD`, `FAIR`, `POOR`]

```
if c_tw = LOW and v = CLEAR and tr = FLUID then
    cft ← GOOD
else if c_tw = HIGH then
    cft ← POOR
else
    cft ← FAIR
end if
```

Moreover, as detailed in the bottom of Figure 3, the `model.fetch_driving_behavior` model returns a ML trained model on how to adapt driving behavior in the input town and comfort level.

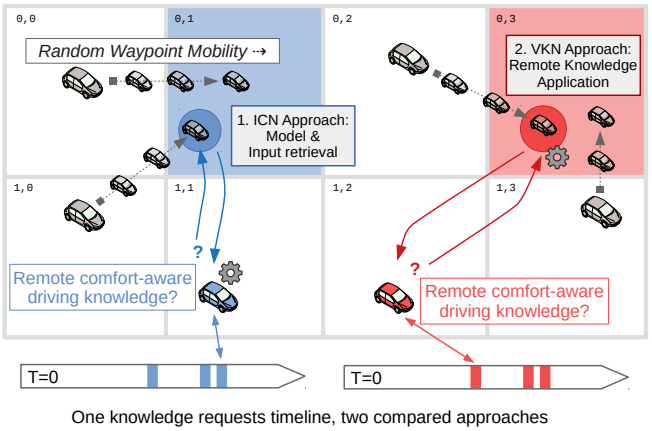

Fig. 6: Knowledge Distribution Overhead Evaluation Setup

B. Evaluation: Comfort Knowledge Distribution

To describe and evaluate the potential performance gains of VKN over traditional information-centric schemes, we study the bandwidth impact of knowledge distribution in vehicular networks. While we use comfort knowledge as a case study, the evaluation scheme is independent from the considered models, and shows an initial comparison of the knowledge distribution performances of VKN and ICN.

We investigate on the following scenario: A set of vehicles wish to obtain comfort knowledge about various areas, as the result of a `model.env_comfort` application, to personalize their driving behavior. As illustrated by Figure 6:

a) A set of vehicles is simulated in a $1km^2$ area divided into a square grid, each cell represents a distinct area where input information can be sensed.
b) Vehicles' movements are simulated following the Random Waypoint mobility model.
c) A timeline of R=10000 knowledge requests is generated. Following a Poisson process, each vehicle periodically requests comfort knowledge from an area, i.e., a grid cell.

We implement both a VKN and ICN-based approach to compare their overhead impact on comfort knowledge distribution. Figure 7 describes the process of knowledge creation performed when a vehicle v requests comfort knowledge implemented for both VKN, and ICN with dashed lines. Through VKN, vehicles use *knowledge application requests* to obtain remotely-created knowledge, instead of direct model and input information requests in the ICN-based approach. Figure 8 summarizes VKN remote application of `model.env_comfort`. A vehicle `ego_node` on the left directly transmits a knowledge creation request to `remote_node` in the target area in possession of the required model's version *1.1* bytecode. In turn, `remote_node` computes the comfort knowledge using locally sensed input.

We run 100 simulations for two distinct scenarios, whose results are summarized in Table I. The first scenario involves a density of 1000 vehicles/km^2 in a grid divided into $50m^2$ cells, to imitate urban conditions. There, the amount of model transfers was reduced by $15 \pm 0.2\%$ using VKN over the ICN approach. The second scenario simulates rural conditions,

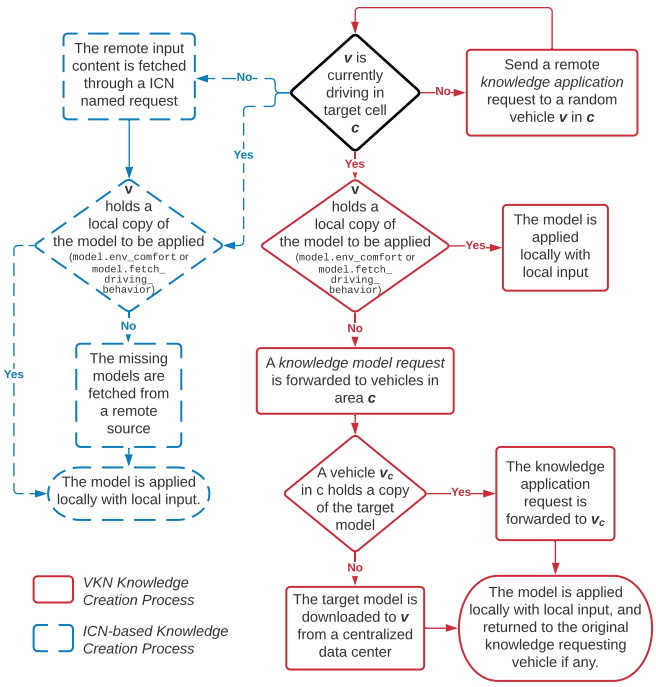

Fig. 7: Comparison of the VKN and ICN-based Knowledge Creation Processes

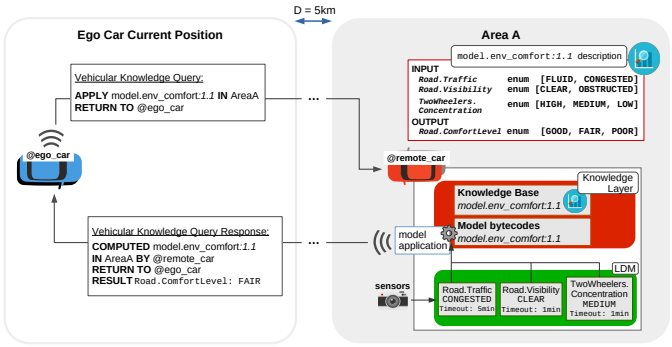

Fig. 8: Comfort Level Retrieval in a Remote Area using VKN

involving a density of 200 vehicles/km^2 in a grid of $200m^2$ cells. There, the amount of model transfers was reduced by $44 \pm 0.4\%$.

The bytecode size of ML models depends on their nature and complexity. It ranges from megabytes to hundreds of megabytes for deep neural networks. Depending on the model, VKN thus allows a moderate to strong reduction of overhead. While an extra 0.3 to 0.6 model discovery messages per request were transmitted using the VKN approach for respectively the urban and rural scenarios, their size is negligible in front of the bandwidth saved through reduced model transfers. Considering an equivalent size of model input and output, the VKN approach becomes beneficial in terms of overhead from a model size of 100 kilobytes. It reaches, from 1 megabyte, a stable 14 or 40% overhead reduction in terms of model transfers for respectively the urban and rural scenarios.

Using VKN, model input transfers were traded with an equivalent amount of *knowledge samples* transfers for the urban scenario, and a slight increase of 2% for the rural scenario. Depending on the model, as knowledge is typically lighter than the input it was extracted from, overhead can be further reduced.

TABLE I: Amount of Communications Per 10000 Requests in the Urban and Rural Scenarios

Transmissions per 10000 requests:	Urban scenario (1000 vehicles)		Rural scenario (200 vehicles)	
	VKN	ICN	VKN	ICN
Model Bytecode	851 ± 2	1000	111 ± 2	200
Model Output	9013 ± 11	0	9822 ± 3	0
Model Input	0	9006 ± 11	0	9592 ± 4
Model Discovery	2864 ± 12	0	5703 ± 30	0
VKN Overhead Reduction:	Urban scenario (1000 vehicles)		Rural scenario (200 vehicles)	
	VKN	ICN	VKN	ICN
1MB / Bytecode	0.85GB	1GB	0.11GB	0.2GB
1KB / Discovery	2.9MB	0	5.7MB	0
Total overhead	0.86GB	1GB	0.12GB	0.2GB
Difference	14%		40%	

V. Research Applicability

We described an application of cooperative knowledge creation. By remotely applying models in the area where their input is sourced, unnecessary transfers of information are avoided to the benefit of knowledge. Similarly, mechanisms have been defined in the literature to train *knowledge models* themselves while avoiding the transmission of training information for privacy and efficiency concerns. Federated Learning (FL) is an open research topic in which multiple nodes cooperatively train a shared model without directly exchanging training information. Rather, model updates are separately trained by each node with local input and subsequently aggregated. However, it is not trivial to ensure that all local nodes are interested in training and using the same model. Before being able to start the training, nodes should be able to determine who among their neighbors is in possession of what type of model and has access to what type of information. VKN can be used to orchestrate the client selection process of FL algorithms, delegating model training to remote vehicular nodes, to select the most pertinent training nodes depending on their available input and knowledge [15].

VI. Conclusion

Vehicular networks have been extensively studied in the past years. Several standards have been developed to store and share information. However, challenges remain to transition from an information-centric networking model to a model where common standards for knowledge characterization, description, storage, and sharing allow nodes in vehicular networks to take full advantage of data-driven AI techniques. In this paper, using a common definition of knowledge, we determined under what forms it exists in vehicular networks, allowing us to concretely propose a structure for knowledge description, storage, and sharing. Through a passenger comfort-based rerouting application, we exemplified the concept and showed significant overhead reduction. Finally, we

note the potential benefits of Vehicular Knowledge Networking for the open topic of Federated Learning. Future work will focus on implementing, simulating and measuring the benefits of using VKN through packet-level simulations.

REFERENCES

[1] Chaim Zins. "Conceptual Approaches for Defining Data, Information, and Knowledge". In: *Journal of the American Society for Information Science and Technology* 58 (Feb. 2007), pp. 479–493.

[2] M. Ruta et al. "A Knowledge Fusion Approach for Context Awareness in Vehicular Networks". In: *IEEE Internet of Things Journal* 5.4 (Aug. 2018), pp. 2407–2419.

[3] Q. Qi et al. "Knowledge-Driven Service offloading Decision for Vehicular Edge Computing: A Deep Reinforcement Learning Approach". In: *IEEE Transactions on Vehicular Technology* (2019), pp. 1–1.

[4] Mohammad Irfan Khan et al. "Deep learning-aided resource orchestration for vehicular safety communication". In: *Wireless Days 2019, IEEE/IFIP Days 2019, 11th edition, Manchester, UK*. Apr. 2019.

[5] D. Wu et al. "Vision and Challenges for Knowledge Centric Networking". In: *IEEE Wireless Communications* 26.4 (Aug. 2019), pp. 117–123.

[6] Benjamin Klotz et al. "VSSo - A vehicle signal and attribute ontology". In: *SSN 2018, 9th International Semantic Sensor Networks Workshop*. Monterey, USA, Oct. 2018.

[7] H. Yao et al. "Artificial Intelligence for Information-Centric Networks". In: *IEEE Communications Magazine* 57.6 (June 2019), pp. 47–53.

[8] H. Hao et al. "Knowledge-centric proactive edge caching over mobile content distribution network". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2018, pp. 450–455.

[9] X. Zhang, H. Wang, and H. Zhao. "An SDN framework for UAV backbone network towards knowledge centric networking". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2018, pp. 456–461.

[10] D. Sapra and A. D. Pimentel. "Deep Learning Model Reuse and Composition in Knowledge Centric Networking". In: *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. 2020, pp. 1–11.

[11] T. Higuchi et al. "On the feasibility of vehicular micro clouds". In: *IEEE Vehicular Networking Conference (VNC)*. Nov. 2017.

[12] P. Mach and Z. Becvar. "Mobile Edge Computing: A Survey on Architecture and Computation Offloading". In: *IEEE Communications Surveys Tutorials* 19.3 (2017), pp. 1628–1656.

[13] David Martin et al. "Bringing Semantics to Web Services: The OWL-S Approach". In: *Semantic Web Services and Web Process Composition*. Ed. by Jorge Cardoso and Amit Sheth. 2005, pp. 26–42.

[14] N. Lal, S. Kumar, and V. K. Chaurasiya. "An efficient update strategy for content synchronization in Content-Centric Networking (CCN)". In: *China Communications* 16.1 (2019), pp. 108–118.

[15] D. Deveaux et al. "On the Orchestration of Federated Learning through Vehicular Knowledge Networking". In: *2020 IEEE Vehicular Networking Conference (VNC)*. 2020, pp. 1–8.