Towards In-Network Computing Infrastructures for Connected Vehicles

Dennis Grewe, Naresh Nayak, Uthra Ambalavanan, Sebastian Schildt
Robert Bosch GmbH, Corporate Sector Research and Advance Engineering, Renningen, Germany
{dennis.grewe, naresh.nayak, uthra.ambalavanan, sebastian.schildt}@de.bosch.com

***Abstract*—The demands of Highly Automated Driving (HAD) applications with respect to the underlying computing and networking infrastructure vary widely from the contemporary cloud applications. Named Function Networking (NFN) as a computing concept along with loose coupling provided by the Information Centric Networking (ICN) enables implementation of several use-cases with respect to autonomous driving. In this paper, we present NFN for automotive applications with modified resolution strategies along with a proof-of-concept implementation.**

I. Introduction

With Highly Automated Driving (HAD) applications just around the corner, vehicle-to-vehicle and vehicle-to-infrastructure connectivity have taken a center stage. These applications rely on timely reception of information derived from processed data streams to steer the automotive. For instance, vehicle entering a blind curve could be safer and faster with the availability of processed sensor data from other vehicles in the vicinity in combination with map and weather information. Processing all these data sources locally within the in-vehicle computing infrastructure is not always feasible, e.g., when several vehicles are crossing a busy junction. Furthermore, the results of these computations could benefit multiple vehicles, if offloaded. The real-time nature of HAD applications, however, challenge the limits of cloud computing with respect to scalability and response times. While recent advances in edge-computing technologies enable offloading of these computations to nearby computing infrastructures (e.g., [1]), additional concepts need to be augmented for handling the mobility aspects of these applications. The underlying host-centric networking models make information dissemination and service invocations difficult in highly mobile applications like autonomous driving.

Information Centric Networking (ICN) [2] provides a loosely coupled communication paradigm by addressing data/information directly instead of the hosts. Named-Function Networking (NFN) [3] further extends ICN enabling execution of computation functions inside the network. NFN functions based on—A workflow definition - *what* to execute and a resolution strategy - *where* to execute a computation. The loosely coupled communication model of ICN enables NFN to move data processing and reuse already computed results by using the network as a content store. While NFN serves as an enabler for several use-cases with respect to HAD applications, it was optimized mainly for usage in cloud environments. In order to handle the network requirements of connected vehicles, the focus of in-network computing infrastructures needs to shift towards time-sensitive data delivery in mobile networks, instead of focusing on efficient data delivery - as ensured by the default NFN resolution strategy.

We present, in this paper, novel modifications to the NFN resolution strategies for aligning it with the requirements of the use-cases for HAD applications. Furthermore, we present a proof-of-concept implementation for the presented resolution strategies. Finally, we conclude with an outlook on extending the NFN concepts towards a Compute-First Network (CFN) [4].

II. Resolution Strategy for the automotive IoT

NFN encodes the function calls within ICN names and transfer the function code within the payload of named data objects. Any node on the forwarding route is able to execute the function, according to the workflow definition. In NFN, such definition is encoded using the λ calculus. Therefore, an NFN request consists of the function name including the names of data items to operate the function. An example for a workflow is:

```
func/traffic/density(/data/daytime,
  func/count/cars(/data/location, /data/direction))
```

In NFN, a forwarding strategy is used to determine how a workflow is forwarded and executed in the network. The default forwarding strategy in NFN follows a “Find-or-Execute” (FoX) semantic. A node executes a function, if it has not found an already computed result within the network caches [3]. Regarding moving vehicles, a result of a computation offloading is never overheard by the car, while the NFN infrastructure still tries to find an already computed result.

For mobile scenarios, the NFN FoX strategy is enhanced to a “Find-and-Execute” (FaX) semantic. Instead of waiting and searching the network for cached computed results, the computation will be started immediately. If a cached result is delivered by the network before the computation has finished, the result will be forwarded to the requester and the execution will be stopped.

Regarding nested computations, the FaX strategy is enhanced to a “Find-or-Pull-and-Execute” ((FoP)aX) strategy making use of NFN’s request-to-compute messages [5]. These messages provide a mechanism to pull intermediate results from other execution nodes in the network. Similar to the FaX strategy, (FoP)aX also start a computation immediately, if possible. In a mobile scenario, the execution of a complex

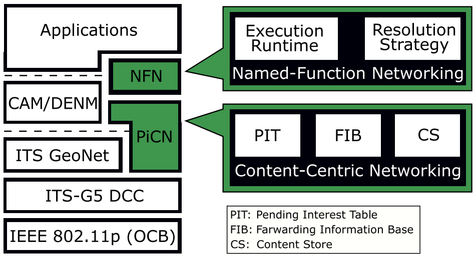

Figure 1: NFN on ETSI ITS-G5 protocol stack.

worflow definition follows the movement of a vehicle by fetching intermediate result from neighboring nodes to reduce the overall computation time [6].

III. In-Network Compute Prototype

The stack developed integrates the NFN implementation PiCN and enables vehicular units to exchange ITS-G5 and ICN messages between other vehicles and an infrastructure simultaneously [7]. Figure 1 illustrates the integration of NFN into the communication stack. Existing safety-critical applications are still supported using the ETSI ITS-G5 stack, while computations can be offloaded to infrastructural components using the NFN layer.

The layers used in the stack include:

- IEEE 802.11p access layer in the 5.9 GHz band through a Linux kernel modification [8].
- ETSI ITS-G5 Network & Transport layer (incl. DCC, GeoNet, CAM/DENM) based on OpenC2X [9].
- OpenC2X NFN module based on the NFN implementation PiCN[1].

IV. System Demonstration

The presented FaX strategy is demonstrated in a real world automotive deployment including vehicles and road-site units (RSU) using ETSI ITS-G5 (cf. Figure 2). The feature set of the stack is deployed on IPC boards (NF36-2600 - 1GHz CPU, 1GB RAM). As part of the demonstration, the communication and the computation offloading capability are presented using NFN computations.

The vehicle sends out NFN requests for a computation intensive operation during the journey (e.g., augmentation of a map). An RSU overhearing the request starts the computation following the FaX strategy, while the car moves out of the communication range of the RSU. As soon as the car connects to the next RSU, it send out the NFN request again. Based on the FaX strategy, the second RSU starts the computation and searches for a cached result in the network. Since the previous RSU has already finished the computation in the meantime, the result can be directly transported to the second RSU. Finally, the result can be directly shipped to the car. While the FaX strategy results in occupying more computational resources at the edge compared to the default FoX strategy, FaX has shown improvements in timely data delivery in mobile scenarios. The

Figure 2: Vehicular unit hardware on the field test.

field experiment has shown the offloading of computations to the infrastructure and demonstrates how a result can be fetched, in case the computation time is longer than the time the car is connected to a single RSU.

V. Conclusion

Automated driving will transform to reality in the near future. Novel computing and networking paradigms are required for handling the requirements of applications within this domain. In this paper, we presented NFN with modified resolution strategies for automotive use-cases. In the future, we will work on extending NFN towards a compute-first networking approach – introducing computations as the first class citizen in the network leveraging principles from data-oriented networks and modern programmable networking technologies like ICNs, P4 (e.g., [4], [10]).

[1] https://github.com/cn-uofbasel/PiCN

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, Fog Computing and Its Role in the Internet of Things, in Proc. of the MCC Workshop on Mobile Cloud Computing, 2012

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, Networking named content, in Proc. Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT), New York, NY, USA: ACM, 2009, pp. 112.

[3] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin, An information centric network for computing the distribution of computations, in Proc. ACM Conf. on Information Centric Networking, Paris, France, 2014

[4] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, Compute First Networking: Distributed Computing meets ICN, in. Proc ACM Conf. on Information-Centric Networking, Macau, China, 2019

[5] C. Scherb, B. Faludi, and C. F. Tschudin, Execution State Management in Named Function Networking, in Proc. of the IFIP Networking Conference and Workshops. Stockholm, Sweden, 2017

[6] C. Scherb and D. Grewe and M. Wagner and C. F. Tschudin, Resolution Strategies for Networking the IoT at the Edge via Named Functions, in Proc. of the IEEE Consumer Communications & Networking Conference (CCNC). Las Vegas, USA, 2018

[7] D. Grewe, C. Marxer, C. Scherb, M. Wagner, and C. F. Tschudin, A network stack for computation-centric vehicular networking, in Proc. ACM Conf. on Information Centric Networking, Boston, U.S.A., 2018

[8] R. Lisovỳ, M. Sojka, and Z. Hanzálek, IEEE 802.11p Linux Kernel Implementation, Tech. Report, Prague, Czech Republic, 2014

[9] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer and F. Dressler, Demo: OpenC2X - An open source experimental and prototyping platform supporting ETSI ITS-G5, in Proc. of the IEEE Vehicular Networking Conference (VNC), 2016

[10] M. Luthra, B. Koldehofe, J. Hchst, P. Lampe, A. H. Rizvi, R. Kundel, and B. Freisleben, INetCEP: In-Network Complex Event Processing for Information-Centric Networking, in Proc. ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Cambridge, United Kingdom, 2019