Demo: In-Vehicle Communication Using Named Data Networking

Zachariah Threet*, Christos Papadopoulos†, Proyash Podder‡, Alex Afanasyev‡, William Lambert*, Haley Burnell*, Sheikh Ghafoor*, Susmit Shannigrahi*,
*Tennessee Tech,
Email: {zgthreet42,wllambert42,hlburnell42,sghafoor,sshannigrahi}@tntech.edu
,‡Florida International University
Email:{aa, ppodd002}@cs.fiu.edu,
† University of Memphis
Email: christos.papadopoulos@memphis.edu

***Abstract*—Data-centric architectures are a candidate for in-vehicle communication. They add naming standardization, data provenance, security, and improve interoperability between different ECUs and networks. In this demo, We demonstrate the feasibility and advantages of data-centric architectures through Named Data Networking (NDN). We deploy a bench-top testbed using Raspberry PIs and replay real CAN data.**

I. Introduction

Data-centric and Service-Oriented communication architectures are recently being considered for in-vehicle communication. Such architectures provide enhanced capabilities such as standard naming, provenance, security, interoperability, and better transport efficiency. H owever, t hey a re m ore complex than standard in-vehicle communication technologies such as the Controller Area Network (CAN), which means they incur higher overhead. Our demo uses Named Data Networking (NDN), a data-centric communication architecture originally applied to the Internet. We have developed a testbed to investigate how NDN can implement the data-centric architectures without incurring unnecessary complexity or overhead. The motivation for NDN is its relationship with security by design that makes it uniquely well-suited for the automotive domain. NDN uses only two packet types, an Interest and a Data packet, the former used by a client to request content, and the latter by a provider to respond with the content. All data is named with unique, immutable names, digitally signed, and can be cached in the network. Traditional vehicle CAN bus systems are very susceptible to DoS attacks, but because an NDN node needs to make an Interest before it will receive Data packets, using unsolicited Data packets as a method of DoS attacks is impossible in the NDN architecture. Additionally, NDN implements authentication that can validate a specific signature's authorization to sign a specific p iece o f d ata [2]. Our demo uses NDN to transfer CAN frames from one NDN node to another. These gateways are implemented by two Raspberry PIs running Ubuntu server 20.04.3 LTS connected with Ethernet running NFD [1]. We replay a real CAN trace and demonstrate: (a) how one NDN node is able to forward a CAN trace to another by request over Ethernet, (b) a simple NDN naming scheme to address CAN frames, (c) low transport delay, and (d) quick recovery from packet loss.

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23011
www.ndss-symposium.org

II. Testbed and Experiments

Our testbed was built to emulate a segmented network with NDN nodes acting as gateways between segments. The producer loads the first one thousand messages of a file from ROAD CAN dataset[3] into the memory to avoid file I/O delays. Future iterations of this testbed will feature these nodes being connected to real CAN segments instead of reading this data from a file. The producer then enters a loop responding to Interests with a Data packet containing the requested CAN message. The consumer enters a loop sending Interests of the form "$/\mathrm{vehicle}/\mathrm{can0}/\mathrm{nextSeqNum}$" with the next sequence number, stopping after receiving an EOF from the producer. To measure the latency NDN introduced, we recorded a timestamp before the consumer sent an Interest and immediately after it received the data packet. The difference between these numbers was used as the transmission time in our log. The average Interest/Data latency after all messages finished was 73ms. The Ethernet latency was under 1ms. Most of this time was spent in the application library, which is Python-based. A faster implementation would reduce the latency substantially. To demonstrate the quick recovery possible through NDN caching, we introduced Data packet loss by programming the consumer to resend every 100th Interest after receiving the data packet. We measured the latency of these duplicate Interest/Data exchanges and found the latency to be 9.7% of a standard exchange. This shows how quickly NDN is able to recover from losing Data packets.

III. Conclusions

We demonstrated that data-centric architectures such as NDN are promising network architectures for in-vehicle communication. We demonstrated naming, the Interest/Data exchange, and the recovery advantages of in-network caching. We plan to extend our work to use signed messages, richer topologies, and a hybrid testbed for different vehicle link-layer technologies.

References

[1] A. Afanasyev, J. Shi, et al. NFD Developer's Guide. Tech. Rep. NDN-0021, NDN, 2015.

[2] C. Papadopoulos, A. Afanasyev, and S. Shannigrahi. A name-based secure communications architecture for vehicular networks. In *2021 IEEE Vehicular Networking Conference (VNC)*, pages 178–181, 2021.

[3] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay, and F. L. Combs. Road: The real ornl automotive dynamometer controller area network intrusion detection dataset (with a comprehensive can ids dataset survey & guide). *ArXiv*, abs/2012.14600, 2020.