VANET via Named Data Networking

Giulio Grassi*, Davide Pesavento*, Giovanni Pau*†, Rama Vuyyuru‡, Ryuji Wakikawa§, Lixia Zhang†
*LIP6 – Université Pierre et Marie Curie, Sorbonne Universités – Paris, France
†Computer Science Department – University of California, Los Angeles, CA 90095, US
‡Toyota ITC U.S.A. – Mountain View, CA, US
§WIDE project – Tokyo, Japan

Abstract—**In this paper we apply the Named Data Networking [1], a newly proposed Internet architecture, to networking vehicles on the run. Our initial design, dubbed V-NDN, illustrates NDN's promising potential in providing a unifying architecture that enables networking among all computing devices independent from whether they are connected through wired infrastructure, ad hoc, or intermittent DTN. This paper describes a prototype implementation of V-NDN and its preliminary performance assessment, and identifies remaining challenges.**

I. INTRODUCTION

Recently manufactured vehicles are equipped with not only computation powers but also a variety of wireless communication interfaces such as 3G/LTE, WiMAX, WiFi, IEEE 1901 (Power Line Communication), and 802.11p (DSRC/WAVE). Ideally a car should be able to utilize any and all of these interfaces to communicate with either infrastructure servers or other vehicles as needed by applications. Whenever more than one interface is available, the vehicle should be able to pick and choose the best one or use multiple in parallel.

Indeed an increasing number of vehicles are connected to Internet today, however they are mainly connected via cellular networks *only*. The concept of connecting vehicles through Road-Side Units (RSUs) has long existed, but only certain regions or countries deployed them. Standards have been developed for direct Vehicle-to-Vehicle (V2V) communication, however the usage is limited to one-hop communication for collision prevention only. There have been numerous publications exploring the use of V2V to support a much broader range of applications, unfortunately those results are isolated point solutions using various patches to overcome the limitations of TCP/IP, hence lacking a basic framework to prove their utility.

In this paper we apply the design of Named Data Networking (NDN) to address VANET challenges. We show that naming data decouples communication from specific interfaces and endpoints, enabling a car to utilize any available interfaces and fetch data from any other nodes as soon as physical connectivity comes into existence. As a proof of concept, we designed and implemented a prototype of Vehicular NDN, V-NDN [2], and demonstrated its utility through real experimentation. We also used simulations to explore V-NDN feasibility at large scale. The contributions of this paper can be summarized as follows: (1) we articulated the new NDN functional requirements for VANET and sketched out initial solutions (Section II-B); (2) we developed a prototype V-NDN implementation (Section II-C); (3) we conducted experimentation via both demonstration (Section III) and simulation (Section IV); and (4) we identified remaining challenges in rolling out V-NDN (Section V).

II. V-NDN: DESIGN AND IMPLEMENTATION

In this section we describe the basic NDN model, then we discuss the modifications needed to accommodate VANET specific features and the V-NDN implementation.

A. NDN

In an NDN network, each application names the data it wants to fetch, and the network uses these application data names directly. Thus the names used in communication are independent from which interface one wants to use, or from whichever nodes the data may come from. The first NDN paper [3] introduced data producers, consumers, and routers as three main types of entities in an NDN network. Conceptually each NDN node maintains three major data structures: *Content Store (CS)*, *Pending Interest Table (PIT)*, and *Forwarding Information Base (FIB)*. The CS is a temporary cache of Data packets that the node has received, which can potentially be used to satisfy future Interests. The PIT stores all Interests that have been forwarded but not satisfied yet. If a received Interest does not have a match in either the CS or the PIT, it will be forwarded toward the data producer(s) according to the FIB (presumably built by a routing protocol). When a Data packet arrives, the router finds the matching PIT entry and forwards the data to all downstream interfaces listed in the PIT entry. It then removes that PIT entry, and caches the Data in the CS. If a Data packet does not have a matching PIT entry, it is unsolicited and is dropped. Neither Interest nor Data packets carry IP addresses; Interest packets are routed toward data producers based on the names carried in them, and Data packets return based on the state information set up by the Interests at each hop.

B. V-NDN

Vehicular networking possesses two fundamental characteristics: ad hoc, intermittent connectivity, and the ability of physically transporting data. In a V-NDN network, a car may play any of the four roles: data *consumer*, data *producer*, *forwarder*, when it is connected to either infrastructure or other vehicles, and *"data mule"*, when it carries data across distance while having no connectivity to anyone else. Unlike other types of mobile devices, vehicles have no concern with computational/storage capacity or power supply.

NDN is a great enabler to vehicle networking by removing the constraints in TCP/IP protocol stack, however several modifications to the baseline NDN operations are necessary for the VANET environment. First, since all communications happen over wireless channels, one should take full advantage from wireless broadcast nature. Instead of only accepting data with matching entries in PIT, a vehicle may want to cache all received data regardless of whether it has a matching PIT entry or whether it needs the data for itself. Since a car can have much bigger data storage than mobile phones, this opportunistic caching strategy can be advantageous in facilitating rapid data dissemination in highly dynamic environments.

Second, Data packets can be carried by running cars even when they have no network connectivity. Data can move away from the producer's location either by requests or by car movements. When a car responds to an Interest with Data, the reply can spread via wireless broadcast channel to neighboring cars and be cached by all the receivers. When these cars physically move around, they serve as data mules carrying the content to wider area. Having large number of mules enlarges data spreading areas and increases rendezvous opportunity between consumers looking for a specific piece of data and mules carrying a copy of it.

Finally, the dynamics of connectivity among moving vehicles makes it difficult, if not completely infeasible, to run a routing protocol to build and maintain the FIB. Therefore V-NDN must develop other means to guide Interest packet forwarding.

C. Implementation

We developed a V-NDN prototype under Ubuntu Linux 12.04 LTS[1]. The implementation overview is depicted in Fig. 1.

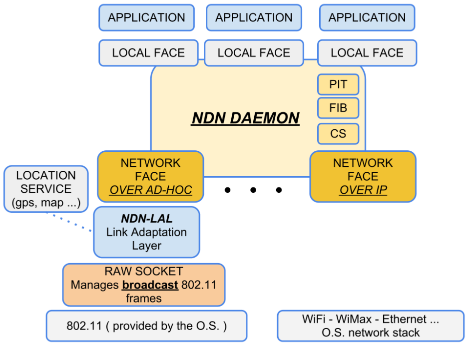


Fig. 1. V-NDN implementation framework

1) **NDN Daemon:** Provides core NDN capabilities by maintaining the key data structures of CS, PIT and FIB, and taking care of name prefix matching and packet forwarding decisions. If the node is equipped with multiple network interfaces, the current implementation takes a simple approach of forwarding each Interest to all the interfaces that are available at the time, therefore the FIB is not used in Interest forwarding at this time. The exploration of forwarding strategy design [4] for VANET is part of our future work. Finally, the CS caches all Data packets overheard on the wireless channel (solicited or not)[2].

2) **NDN Local Faces:** These are interfaces between the applications and the NDN daemon. They support application registration (an application registers with the local FIB the name prefix of the data it produces), Interest request, and content delivery.

3) **NDN Network Faces:** These faces provide the adaptation functions coupled with the specific technology used in the communication. We use IEEE 802.11 in *ad hoc* (IBSS) mode for V2V, and provide the interface with the *Link Adaptation Layer* which supports WiFi broadcast (see below). We use several wireless technologies for V2I including WiMax, 3G, and WiFi mesh networks. For 3G connectivity, the NDN Network Face provides the necessary adaptation for IP tunneling between the mobile and some NDN node in the core network.

4) **Link Adaptation Layer (LAL):** LAL is conceptually layer-2.5, designed to efficiently take advantage of the particular layer-2 mechanisms. The LAL sends all packets as L2 broadcast, using a raw 802.11 frame to carry NDN packets directly. However, broadcast support in WiFi is practically nonexistent, leaving a number of tasks to be surrogated by LAL [5] [6]. We discuss our solution to this problem below.

5) **Location Service:** Provides reverse-geocoding capabilities, as well as high-level functions on distance and heading, to the LAL, which uses them to geographically scope the communication. Moreover, some applications may choose to encode location information in data names when the content concerns a limited area, such as traffic or parking data [7]. The location might then be used to help Interest forwarding [8].

D. Enhancing WiFi broadcast for V2V communications

We use L2 WiFi broadcast for all V2V communications. However, current IEEE 802.11 standards provide neither collision prevention for broadcast transmission, nor a collision detection/recovery mechanism. Thus WiFi broadcast can suffer high losses due to collision, which can be further exacerbated by the nature of vehicular networks that feature relatively short link durations and fast changing topologies [9].

To enable efficient and resilient broadcasting, which in turn enables opportunistic forwarding and caching, we developed a simple set of mechanisms to provide WiFi broadcast support in VANET communications. Our WiFi broadcast support is coupled with our packet forwarding algorithm, as described below. We assume that each vehicle is equipped with GPS and a digital map. We use a simple greedy forwarding strategy to spread NDN Interest packets in all directions in the following way. Each Interest packet I carries the location information of its sender S. When I is received by multiple surrounding vehicles, the farthest one from S forwards I; the other receivers simply do nothing. Furthermore, S needs to know whether I

---

[1] Source code available at `https://github.com/named-data/vndn`. There are no kernel dependencies and the software is expected to run smoothly under any Linux distribution.

[2] The current implementation keeps the CS in main memory; cached items have no expiration time. A future version will employ disk-based CS and support smarter caching policies.

has been received and further forwarded, otherwise S needs to retransmit I.

We implemented the above mechanisms in the Link Adaptation Layer. Equipped with GPS data, the LAL at each node N computes the distance between the sender and itself, then sets a random wait timer (*Forwarding Timer*) based on this distance value: the shorter the distance, the longer the wait. During this wait time, if N hears the forwarding of I by another node F, N uses reverse geocoding to locate F on the map and identify the road segment in which F is located. This transmission is considered by N as an implicit partial acknowledgment of I. If N hears implicit partial acknowledgments from each of the streets stemming from where N is located (except the street where I was initially coming from, *i.e.* the location of S), it considers the packet as completely acknowledged and cancels its own Forwarding Timer. Otherwise, when the timer expires, N forwards the packet. Similarly, the forwarded packet can be heard also by S, and if S doesn't hear implicit ACKs from all the streets stemming from its location, it will retransmit the packet. All retries are upper-bounded by a preset limit n, the packet is dropped after n unsuccessful attempts.

This weighted random wait scheme statistically allows the node furthest away from a packet sender to forward the packet, resulting in fast packet propagation. LAL combines two different components to compute the timer: a deterministic component $\frac{1}{D(sender,receiver)}$ where the distance D is computed using the location service, and a small random component used to randomize the transmission time. The first component favors cars further away from the last hop, the second one reduces the chances of collision among the nodes at the same distance.

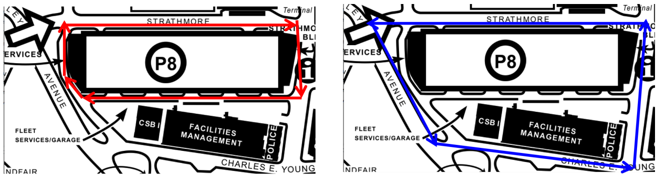
To limit excessive spreading of Interests on the vehicular network, we added a hop count to the LAL header which is incremented at each hop. An Interest is dropped when its hop count exceeds a preset limit (the experiments described below use a limit of 5 hops).

III. DEMONSTRATION

We implemented a prototype of V-NDN as proof of concept and tested it using UCLA Vehicular Testbed. The experiments involved multiple vehicles (up to 10 cars). We implemented two simple applications over NDN: **Info-Traffic** and **Road-Photo**. The first application emulates traffic information requests for a specific area; the area is encoded in the name carried in Interest packets. Instead of coordinates, we name intersections and streets stemming from that intersections, e.g. /traffic/westwood-at-strathmore/ refers to traffic information from the area surrounding the Westwood/Strathmore intersection. A car that has been or currently is at the location can respond to the Interest with proper traffic information. The Road-Photo application emulates photo requests for a specific area. Any vehicle that has been in that area recently and has taken a photo may respond.

The experiments were designed to investigate V-NDN behavior in the following communication scenarios typical for the vehicular applications domain:

- **V2V**: Vehicles exchange Interest and Data packets over WiFi; requested data is relatively local (a few hops away).
- **I2V**: The consumer node resides on the wired network (*e.g.* traffic control center server) and data producer nodes are the vehicles that are close to, or have been recently close to, a point of interest and have information about the traffic (Fig.2(b)).
- **V2I**: The consumer is on the vehicle while the producer is connected to the Internet through wired or wireless network, *e.g.* a vehicle wants to retrieve data about the traffic of an area from a centralized server (Fig.2(c)-2(a)).
- **Network disruption**: Vehicular networks are prone to disruption, the link duration is relatively short and the topology is continuously changing, resulting in frequent network partitioning [9].
- **In-network storage**: One of the NDN advantages is enabling caching in the network. This feature turns out to be essential for communication in VANETs.



(a) V-NDN Driving Route, Clock Wise one block around Parking Lot 8

(b) V-NDN Driving Route, Clock Wise two blocks around Parking Lot 8

Fig. 3. Mobility patterns

A. Field Experiments

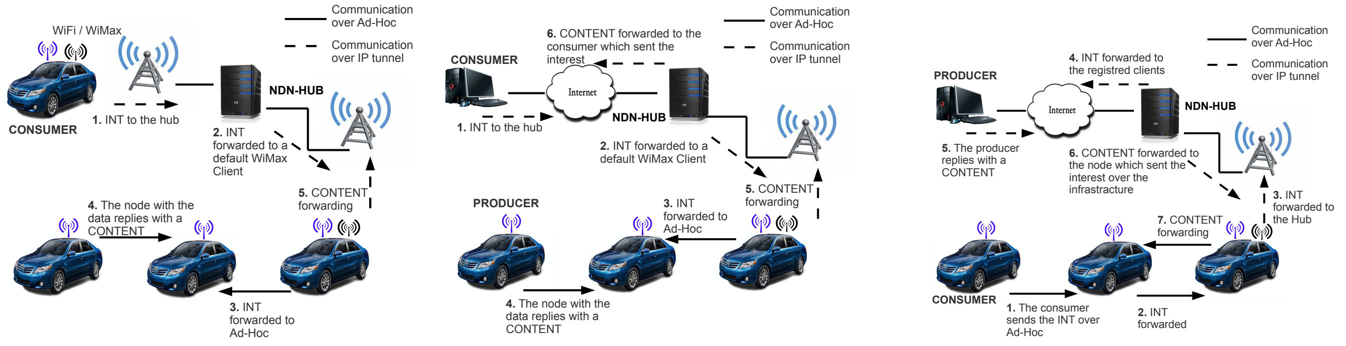
We performed a number of field experiments for both applications, varying mobility patterns and the types of communication involved (V2V, V2I/I2V, V2V2I).

1) **Still***:* These experiments were performed on the rooftop of UCLA Parking Structure P8 (265m by 76m) with *no mobility* and cars in line.

2) **Platooning***:* These experiments were performed on the rooftop of P8 with very basic *slow urban platooning*.

3) **Moving around campus***:* These experiments were performed by driving around P8. The pool of 10 vehicles was divided in two groups of 6 and 4 cars each. The smaller group of vehicle ran clockwise around P8 as shown in Fig. 3(a); the larger group ran counterclockwise and covered a larger road block which also includes P8 (see Fig. 3(b)). Car speeds ranged from 6.3 m/s to 21.2 m/s (*i.e.* ∼14 to 47 mph). This mobility pattern allows vehicles traveling in opposite direction to meet, but prevents continuos connectivities between the two groups. Traffic lights and pedestrians created dynamic partitioning in each group, as well as voids as they happen during regular course of campus traffic [9].

4) **Hardware/Software Setup***:* V-NDN was installed on low-cost netbooks, the *Asus EeePC* 1011CX, powered by an Intel Atom N80 at 1.6GHz; each node was retro-fitted by us with a MIMO-capable WiFi Card, the Unex DNXA-92, based on Atheros AR9280. All the experiments were performed using the 2.4GHz band (channel 1). Two nodes out of the 10 were equipped with a second WiFi interface operating in *infrastructure* mode (we used an Ubiquity Networks SR71USB); two



(a) V-NDN infrastructure assisted V2V: packet route

(b) V-NDN I2V packet route

(c) V-NDN V2I: packet route

Fig. 2. Packet route in V-NDN : V2I, I2V, and infrastructure assisted V2V scenarios

other nodes were equipped with a WiMAX USB dongle. We used off-the-shelf Ubuntu Linux 12.04 LTS as operating system and V-NDN was installed in user space, no kernel changes are required. V-NDN transmits broadcast packets in the WiFi interfaces operating in ad hoc mode, and tunnels NDN traffic over IP for the other interfaces (see Fig.2). A V-NDN hub is connected at the other tunnel endpoint and performs the appropriate forwarding tasks.

B. Preliminary Results

1) **V2V experiments***:* Here we focus on the effectiveness and cost of our LAL design as described in Section II-D. More specifically, we measured the number of retransmissions needed to forward a packet with a 1-hop broadcast, the response time at application level, and the effectiveness of caching. Fig. 4(a) shows the CDF for the number of retransmissions for the Info-Traffic application in all the 3 types of mobility aforementioned. For the static case, about 75% of the packets need no more than one retransmission. In the mobility scenario this number goes down to about 65%, however the type of mobility (either on the P8 roof or on the roads) has a negligible impact on the number of retransmissions. 95% of the packets are acknowledged within 5 retransmissions or less (the max-retransmission was set to 7), which is also confirmed by simulation results in Fig.6(b). Also note that 15% of packets in the static case were acknowledged before being actually transmitted (therefore they never went on the air), because neighboring cars (presumably in better positions) already forwarded the packet (see Section II-D).

The response time CDF for the Info-Traffic application is shown in Fig.4(b). In the static case, about 75% of the Info-Traffic Interests retrieved data back in less than 1 second. For the mobility case, experiments with looping around on the roof of P8 performed worse than the experiments on the roads around campus. We believe this is due to the effect of traffic lights, which lumped multiple cars together with relatively long breaks in mobility, thus facilitating data propagation between cars. In addition, P8 is in the midst of WiFi APs and our measurement detected a high noise background.

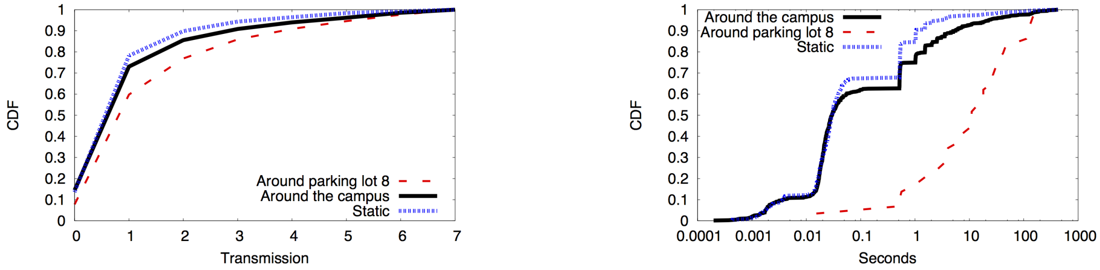
TABLE I
CACHE VS FORWARDING AMONG CONSUMERS AND MULES

	Static (#Times)	Around P8 (#Times)	Around Campus (#Times)
Use of Cache	96	3054	1840
Interest Forward	715	4685	13195

To better understand the impact of in-network cache, we excluded the data producers from the dataset and analyzed the cache/forward statistics for consumer and mules. Results are shown in Table I. Caching is more effective during mobility and particularly when mobility happens in a relatively limited area (rooftop of P8). Observing mules only (no packets are generated by these nodes), the benefit of caching becomes even more evident: in this case about 66% of the Interests found the requested data in the local cache, thanks to the aggressive caching strategy we proposed in Section II-B.

We also experimented with the Road-Photo application by letting one of the cars issue an Interest for a photo of a given area. The Interest carries a name “/picture/westwood-strathmore/”, requesting a picture from a camera that is on board of any vehicles close to Westwood/Strathmore intersection. A vehicle in that area, if any, will turn on the camera, take a snapshot, and send it back. While our design is plain and simple, we were able to retrieve 51 camera shoots. Photos had an average size of 6.3KB, resulting in an average of 5 Data packets. A photo is considered received when all its chunks arrive at the consumer. The Road-Photo application experienced an average response time of 81 seconds for the mobile case (Fig. 3), and 28 seconds in the static case; the median point was 55.6 seconds and 1 second, respectively. We believe that the sparse presence of cars in the mobile tests affected the response time, as the multiple-chunks delivery suffered from connectivity disruption much more than the Info-Traffic application, whose response entailed a single Data packet.

2) **Robust Data Availability***:* Once a piece of data is spread on the network, its availability becomes independent from the connectivity of its producer. Indeed the decoupling of data from its container and the caching enable every node with the data to use it and pass it around to anyone who expresses an Interest for it. The following experiment shows a proof of concept: a consumer asks for a content that can only be produced by one car. As soon as the consumer receives the content, the producer is shut down. Afterwards another consumer issues the same Interest. As expected, the second consumer is able to get the desired content even after the original producer is gone. Moreover, the response time does not seem to be negatively affected by the absence of the producer, thanks to the broadcast nature of the V2V communication that easily allows the spreading of content.



(a) V-NDN Info-Traffic application CDF number of Link Adaptation Layer retransmission

(b) V-NDN Info-Traffic application: CDF Response time - from the first Interest to the corresponding data

Fig. 4.

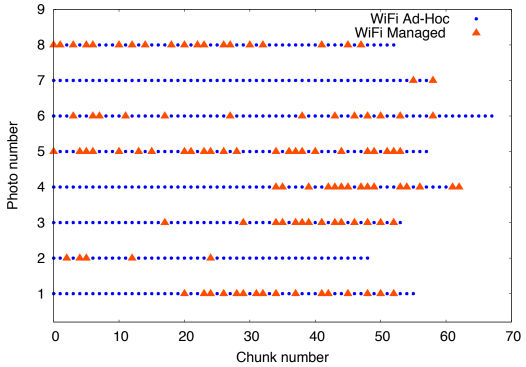


Fig. 5. V2X communication channels used for receiving photo chunks

3) **V2X scenarios and the role of the Infrastructure***:* Because one fetches data by name, V-NDN possesses the innate ability of utilizing node multihoming to communicate with other cars via ad hoc WiFi and with servers on the Internet via 3G/4G/WiFi connectivity simultaneously. The current V-NDN implementation simply forwards an Interest through all the interfaces that are available at the time. To observe V-NDN operations in a multihomed scenario, we performed an experiment of having two cars, one consumer and one producer, running around Parking Lot P32 in a clockwise fashion. At one corner of P32 we set up a WiFi access point connected to the campus network. The *consumer* was equipped with two WiFi interfaces, one operating in ad hoc mode and the other operating in infrastructure mode. The *producer* was equipped with one WiFi interface configured in ad hoc mode and one WiMAX interface connected to the campus network. We ran the Road-Photo application: the consumer requested a photo to be taken by the producer. Interest and Data packets were transmitted via all available interfaces. Photos were taken in real-time upon receiving an Interest, their sizes were between 68KB and 100KB. Each photo was split into several Data packets of 1300 bytes each. Fig. 5 shows on which interfaces the consumer received a chunk of content. The consumer was able to seamlessly receive consecutive chunks of the same picture from different interfaces via different communication channels.

IV. V-NDN: V2V COMMUNICATION AT SCALE

Since our experiments with real cars are limited in scale, we explored the scalability of V-NDN approach for ad hoc communication through simulations. We selected a dense scenario for V2V: an urban environment with high vehicular traffic, where cars travel in every direction allowed by the roads with different speed, and both traffic jams and empty streets can happen.

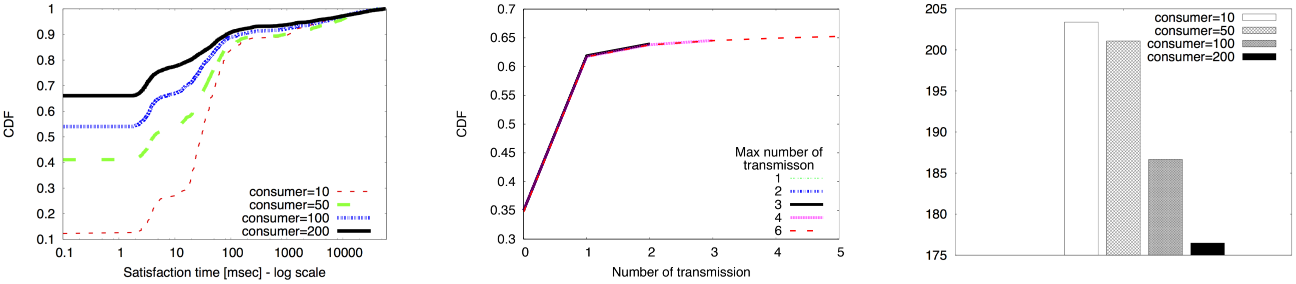
We implemented V-NDN in ndnSIM, an ns3-based NDN simulator [10]. The simulation consists of 695 cars moving in a residential area of 2100 meters × 2100 meters in the city of Los Angeles, CA $(34.040569, -118.463308)$. The cars mobility is generated using SUMO [11]. To make the simulated scenario as close to reality as possible, the traffic volume is shaped according to the importance and size of each street. The radio signal propagation is modeled using CORNER [12], an high-fidelity propagation model for urban scenarios that accounts for the presence of buildings as well as fast fading effects. We ran 300 seconds of simulated time. All the cars were equipped with a WiFi ad hoc interface, but only a subset of them ran the Info-Traffic application as either consumer or producer; the others play the role of forwarders and/or data mules. We configured 14% of the cars as producers, and varied the number of consumers.

A. Results

Since the V-NDN forwarding strategy design is part of our future work, the simulation uses the rudimentary scheme of greedy packet dissemination along all directions as we mentioned earlier. Thus the current simulation results can only serve as a quick evaluation to understand V-NDN's feasibility at scale.

Fig. 6(a) shows how long it takes for a consumer to obtain the desired content. Fig. 6(c) shows how many Interests are really transmitted on the network each time the consumer application issues an Interest. Both graphs reveal that, when the number of cars interested in the same information increases, the performance of the entire system improves substantially, as measured by the satisfaction time and overhead metrics. Indeed by letting all cars cache overheard Data packets, they become data mules. Not only a car can help forward a content that it has just received, but it can also carry the content and reply to requests for the same content in the future. As described in Section II-D, the LAL performs a retransmission when a packet is not acknowledged.

Fig. 6(b) shows the percentage of packets considered ACKed (*i.e.* the car has heard an implicit ACK from every possible



(a) V-NDN traffic interest satisfaction time

(b) V-NDN ack distribution per transmission at 2.5 level

(c) V-NDN overhead: number of interest sent in the air for each interest issued by an application

Fig. 6. V-NDN Simulation Results

direction) as a function of the maximum allowed transmissions for an Interest before giving up. We see that 35% of Interests are ACKed even before being transmitted once, because the cars have already cached the requested data earlier while overhearing on the channel. Moreover, after the second transmission, further transmissions do not seem to bring much improvement: after the second transmission, if a packet is still not ACKed, it is unlikely to be ACKed with further retries. We speculate that this effect could be due to loss of connectivity at the time.

V. DISCUSSION

While cellular networks have been viewed as the only global wireless infrastructure, in reality they suffer from spectrum scarcity and coverage limitations. At the same time vehicles are being equipped with computational power, storage, and multiple communication interfaces via various communication technologies that can be exploited to take advantage of opportunistic connectivity with other vehicles and physically transporting data over distance.

V-NDN, by naming data rather than hosts and decoupling data from IP addresses, can bring substantial benefits to vehicular communication: it removes the isolation between applications and network transport, allowing forwarding nodes to handle data based on application needs. The communication can start spontaneously—the infrastructure for the assignment of IP addresses is no longer required. Naming data also enable mules to cache it, making V-NDN resilient to connectivity disruptions that characterize vehicular networks: even when the communication between consumer and producer is interrupted, mules can bring the required data to the consumer over time. Furthermore, locally produced data and data with local meaning, such as traffic information, no longer need to be transferred to remote servers before being available to neighbor nodes; data that is produced and consumed in loco can remain in loco and be delivered to the consumers along the shortest physical path.

A. Remaining Challenges

We have taken only the first step in exploring NDN for VANET, a number of challenges remain to be addressed. We already identified, as part of our future work, the study of a V-NDN forwarding strategy to make the best use out of node multihoming.

Another challenge is data naming. [8] shows that encoding geolocation into names can help direct Interest forwarding for applications using location-based data; however other types of applications, *e.g.* fetching today's news, are unable to make use of geolocation. In addition, one may desire to encode multi-dimensional information into names, for example ideally Info-Traffic data name should carry both location *and* time information. [7] does not fully solve these problems.

More work is also needed to address the security and privacy concerns. VANETs can greatly benefit from data crowdsourcing, one challenge is how to provide consumers with authenticated data without exposing the producers' identities.

REFERENCES

[1] "Named data networking." http://named-data.net.
[2] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Acm hotmobile 2013 poster: Vehicular inter-networking via named data," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 17, pp. 23–24, Nov. 2013.
[3] V. Jacobson *et al.*, "Networking named content," in *Proc. of CoNEXT*, 2009.
[4] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," *Computer Communications: ICN Special Issue*, vol. 36, pp. 779–791, April 2013.
[5] IEEE802-WG, "IEEE 802.11, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," in *ANSI/IEEE Std 802.11, 1999 Edition (R2007)*, 2007.
[6] "IEEE P802.11p TM/D3.0 Draft Standard for Information Technology - Amendment 7: Wireless Access in Vehicular Environments," in *ANSI/IEEE Std 802.11, 1999 Edition (R2007)*, IEEE 802.11 working group, July 2007.
[7] D. Pesavento, G. Grassi, C. E. Palazzi, and G. Pau, "A naming scheme to represent geographic areas in ndn," *Wireless Days (WD), 2013 IFIP*, pp. 1–3, 2013.
[8] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid traffic information dissemination using named data," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pp. 7–12, ACM, 2012.
[9] A. Rowstron and G. Pau, "Characteristics of a vehicular network," *University of California Los Angeles, Computer Science Department, Tech. Rep*, pp. 09–0017, 2009.
[10] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," Tech. Rep. NDN-0005, NDN Project, July 2012.
[11] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.
[12] E. Giordano, R. Frank, G. Pau, and M. Gerla, "Corner: a realistic urban propagation model for vanet," in *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, pp. 57–60, IEEE, 2010.