

NDVN: Named Data for Vehicular Networking

Amine Kardi

Higher Institute of Computer and Multimedia of Gabes
University of Gabes
Tunisia

Haifa Touati

IResCoMath Research Unit
University of Gabes
Tunisia

Abstract—Named Data Networking (NDN) [1] is a newly proposed architecture for the future Internet that also opens new perspective in the way data can be disseminated and retrieved in Vehicular Ad hoc Networks (VANETs). This paper explores the potentialities of the NDN architecture applied to VANETs scenarios and proposes an enhancement to the NDN naming scheme by including geo-location and timing information. Our design, dubbed Named Data for Vehicular Networking (NDVN), encodes geographic and timing attributes in the name to guide the forwarding and optimize the traffic information retrieval along a highway. Preliminary results achieved through simulations confirm the viability and effectiveness of our proposal.

Keywords—VANET; Named Data Networking; Naming scheme;

I. INTRODUCTION

Today, thanks to various research efforts in Vehicular Networking [2], we are facing a new generation of intelligent vehicles, equipped with a variety of the latest wireless technologies such as DSRC [3]/WAVE, 3G/LTE and WIMAX. These wireless communications capabilities allow vehicles to run a variety of applications and to exchange various data with neighboring vehicles (e.g. to get traffic information) to achieve a safer and a more comfortable driving experience. Unfortunately, the specific characteristics of vehicular networks (highly dynamic network with frequent partitioning and topology changes) [4] make current TCP/IP implementation unable to optimally manage this type of communications. In fact, IP architecture based on point-to-point communication model of the wired Internet does not fit the short-lived and intermittent connectivity of the ad hoc vehicular networks (VANET).

Named Data Networking (NDN) [5,6], in contrast, represents a radically different design. Based on a different data manipulation concept, an NDN network enables data to exist in the absence of connectivity, which overcomes IP's imperfections with vehicular networks. In fact, NDN eliminates the use of node addresses and retrieves data using the application data names [7] directly without identifying content holders. This new architecture can name all kinds of data: an end point, a piece of data in a movie or a book, a data, a signal. . .

NDNs communication model is based on the exchange of two types of packets: the *consumer* requests a content by broadcasting an *Interest packet*, which carries the name of the requested data. Intermediates routers forward the *Interest* until a *provider*, node owning or caching the requested content, replies with the related *Data packet*. To manage this

request/reply exchange model, each NDN node maintains three data structures: (i) the *Pending Interest Table* (PIT) that keeps track of the forwarded Interests that are waiting for matching data packet to return, (ii) the *Forwarding Information Base* (FIB) used to relay Interests toward the data source and (iii) the *Content Store* (CS) that serves as a temporary cache of delivered data packets.

On the Interest reception, an intermediate NDN node having the content cached in its Content Store, can reply with the cached data. Otherwise, if a matching entry is found in its PIT, the node discards the Interest to prevent duplicate requests traveling down the network. If not, a new PIT entry is created and a FIB lookup returns the interface where to forward the Interest. Data packets follow the chain of PIT entries in each crossed node back to its original requester. Each intermediate node may cache incoming and overheard contents, so to act as a *provider* to serve future requests.

This paper focuses on the design of a new prototype which seeks to adapt the NDN architecture to VANET networks to optimally disseminate traffic-related information along a highway. The contributions of this work can be summarized as follows: First, we propose new structures for the naming scheme using geo-location and timing information to enhance forwarding decisions. Second, we develop an NDN simulation module for the Network Simulator NS3 that implements our proposed adaptation of the NDN architecture for VANET scenarios. Third, we take a first step to evaluate the proposed design through simulation experiments. Preliminary results prove the effectiveness of our solution for traffic-related information application using V2V communication scenario along a highway.

In the rest of this paper, we first discuss the related works in section II., then we elaborate the design details of our solution in section III. In Section IV, we present the simulations scenarios and results. Lastly, we conclude the paper and highlight our future work in section V.

II. RELATED WORKS

The feasibility of applying NDN architecture in VANETs scenarios has been recently discussed in the literature, with preliminary deployments in some cases [8, 9].

Among the proposed solutions we quote the **Vehicular Named Data Networks (V-NDN)** [8]. It has been proposed by Grassi et al. in 2013 [8, 9] The authors of V-NDN propose an implementation of the NDN architecture, written in C++, to allow vehicles to use all interfaces, technologies and available channels to communicate optimally. The design of the V-NDN solution is based on a core called the NDN Daemon which provides all the tasks of the NDN architecture

and holds its main structures such as the PIT and the Content Store. It is also composed of a Local Face that communicates with the NDN Daemon to provide different applications and a Network Face which provides communication functions using the standard IEEE 802.11 for V2V communication and several other technologies like WiMax, 3G, WiFi and DSRC for V2I communication. V-NDN has been implemented at UCLA (University of California, Los Angeles) and tested using UCLA vehicles (UCLA Vehicular Testbed) in November and December 2012.

There is also another solution proposed in 2014 by M.Chen et al. and is called **VEhicular Named Data NETwork (VENDNET)** [10]. VENDNET adapted the NDN architecture to different types of vehicular communication (V2V, V2I, Hybrid). This architecture proposes the use of the popularity prediction mechanism to find quickly the most popular content [11] in order to take advantage as much as possible of the performance of the NDN architecture. A lifetime is associated with each content based on its occurrence frequency which increases with the growth of the frequency. VENDNET proposes several changes to different types of vehicular communication such as the use of the newest cellular network Long Term Evolution (LTE) [12] in order to optimize vehicular communications [13]. VENDNET has been implemented and evaluated through simulations using OPNET modeler [14].

Another solution named **Content-Centric Vehicular Networking (CCVN)** is presented by Amadeo et al. in 2012 [15] as the first solution using IEEE 802.11p [16] for communication in this type of networks. CCVN identifies two types of roles for vehicles: a *consumer* and a *provider* and uses a communication mechanism based on three types of packets: the Basic Interest (Int-B), the Advanced Interest (Int-A) and the Content Object (C-Obj). This solution can be considered as one of the first attempts to define a CCN-like solution in a vehicular environment. The overall CCVN architecture has been deployed in NS2 simulator [17].

III. NDVN DESIGN

A. Design assumptions and targeted application

This paper focuses on highway scenarios. Targeted application is traffic-related information retrieval (e.g. accident, traffic jam, flooding, etc.) which can be very useful to increase the safety and the comfort of road-users. Traffic information is spread from a vehicle to another (V2V) and through the infrastructure servers (V2I/I2V), as depicted in Figure 1.

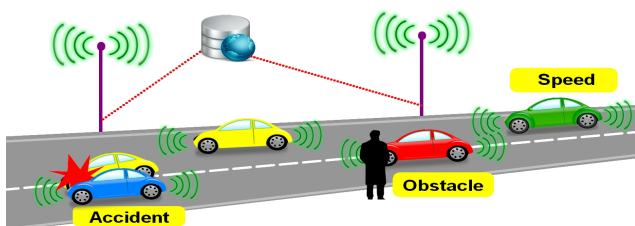


Figure 1: Traffic information dissemination through V2V, V2I

We assume that all vehicles are equipped with GPS, necessary radio interfaces, sensors, NDN modules, sufficient

memory space and requisite transmission power. We note that the data generation process is out of the focus of this paper. Furthermore, we assume that applications communicate with necessary modules such as GPS and the Content Store is managed by a caching policy to remove the least requested data if necessary.

B. Naming scheme

Naming data is one of the most fundamental axis of the NDN architecture; when a *consumer* needs a specific data, it sends out an Interest packet containing a hierarchically structured name of the desired data and waits for a Data packet(s) as a response. This mechanism seems very adaptable in the case of vehicular networks to ensure maximum dissemination of traffic information and therefore a minimum search time.

It is clear here that such system requires a very specific naming design allowing producers to describe precisely generated information and helping *consumers* to express clearly which information they need. To ensure the thoroughness of this model, we define the following requirements:

- **Uniqueness:** the data should be uniquely identified by its name.
- **Correctness:** the final structure of generated name must be correct and accurate;
- **Precision:** the name must contain a spatio-temporal indication: Where the information has been generated? / Where the desired information is located? And when the information has been generated? / At what time information is sought?
- **Security and Transparency:** the name of conveyed data must be encrypted in order to eliminate the risk of modification or injection of false data in the network.
- **Application:** the name must express the type of the requesting application.
- **Type:** the name must express the type of the conveyed data.

Starting with requirements set out above, we have inspired the following naming model:

APPLICATION TYPE LOCATION TIME

The element APPLICATION serves to define the appropriate application of this data and its possible values must be conventional and standardized among all vehicles. For example, data containing traffic information start with the standard symbol: TRF and data containing a photo start with the symbol: PCT. Traffic applications are only interested in data with names containing the keyword: TRF and ignore the rest.

The next attribute TYPE is a subtype of the previous element, it specifies the exact meaning of data and its possible values must also be normalized. For example, among the subtypes of the traffic class TRF, we define the type SPD returning the speed of the vehicle generator of the data, the type ACD encodes data traffic generated due to accidents etc.

The next element LOCATION, generated by the GPS, contains the exact location (latitude and longitude) of the data

generated / sought. Likewise, the last field TIME serves to contain the exact time of generation / wanted of the data.

This proposed structure permits to check the uniqueness condition of generated names thanks to the couple of values (LOCATION, TIME).

Finally, we note that each node should contain straightness verification mechanisms of generated names such as the encryption mechanism to guarantee the security and transparency of the conveyed data.

C. NDVN Operation

As based on the NDN architecture, NDVN defines 3 roles that can be played by different vehicles: a producer (data generator), a *consumer* (data applicant) and a “data mule” (data collector). Our proposed architecture aims to enable communication between all vehicles via any interface by promoting the use of the most appropriate one. Figure 2 shows our proposed model NDVN.

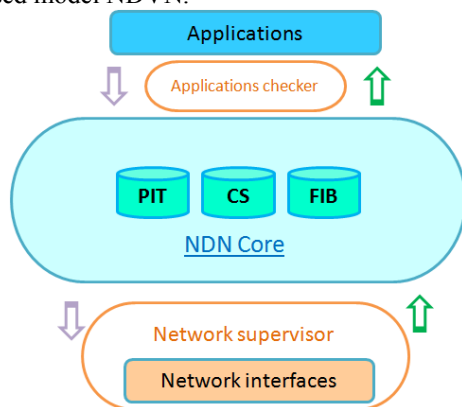


Figure 2: NDVN architecture

- **NDN Core:** it presents the core of our architecture; it includes basic structures of the NDN architecture such as the PIT, the FIB and the CS. It is clear here that forwarding tasks will be taken.
- **Applications:** they are different applications provided to users of vehicles. Generating Interest packets is the task of these applications.
- **Applications checker:** It acts as an intermediary between applications and the *NDN Core*. It is used to check the validity of generated Interest packets, to verify the correctness of generated data names and to guide data coming from the core to the corresponding application.
- **Network interfaces:** it represents available radio interfaces in each vehicle.
- **Network supervisor:** it takes a fundamental role in our design; it manages available radio interfaces, checks the validity of all received packets and directs them in and out of the system.

When an application wants to have information, it generates an Interest packet, the application checker verifies the validity of the Interest already created, if it is valid it is passed to the *NDN Core* else it will be canceled and the application will be notified in order to generate another valid Interest. Upon receipt of an Interest by the Core, it updates the appropriate tables and passes the packet to the “*Network supervisor*” which in turn passes it to the appropriate radio

interface. If the application does not receive the requested data in a definite time set by the application, it retransmits again the same Interest.

At the reception of an Interest packet, the “*Network supervisor*” checks its validity (packet structure, maximum number of hops), if it is invalid it will be removed else it will be passed to the *NDN Core*, which sends the data to the concerned *consumer* if it exists in its CS, else it checks his PIT to send the packet to the appropriate interface. If no entry exists, it updates the appropriate tables (PIT, FIB) and passes the packet to the “*Network supervisor*” to disseminate it if its validity persists.

At arrival of a Data packet, the “*Network supervisor*” checks its validity as for Interest packets if it is invalid it will be removed else it will be passed to the *NDN Core*, if the information is requested by the node, the package will be passed to the requesting application through the Applications checker else the packet will be disseminated through the “*Network supervisor*” if its validity persists. In both cases, the information in the packet will be saved in the Content Store of the vehicle.

IV. PERFORMANCE EVALUATION

A. Simulation Scenarios and Parameter Settings

To test the effectiveness and the performance of our architecture, we implemented all related modules under the NS3 simulator [18]. For the initial evaluation of our design we considered a highway scenario. We used several random mobility models to disseminate traffic information along a highway to simulate as much as possible reality. As depicted in figure 3, N vehicles are placed randomly on a 5km straight highway. The distance between neighboring vehicles is variable during the simulation time. Each vehicle moves at a variable speed of maximum 50mph. The traffic information producer is placed at the head of the vehicle sequence, while the *consumer* is placed at the tail of the sequence. This latter, broadcasts an Interest packet to request a definite traffic information data packet. For V2V communication, we used the IEEE 802.11 [19] wireless technology operating at 20Mb/s.

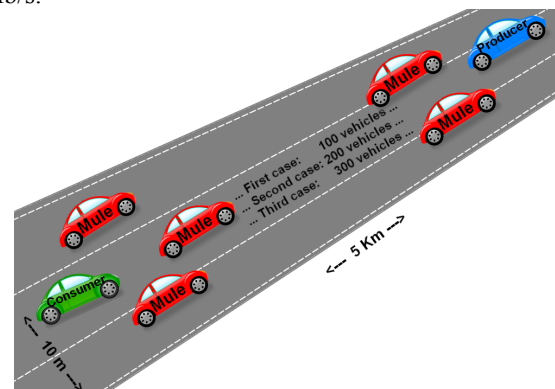


Figure 3: Simulation topology

Using these settings we evaluated the data retrieval delay metric expressed as the time needed for an Interest to be satisfied while varying vehicles density on the highway. Specifically, we evaluated three cases:

- case 1: 100 vehicles moving randomly on the highway.
- case 2 : 200 vehicles are used.

(c) case 3 reflects a much higher density scenario with 300 vehicles.

For each of the above cases, three scenarios have been investigated :

- IP model* scenario where all vehicles run the traditional IP stack using the OLSR routing protocol,
- NDVN without caching* scenario where all vehicles follow our proposed communication model without introducing the NDN caching functionality,
- NDVN with caching* scenario in which requested data could be satisfied by mule vehicles.

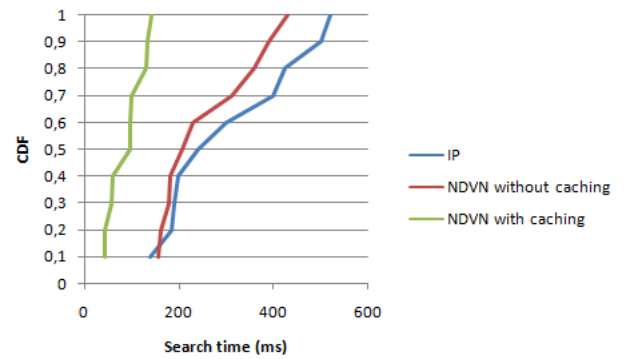
Several tests were conducted using different mobility model; each test is repeated for the three scenarios for each of the three cases.

B. Simulation results

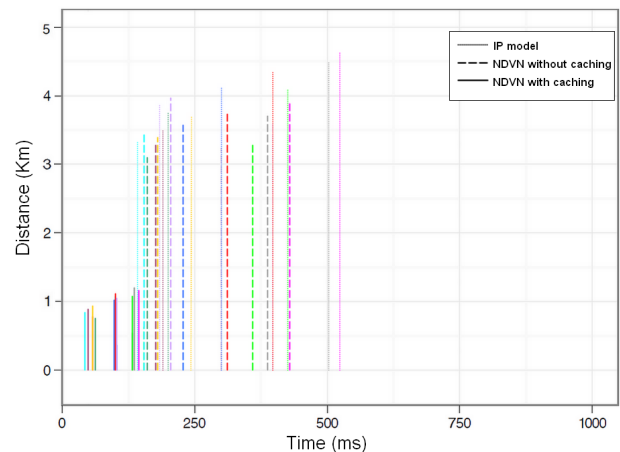
• First case (100 vehicles) :

Table 1 compares the retrieval delay for each of the above scenarios. Compared to scenario 1 (*IP model*), results of scenario 2 (*NDVN without caching*), show that for 80% of the tests, the duration of the search for information clearly decreases when the NDVN model is applied even if no caching policy is applied. The usefulness and the effectiveness of caching functionality are clearly confirmed in the third scenario, where data retrieval delay does not exceed 142ms (whereas it reaches 430ms in scenario 2 and 520ms in scenario 1) because the information is not retrieved from the very distant source but from the nearest vehicle containing the requested data. This conclusion is confirmed by figure 4.b which draws, for each test, the retrieval delay and the distance between the *consumer* and the data *provider*. Results show that when caching is applied, in the worst case, requested data is delivered by a node 1.2 km away from the *consumer*.

Figure 4.a shows the CDF for the retrieval delay of the three aforementioned scenarios. The traffic information application experienced an average search time of 310ms for the IP case, 259.8ms in the NDVN without caching and 89.7 ms when caching is applied.



(a) CDF of the search time



(b) Search time coupled with the distance between the consumer and data provider (First case: 100 vehicles)

Figure 4: Traffic information search time in the low density case (100 vehicles)

• Second case (200 vehicles) :

Using 200 vehicles, simulations also show that Scenario 3 is the best in finding the requested data within an optimum time which does not exceed 180ms. Table 2 and figure 5.a and b show that 70% of the tests are better when using the NDVN model instead of the IP architecture.

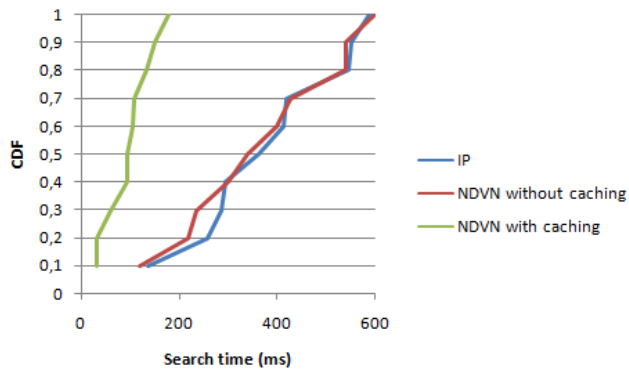
Scenarios Tests	IP model	NDVN without caching	NDVN with caching
Test1	200 ms	161 ms	60 ms
Test2	140 ms	155 ms	42 ms
Test3	300 ms	230 ms	95 ms
Test4	190 ms	177 ms	43 ms
Test5	400 ms	310 ms	97 ms
Test6	185 ms	205 ms	100 ms
Test7	425 ms	360 ms	130 ms
Test8	240 ms	180 ms	55 ms
Test9	520 ms	430 ms	142 ms
Test10	500 ms	390 ms	133 ms
Average search time (ms)	310 ms	259.8 ms	89.7 ms

Tableau 1: Difference between search times (First case: 100 vehicles)

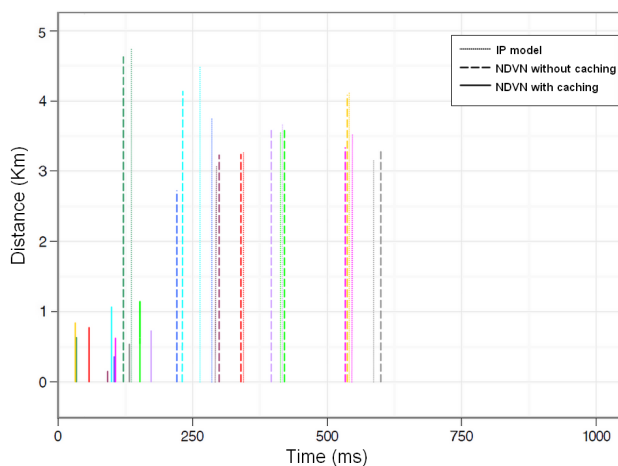
Scenarios Tests	IP model	NDVN without caching	NDVN with caching
Test1	137 ms	120 ms	32 ms
Test2	259 ms	235 ms	95 ms
Test3	287 ms	220 ms	105 ms
Test4	294 ms	300 ms	95 ms
Test5	362 ms	340 ms	60 ms
Test6	419 ms	400 ms	180 ms
Test7	412 ms	428 ms	152 ms
Test8	544 ms	542 ms	33 ms
Test9	550 ms	540 ms	110 ms
Test10	587 ms	600 ms	135 ms
Average search time (ms)	385.1 ms	372.5 ms	99.7 ms

Tableau 2: Difference between search times (Second case: 200 vehicles)

The average search time reaches 385ms when the IP model is used and it decreases to 99.7ms when the NDVN with caching model is applied.



(a) CDF of the search time



(b) Search time coupled with the distance between the consumer and data provider (Second case: 200 vehicles)

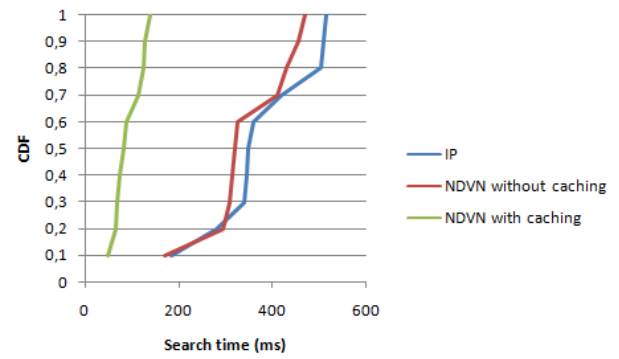
Figure 5: Traffic information search time in the medium density case (200 vehicles)

• *Third case (300 vehicles) :*

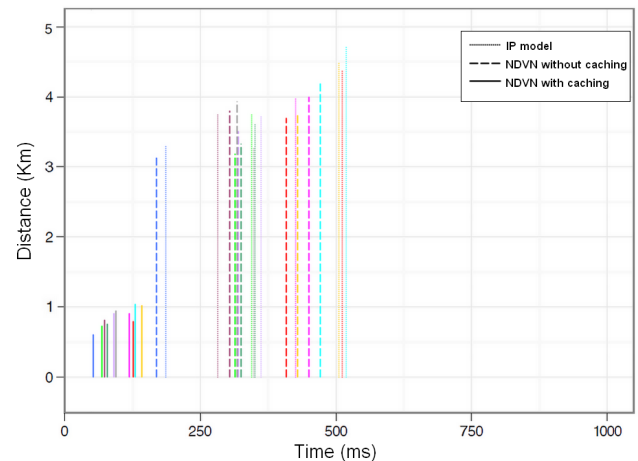
Even for the high density scenario (using 300 vehicles), the simulation results reported in Table 3 and figure 6.a and b show that the scenario 2 (*NDVN without caching*) is more efficient than scenario 1 (*IP model*). Indeed, 80% of tests using the NDVN without caching scenario are faster to meet consumer demands than tests using IP model while the NDVN with caching scenario remains the best for all tests.

Scenarios Tests	IP model	NDVN without caching	NDVN with caching
Test1	350 ms	325 ms	75 ms
Test2	515 ms	470 ms	130 ms
Test3	185 ms	170 ms	50 ms
Test4	280 ms	295 ms	70 ms
Test5	510 ms	410 ms	126 ms
Test6	360 ms	319 ms	85 ms
Test7	340 ms	310 ms	67 ms
Test8	505 ms	430 ms	140 ms
Test9	420 ms	455 ms	115 ms
Test10	345 ms	315 ms	90 ms
Average search time (ms)	381 ms	349.9 ms	94.8 ms

Tableau 3: Difference between search times (Third case)



(a) CDF of the search time



(b) Search time coupled with the distance between the consumer and data provider (Third case: 300 vehicles)

Figure 6: Traffic information search time in the high density case (300 vehicles)

Finally to better show the performance improvement introduced by the NDN inspired solution, we summarized in figure 7, the average retrieval delay of the ten simulated tests as a function of the total number of the vehicles on the highway. Curves in this figure, reveal that for different density patterns, our proposed model keeps its effectiveness in meeting the vehicular communication. We also note, that even if the caching functionality is not activated, the NDN based model achieves better retrieval delay. In fact, even if the requester gets the data from the same node in scenario 1 and scenario 2, the NDN communication process is faster than the IP one.

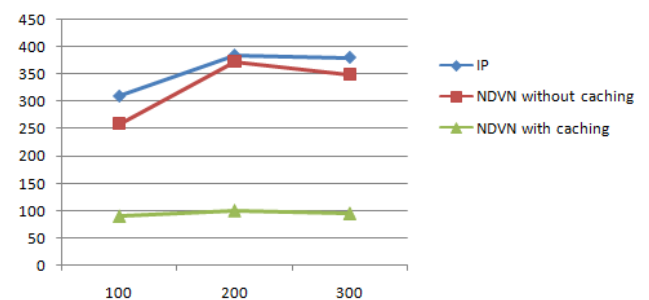


Figure 7: Average Search time for different vehicles density scenarios

V. CONCLUSION AND FUTURE WORK

In this paper we have explored the potentialities of the named data networking architecture in vehicular ad hoc networks. Communication in our proposed solution is based on Interest/Data exchange that follows the NDN framework enhanced with a customized naming scheme. To validate our proposal we developed it as an NDN simulation module under the Network Simulator NS3. Achieved results confirm that the NDN based communication model performs better than traditional IP paradigm and that the proposed naming enhancement is effective and efficient and fit well with the requirements of the considered traffic information application along a highway. Future work will be devoted to evaluate performance under different applications and scenarios (including urban environment).

REFERENCES

- [1] Named data networking, <http://named-data.net>.
- [2] Ning Ye, Zhong-qin Wang, Reza Malekian, Ying-ya Zhang, and Ruchuan Wang, A Method of Vehicle Route Prediction Based on Social Network Analysis, *Journal of Sensors*, pp1-9, 2015
- [3] J. B. Kenney, Dedicated Short-Range Communications (DSRC) Standards in the United States, in *Proceedings of the IEEE*, vol. 99, no. 7, July 2011, pp. 11621182.
- [4] A. Rowstron and G. Pau. Characteristics of a vehicular network. University of California Los Angeles, Computer Science Department, Tech. Rep, pages 090017, 2009.
- [5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, Networking Named Content, in *ACM CoNEXT*, December 2009.
- [6] L. Zhang et al., Named Data Networking (NDN) Project, in *PARC Technical Report NDN-0001*, 2009.[Online].
- [7] V. Jacobson et al. Networking named content. In *Proc. of CoNEXT*, 2009.
- [8] Giulio Grassi Davide Pesavento LucasWang Giovanni Pau Rama Vuyyuru RyujiWakikawa Lixia Zhang, Vehicular Inter-Networking via Named Data ACM, *SIGMOBILE MCCR*, vol. 17, pp. 2324, November 2013
- [9] Giulio Grassi, Davide Pesavento, Giovanni Pau, Rama Vuyyuru, Ryuji Wakikawa, Lixia Zhang VANET via Named Data Networking, *IEEE INFOCOM Workshop on Name Oriented Mobility (NOM)*, Toronto, Canada, April-May 2014
- [10] MinChen, DungOngMau, YinZhang, TarikTaleb, Victor C.M.Leung VENDNET: Vehicular Named Data NETWORK, *Vehicular Communications*, 2014
- [11] W. Quan, C. Xu, J. Guan, H. Zhang, L.A. Grieco, Scalable name lookup with adap-tive prefix bloom filter for named data networking, *IEEE Commun. Lett.* 18(1) (January 2014) 102105.
- [12] K. Zheng, F. Liu, W. Xiang, X. Xin, Dynamic downlink aggregation carrier scheduling scheme for wireless networks, *IET Commun.* 8(1) (Jan. 2014) 114123.
- [13] Y. Yu, T. Punhaole, M. Gerla, M.Y. Sanadidi, Content routing in the vehicle cloud, in: *Military Communication*, vol.2012, Nov. 2012, pp.16.
- [14] OPNET Modeler, available: www.opnet.com.
- [15] Marica Amadeo, Claudia Campolo, Antonella Molinaro Content-Centric Networking: is That a Solution for Upcoming Vehicular Networks?, *ACM, VANET12*, June 25, 2012
- [16] IEEE802.11 working group. IEEE P802.11pTM/D3.0 Draft Standard for Information Technology - Amendment 7: Wireless Access in Vehicular Environments. In *ANSI/IEEE Std 802.11, 1999 Edition (R2007)*, July 2007.
- [17] S. McCanne and S. Floyd. ns Network Simulator. <http://www.isi.edu/nsnam/ns>.
- [18] A. Afanasyev, I. Moiseenko, and L. Zhang, Developing NS-3 based NDN simulator, *CCNx Community Meeting held at PARC*, September 2011. [Online]. Available: <http://www.ccnx.org/ccnxcon2011program>
- [19] C. Campolo, A. Molinaro, C. Casetti, and C.-F. Chiasserini, An 802.11-based MAC protocol for reliable multicast in multihop networks, in *Proc. of Vehicular Technology Conference*, april 2009, pp. 15.