DMND: Collecting Data from Mobiles Using Named Data

Jiangzhe Wang
Computer Science Department
University of California, Los Angeles
lucas@cs.ucla.edu

Ryuji Wakikawa
Toyota Infotechnology Center
Mountain View, CA
ryuji@us.toyota-itc.com

Lixia Zhang
Computer Science Department
University of California, Los Angeles
lixia@cs.ucla.edu

***Abstract*—Technology advances in both computations and wireless communications have made it economically feasible for manufacturers to collect data from all the cars in order to monitor their operations and detect any potential problems. However to make this a reality requires a new architecture that can effectively handle vehicle mobility, intermittent connectivity, and data security, as well as scale to large number of vehicles. In this paper we address these design challenges by exploring the direction of Named Date Networking (NDN) (aka CCN[1]). We evaluated our design, dubbed DMND, through simulations in Qualnet. Our results show that, when data publishers (vehicles) are stationary, more than 99% of collection requests can successfully pull data packets back; even when vehicles move at a high speed of 40-50 meters per second (89.48-111.8 miles/hour), DMND can still retain its high efficiency of 97% of data replies. In contrast, under the same simulation experimental setting, the request-reply ratio of MobileIP drops from 97.9% for static publishers to 9.6% when publishers are moving at a speed of 10-20 meters/second (22.37-44.74 miles/hour).**

***Index Terms*—Communication Systems; Protocols**

I. Introduction

Nowadays more and more mobile devices get connected to the Internet, bringing great opportunities for new Internet applications. Vehicle manufactures desire to monitor the operation conditions of their released cars in order to detect any potential problems, as well as to collect information from cars for traffic congestion maps and weather maps, etc. to achieve distributed sensing, known as probing system. In the past network connectivities to vehicles were accomplished either through low-speed cellular networks or over satellite channel for a small number of luxury cars. Today the throughput of wireless links between vehicles to base stations is much higher than before. Vehicles are also being equipped with more powerful communication devices, such as WiFi and 3G/4G cards. Additionally, Federal Communications Commissions (FCC) has allocated 75 MHz of spectrum in the 5.9GHz band for WAVE/DSRC technology [2] used for vehicle-to-vehicle communications.

Although the existing TCP/IP architecture has been a great success to interconnect multiple millions of stationary hosts around the world, it faces great challenges to meet the needs of vehicular communications. In today's practice, each host is assigned an IP address; when a host moves, it must obtain a new IP address from its new location. Within the network, routing protocols build a single best path between any pair of communicating hosts. However, when end-hosts are mobile and thus their network connectivities become intermittent, the traditional session-based communication is no longer the most appropriate model for networking and information sharing.

Consequently a plethora of proposals for a new Internet architecture has emerged in recent years (see [1][3][4][5] as a few examples). Among these new proposals, a widely shared vision is to make data a first-class entity in the architecture. Thus different from the current Internet architecture, data is able to stand by its own independently from its container or any ongoing end-to-end sessions.

In this work we choose to explore the Named Data Networking (NDN) proposal to design new protocols for vehicle data collection. Different from approaches such as DONA [5] and PSIRP [6] which name data using cryptographic-based flat identifiers, NDN assigns each piece of data a name that can be directly used by the applications. One beauty of naming data in such a way is that applications can request data that may or may not have been produced yet, and the requests will be honored as soon as desired data becomes available [7]. Applications such as RSS [8] and Linda [9] are existing evidences showing the usefulness of this approach. In addition, the network can use the application names directly for data communication, eliminating the need for any mapping system between application names and flat identifiers as required by DONA or PSIRP. Furthermore, with each piece of data standing on its own, one can secure the data directly instead of securing its containers. Thus a requested content does not have to be delivered directly from its originator to the data requester, as long as the latter has effective means to verify the integrity and provenance of incoming data. These properties provided by the NDN design essentially eliminate the requirements of (1) each mobile must obtain an IP address in order to be connected, and (2) requesters and data publishers must be online simultaneously for a network communication to happen.

Following the NDN direction and expanding the initial design as described in [1], we designed a highly efficient, reliable and secure vehicle data collection system, DMND, as describe in Section III. We conducted preliminary evaluation of the DMND design through simulation in Qualnet, and compared the DMND performance with that of a system

[1] This new Internet architecture model was originally called Content-Centric Networking [1], it has been renamed to NDN recently.

978-1-4244-9525-2/10/$26.00 ©2010 IEEE

running MobileIP. Our experience from the DMND design and evaluation process can be summarized as follows.

- Data communication by names and utilization of redundant paths relieve routing protocols the burden of figuring out *exactly* where the requested data may reside, and gives the network the power to utilize all available physical channels to get the data, making communications highly efficient and robust in the presence of vehicle mobility.
- Caching of requests (NDN interest packets) and data mask intermittent connectivity of vehicles, which is the key to support delay-tolerant/disruption-tolerant applications.
- When data publishers are mobile and network connectivity is dynamic and intermittent, communication security can be achieved in a simple and straightforward way by securing data itself, rather than the communication channels.

The rest of this paper is organized as follows. Section II introduces the background of NDN and MobileIP. Section III illustrates our DMND design in details. We then evaluate the DMND performance and data collection efficiency in Section IV and discuss design trade-offs in Section V. Finally we differentiate our work with related projects in Section VI and conclude in Section VII.

II. Background

In this section, we briefly describe the basic concepts in the Named Data Networking and MobileIP; the latter is used as the baseline in comparison with the DMND design in section IV.

A. NDN Overview

Jacobson *et al* presented the initial design of Named Data Networking in [1]. We use a simple NDN topology as shown in Figure 1 to illustrate NDN's major components in the context of a mobile setting: R1-R4 are NDN routers. B1 and B2 represent two base stations connected to R1 and R3, respectively. m1 and m2 are two vehicles within the communication range of B1 and B2. On the top right of the figure, R2 is connected to a database server which collects data from mobiles such as m1 and m2.

Every NDN router contains three major components: Content Store (data cache), Pending Interest Table (PIT) and Forwarding Information Base (FIB), as shown in Figure 2. Data communication in NDN follows a 3-step process: routing announcement from (potential) data sources, forwarding of Interest packets which are originated by data requesters, and data flow from the data source to the requester. In the example system of Figure 1, the whole system starts by base stations B1 and B2 announcing name prefix *ndnx:/vehicle-data/* to the network in anticipation of passing by mobiles that produce data. Similar to how routing protocols work today, R1 and R3 in Figure 1 receive the name announcements and forward to their neighbors. As a result, in Figure 2 R1 adds to its FIB the name prefix associated with interfaces f0 and f2, where the prefix announcement is received from (f2 is due to receiving the announcement from R3). Because the database server wants to receive data from mobiles, it sends out an *Interest* message with the name *ndnx:/vehicle-data/s1* to the network. The trailing part of the Interest name serves as a sequence number. After a NDN router, say R2, receives the interest packet, it would first look up its Content Store to see whether there is already cached data with the same name that can satisfy the interest. If the router finds one, a response will be issued immediately with the cached data. Otherwise, the router checks its PIT to see whether it has already forwarded an interest with the same name. If yes, the incoming interface of the interest will be inserted into the matching name entry in the PIT table, to wait for requested content by the previously forwarded interest. In our example, where all the Content Store and PIT tables are assumed to be empty, the interest message I1 generated from the database server would be forwarded along paths R2-R1-B1-m1 and R2-R4-R3-B2-m2. Note that R1 will also receive an interest message from R3 since it announced the name prefix *ndnx:/vehicle-data/* to R3 before. The duplicate interest will be put into R1's PIT as explained above.

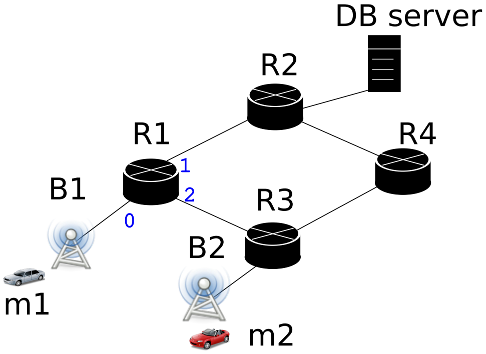
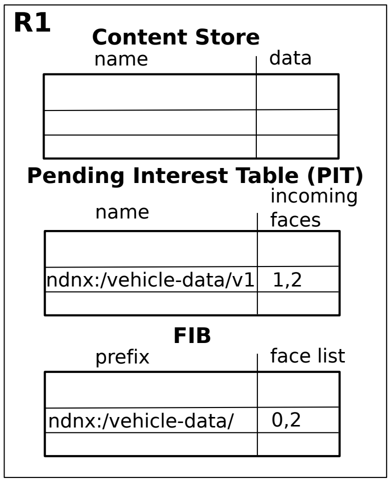


Fig. 1. A sample topology of NDN



R1

Content Store

name	data

Pending Interest Table (PIT)

name	incoming faces
ndnx:/vehicle-data/v1	1,2

FIB

prefix	face list
ndnx:/vehicle-data/	0,2

Fig. 2. Components of NDN router

Figure 2 is an illustration of the NDN router R1's structure after it received two interest packets, and the requested data is yet to arrive. NDN routes *interest* packets, but not *data* packet. A *Data* packet is delivered back to original requesters by traversing the paths following the PIT state that the corresponding interest packet has laid. When m1 sends a data packet as a response to the internet, the packet arrives at R1, and two separate copies are sent out: one through f1 to R2, and the other through f2 to the path R3-R4-R2, due to the PIT entry's memory of incoming interfaces f1 and f2. After data is sent out from R1, the corresponding PIT entry is removed. Each NDN router may also cache incoming *Data* packets in its Content Store until they become obsolete. Though R2 receives two duplicate copies of the same content, one from R1 and the other from R4, it is able to detect the duplication by comparing the data names. Therefore there is no data forwarding loops in a NDN network.

In summary, NDN proposes a named-data communication paradigm which focuses on delivering *what* data users want, rather than nailing down exactly *where* to get the data. NDN uses human-readable names as the primitive of network communications and decouples data from its topological location.

B. MobileIP

We introduce the mobility support of IPv6 in this section, which is used as baseline of evaluation for our DMND design in section IV. MobileIP [10] is an extension to handle end-host mobility. In TCP/IP, when a node moves from one subnet to another, its old IP address becomes topologically incorrect and it's required to be assigned a new address within the new subnet. Therefore packets destined to the previous network location will be dropped, and any on-going services will be broken. This is the long known problem that an IP address binds a host's identity and its network location [11]. MobileIP solves the problem by introducing the concept of permanent address and care-of address, and thus decoupling the two roles of an IP address. The permanent address represents a host's identity no matter where it is in the Internet, whereas the care-of address is dynamically assigned while the mobile is roaming in a foreign network, and thus identifies its network location. A special box called Home Agent (HA) inside the mobile's home network takes care of the mapping between permanent address and care-of address. Remote entities communicate with a mobile using its permanent address, and therefore all the packets will be destined to the home network first. The HA would receive all the packets and replace the destination address in their IP headers with the care-of address, whereby they will then be delivered to the mobile. When the mobile changes from one subnet A to another B, Foreign Agent (FA) routers in the new subnet B will notify the mobile's HA with the most up-to-date care-of address so that the HA can update its mapping table from the permanent address to the care-of IP address.

III. System Design

We start with the requirements for our data collection system in Section III-A. Then we discuss our assumptions on the devices and their capabilities in the system in Section III-B, followed by in-depth description regarding how the system works in Section III-C and Section III-D.

A. System Requirements

Our DMND design aims to fulfill the following requirements, which are listed according to their order of priority. Realization of requirements listed in the latter part depends on the success of the ones beforehand, but they do not necessarily mean of less importance.

1) **Scope specification**: the system should provide mechanisms so that the database server(s) can easily specify the scope of a data collection process. The database server must be able to specify a topological scope it's interested in collecting data from. And very often the server needs to specify a certain type of device in a data collection study, such as vehicles of an indicated model released in a certain year.
2) **High data collection efficiency**: the system should be able to achieve high data collection efficiency. That is, majority of data collected by mobiles should be able to successfully delivered to the database server in a timely manner. Otherwise if content delivery ratio through the network is low, it's a waste of efforts and resource collecting data at the mobile side.
3) **Robustness to high mobility speed**: the system is desired to retain high data collection efficiency when mobiles are moving at a speed as high as around 100 miles/hour.
4) **Scalability to large number of mobiles**: When the number of mobiles conducted in a study process increases, the overall performance of the system should be able to evolve accordingly. That means the system design should be prepared to incorporate an increased number of database servers in large-scale collection scenarios.
5) **Verification of incoming content**: The database server should be able to verify integrity and provenance of incoming content to prevent malicious reporting of fake data.
6) **Protection of users' privacy**: The data collection process is highly desirable to protect user's privacy as they upload collected data through public network to the database server so as to prevent content being used for malicious purposes.
7) **Mitigation to DDos attacks**: The design should be prepared for potential DDos attacks and mitigate DDos to the maximum extent.

B. Assumption of device functionalities

Currently NDN has not been rolled out in large scale yet. But as mentioned in [1], it can be incrementally deployed as an overlay to the current IP network in a bottom up manner. For instance, hundreds of users request for the same video streaming from a Youtube server, edge networks have the most straightforward incentive to deploy NDN in order to reduce upstream traffic volume and improve end users' content request performance through caching. We believe the assumptions made below are reasonable 10 to 20 years down the road after several rounds of hardware upgrade circles.

- Database servers are capable of expressing NDN interests and handle incoming content.
- NDN routers have been deployed on edge networks in a global scale.
- A base station may be able to run NDN functions, or could only serve as a layer 2 device if the access router directly connected to it runs NDN.
- Special NDN boxes, provided by vehicle manufacturers, are installed at home, and serve as the first access point for mobiles within home network.

C. Routing Announcements

Instead of IP prefixes, routing announcements in NDN carry reachability information of name prefixes, such as ndnx:/vehicle-data/ in the previous example. Here we use Toyota as an exemplar company who wants to collect diagnosis information from vehicles. Toyota can talk with network providers to have their base stations announce Toyota's name prefixes. The name prefix originated from base station 1 could appear as *ndnx:/toyota/diagnosis/us/ca/mountain-view/bs-1/*, and similarly that from base station 2 could be *ndnx:/toyota/diagnosis/us/ca/mountain-view/bs-2/*. Each name prefix has a hierarchical structure, and the '/' character represent delimiter between different components. In case the base stations 1 and 2 are connected with the same access router, the two routing announcement entries would be aggregated as *ndnx:/toyota/diagnosis/us/ca/mountain-view/* and then be propagated in the network. It's possible to aggregate the prefix further with similar ones, say *ndnx:/toyota/diagnosis/us/ca/los-angeles/* into *ndnx:/toyota/diagnosis/us/ca/*. Aggregation hides details in edge networks, and facilitate routing scalability in the Default Free Zone(DFZ), where the prefixes could appear as simple as *ndnx:/toyota/*.

Similar with BGP [12], NDN routing announcements play the role of configuring FIBs, so as to route incoming *Interest* messages. However, from a router's perspective, the semantic difference between an IP FIB entry and a NDN entry is that packets destined to an IP address could only be forwarded out along one single interface, because **1)** there is only one unique Ethernet card identified by the IP address[2]; **2)** forwarding a packet along sub-optimal paths may possibly create routing loops in the network, whereas in NDN, **1)**multiple data owners could possibly satisfy an *Interest*; **2)**as explained in section II, PIT and Content Store are able to suppress duplicate Interest and Data messages, and therefore avoid routing loops. So as a result of routing announcements, FIB entries in NDN can be associated with multiple out-going interfaces. Moreover, there might exist bidirectional forwarding relationships between two adjacent routers. Take the example of Figure 1, both R1 and R3, who are capable of forwarding *Interest* messages to some content provider, announce the same prefix to each other. So *Interests* with the same name will be forwarded in both directions: from R1 to R3 and from R3 to R1.

D. Interest and Data messages

We assume that data collectors know the structure of name prefixes, i.e. *ndnx:/toyota/diagnosis/us/ca/mountain-view/*, *ndnx:/toyota/diagnosis/us/ca/los-angeles/* etc. Scope of a data collection process can be easily specified, thanks to the rich expressiveness of NDN naming. For instance, the database server in Toyota is only interested in collecting diagnosis information from Prius(a car model) released in 2009, and only from those in California. The interest name can be expressed as *ndnx:/toyota/diagnosis/us/ca/*/prius/2009*, wherein the character '*' serves as a wild card to match any city name right after 'ca' in name prefixes. The interest will be propagated to all base stations in California that have made routing announcements before, and then broadcasted to vehicles within base stations' wireless coverage. Interests received by vehicles could be further propagated to nearby cars through Dedicated Short Range Communications(DSRC) channel [13] [2]. Forwarding of interest messages between nearby vehicles extends Internet connectivity: in case one car is going through a tunnel and other vehicles outside the tunnel within line-of-sight communication distance offer to forward *Interest* messages for the car, it doesn't have to go off-line and very likely break on-going services. Another advantage of NDN is its capability to fully utilize the broadcast nature of wireless channels. An *Interest* message broadcasted by a base station hunts for content from nearby vehicles, and therefore it's not destined to a specific destination. In this sense, NDN fully utilizes wireless channels, saving the effort of transmitting duplicate *Interests*, each for a different vehicle.

Upon receiving an *Interest* message, NDN router function in vehicles would provide it to application layer, where it's determined whether the vehicle is actually a 2009 Prius and if yes, how to produce requested content accordingly. Although other car models may also receive the interest, application layer in these vehicles should decide not to respond with data.

E. Long-lived interest

Due to the intrinsically unreliable transmission property of wireless channels, *Interest* messages broadcasted from base stations are subject to loss. In order to overcome this difficulty and also mask intermittent connectivity, we introduce the concept of long-lived *Interest*. In the original design, pending Interests in PIT tables would time out if there were no incoming content for a period of a few times of Round Trip Time(RTT). But in the DMND design, we propose to

1) increase the timer for PIT entries in intermediate routers and
2) have base stations broadcast a pending Interest several times before timing it out.

[2] A private address defined in RFC 1918 is not globally unique, and identifies an interface in a local network behind a Network Address Translation(NAT) box.

The con for long-lived *Interests* is that they hold slots in PIT for longer time. And when PIT tables are full, subsequent *Interest* messages will be dropped.

F. Security

Securing content is a very important goal of the DMND design. As stated in the 4th and 5th design requirements, The system must not only allow data collectors to verify integrity and authenticity of incoming content, but also protect privacy of mobile users from malicious use their collected data. NDN requires each content to be tagged with its publisher's signature. Additionally in the DMND system, we require all content to be encrypted using the public key of the database server(s) before being replied back from mobiles. And we assume the data collector has access to each mobile's public key. This is reasonable for vehicle manufacturers as they could not only record public keys of vehicles, but also store the public key of the database server inside vehicles before release. An alternative approach is to use DNSSEC [14] for public key distribution. In this way, though data packets are cached in *Content Store* of intermediate routers, and anyone expressing interests is able to pull data back, the data is useless to a third party since they won't be able to decipher received content.

Another important property of the DMND design is that the server is able to mitigate DDos attacks and govern incoming data flow by controlling the number of interest messages sent out. The publish/ subscribe model of NDN design says if the server does not send *Interest* messages out, it will not receive any data. Therefore data collection servers are able to prevent DDos attacks through *Data* messages. For those who want to attack vehicles through *Interest* messages, the minimum PIT table size between the attacker and vehicles sets a maximum limit the the attacking effect. Not only that, memory of incoming interface for *Interest* packets in PIT tables provides an way to the trace the *Interest* originator's network location, making it a big concern for attackers before launch large-scale DDos attacks.

IV. Simulation

In this section, we evaluate our design through simulation in Qualnet [15]. We compare DMND with an alternative Mobile IP solution in similar settings to show its advantage in high efficiency of data collection and resiliency in handling node mobility. The simulation topology is shown in Figure 3, where nodes 1 through 16 are laid out in grid and serve as access points(APs). Each AP establishes a wireless network that covers its vicinity and is the last hop communication between mobiles and the Internet. Adjacent APs are connected with wired Ethernet to simulate core of the network. The distance between two adjacent APs is 450m, roughly the transmission range of 802.16(WiMax) in Qualnet. Node 18 simulates a database server that periodically send requests to the network in order to collect data from mobiles. In the following subsections, we use the following metric to quantify data collection efficiency(dte) of MobileIP and DMND:

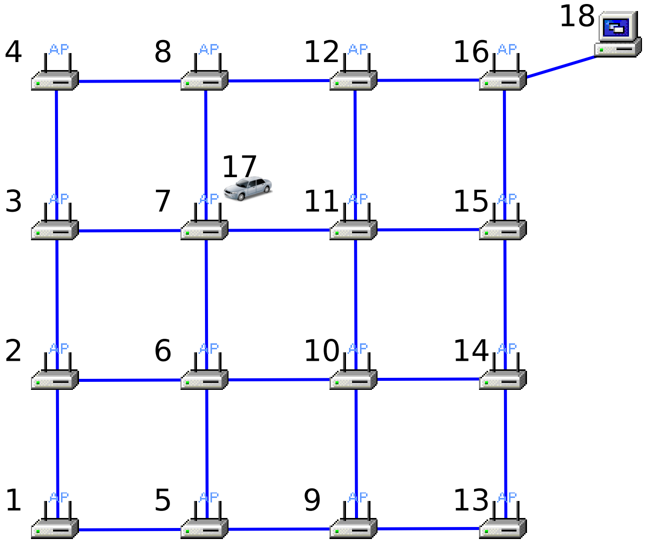


Fig. 3. Simulation Topology

$$dte = \frac{\text{Number of reply packets received by a requester}}{\text{Number of request packets sent by the requester}}$$

A. Data collection through Mobile IP

1) Single static data publisher: In this section we first evaluate the efficiency of data collection through MobileIP, where there is only one static device in the network, denoted by node 17 in Figure 3. The application layer traffic is simulated through Qualnet SuperApplication. It's configured as a two way flow traffic where data requester (Node 18) sends requests to the publisher (Node 17), and the publisher sends reply packets back to the requester. Home agent(HA) of the device is coined to be Node 1. We use UDP to deliver request and reply packets. In the first scenario, the publisher is located near AP 7, away from its home agent node 1. A total number of 500 requests are generated and sent out from the requester(Node 18), each with fixed packet size of 1460 bytes. The time interval between adjacent requests is 5 seconds. If a request successfully arrives at the publisher, a reply packet of the same size will be scheduled to be sent back after a delay of 1 second.

In order to make the measurement result more confident, we run experiments in the same topology for 20 times, each with a different seed for random event generator. Figure 4 shows the simulation result of the 20 experiments. Take the first experiment as an example, the number of request packets sent by the requester was 500, among which the publisher received 490. In response to requests, the publisher sends back 490 data packets, and the requester received all of them. Note that for each request handled in application layer of the mobile publisher, exactly one reply message is sent back, so the number of replies sent back is the same as that of requests received by the publisher. We show the number of bytes sent out and received by both the requester and the

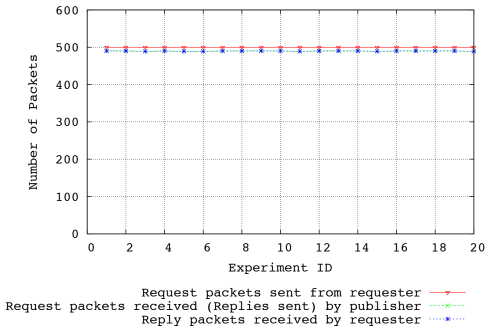


Fig. 4. (MobileIP) Packets sent and received when publisher is static

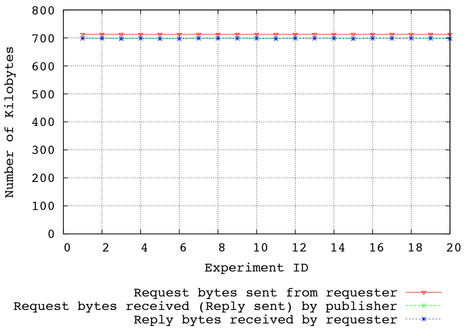


Fig. 5. (MobileIP) Bytes sent and received when publisher is static

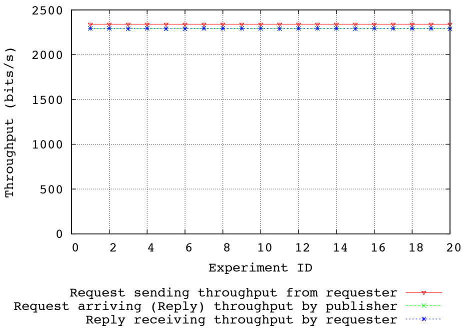


Fig. 6. (MobileIP) Throughput sent and received when publisher is static

Speed	Requests sent by requester	Requests received by publisher	Replies received by requester	dte
0	500	489.7	489.7	97.94%
0-10	500	227.4	219.85	43.97%
10-20	500	60.65	48.1	9.62%

TABLE I
AVERAGE NUMBER OF PACKETS SENT AND RECEIVED IN MOBILEIP

Speed (mps)	Requests sent by requester	Requests received by publisher	Replies received by requester	dte
Static	500	496.4	495.9	99.18%
0-10	500	495.6	491.7	98.34%
10-20	500	497.3	491.6	98.32%
20-30	500	496.3	490.15	98.03%
30-40	500	496.6	489.65	97.93%
40-50	500	497	489.35	97.87%

TABLE II
AVERAGE NUMBER OF PACKETS SENT AND RECEIVED IN DMND

publisher in Figure 5. In the case of experiment 1, a total number of 730,000 bytes(about 712 Kbs) of request were sent out from the requester. The publisher successfully received 715,400 bytes(about 699 Kbs) contained in request packets, and sends the same amount of reply packets back, of which the requester received all the bytes. Since the request and reply packets have the same size(1460 bytes), curves in Figure 5 exhibit the same shape as those in Figure 4. Also because request packets are pushed to the network at a constant rate (1 packet every 5 seconds) and reply is sent back after a fixed time interval (1 second), the throughput curves in Figure 6, in different unit, retain the same shape as those in Figure 5. Therefore, we'll use the number of packets as an indicator of data collection efficiency in the following scenarios. As shown in the first row of Table I, when the publisher is static, on average the requester would receive 489.7 replies for the 500 request packets sent out, the *dte* of MobileIP is 97.94%.

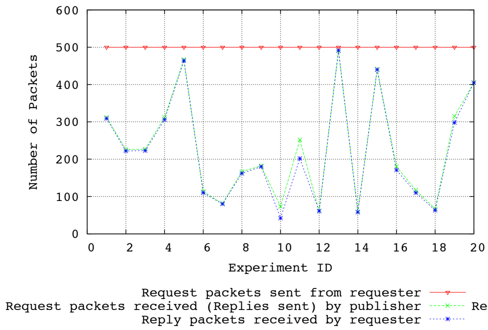
2) Single mobile data publisher: In this section we quantify the MobileIP's performance in handling mobility of the data publisher. In the same topology of Figure 3, when the publisher was moving at a speed between 0 to 10 meters/second according to random-way mobility model, we see a lower efficiency of MobileIP than when it's static as shown in Figure 7. On average of the 20 experiments, the *dte* dropped to 43.97%, as in Table I. And as the mobility speed of the publisher increased to 10-20 mps, the *dte* dropped even further to as low as 9.62%.

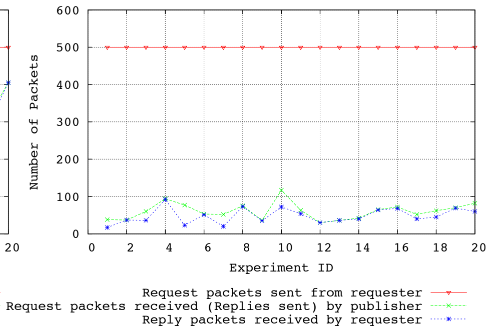
We believe the reason for low *dte* of MobileIP in handling mobility are due to the following two reasons, and we'll explain how DMND get around them in the next section IV-B. They explain why more than half of request packets are lost before they reach the publisher,

- It's well known that MobileIP suffers from the problem of triangle routing. Each request destined to the data publisher (Node 17) has to be delivered to its home agent (Node 1) first. More hops of transmission makes the request packets subject to a higher probability of being dropped.
- While the mobile publisher moves from a base station to another, there is a delay before it can be connected to the Internet again. The delay includes the time of assigning an new IP address to the mobile, as well as the time for foreign agent in the new subnet to notify the mobile's HA of its up-to-date care-of address.

B. Data collection through NDN

1) Single mobile data publisher: In this section, we evaluate the performance of DMND under different mobility settings, and show its advantage in high efficiency and robustness in handling node mobility in the same topology of Figure 3. NDN forwarding table(FIB) on each node is configured so that an name prefix entry ndnx:/toyota/ points to all available interfaces. Take the example of node 16, it has three wired interfaces and a wireless one. That means if it received an interest with name *ndnx:/toyota/diagnosis/ca/*/prius/2009* from node 18 and there were no interest of the same name in its PIT, node 16 would forward the interest to node 12, 15 and its wireless network. As a matter of fact, since the reachability of the name prefix ndnx:/toyota/ is configured to be so diverse





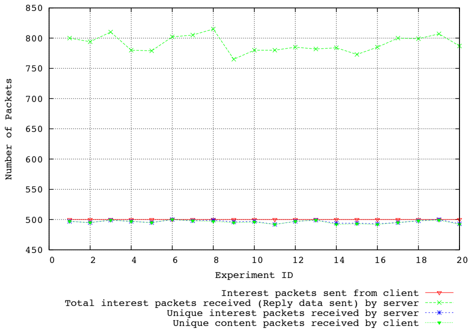
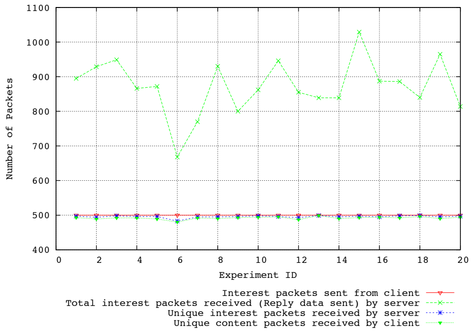
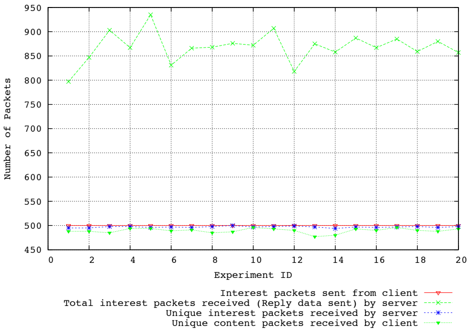


Fig. 7. (MobileIP) Packets sent and received for mobile publisher using random-way mobility model (speed 0-10mps)

Fig. 8. (MobileIP) Packets sent and received for mobile publisher using random-way mobility model (speed 10-20mps)

Fig. 9. (NDN) packets sent and received when publisher is static





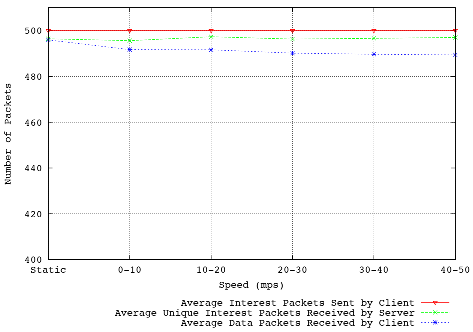


Fig. 10. (NDN) Packets sent and received for mobile publisher using random-way mobility model (speed 0-10mps)

Fig. 11. (NDN) Packets sent and received for mobile publisher using random-way mobility model (speed 40-50mps)

Fig. 12. NDN Packets sent and received as mobility speed increases

that an interest message like ndnx:/toyota/diagnosis/v1 would be flooded in the network. Similar with the MobileIP scenario, node 18 periodically sends out *Interest* messages appended with sequence numbers to retrieve most up-to-date content from the network in a fixed time interval of 5s.

2) Single static data publisher: Firstly, we test DMND in the simplest scenario when the data publisher (Node 17) is stationary and located near AP 7. The simulation result is shown in Figure 9. The requester (Node 18) sends out a total number of 500 interest messages. Because the publisher is within transmission range of three APs, namely node 3, node 7, and node 8. It's likely to receive the same interest message multiple times, therefore the total number of interest messages is more than that sent from node 18. In experiment 1, the publisher received 800 interest messages in total, and 497 of them are unique ones. For the 497 replied data packets, node 18 received all of them. We quantify the *dte* of our DMND design in handling data publisher mobility in this section. Figure 10 and Figure 11 illustrate DMND's performance when the data publisher is moving in random-way mobility model at a speed at ranges 0-10mps and 40-50mps respectively. We show the average number of interest messages sent by the requester, unique interests received by the data publisher, as well as data messages received by the requester in Figure 12 and Table II. Our measurement results show that DMND is highly resilient in face of node mobility and retains a high *dte* of 97.87% even when the mobile is moving very fast at 40-50 meters/s (89.48-111.8 miles/h).

V. Discussions

In the design of DMND, we explored several alternative choices. We studied the approach of adding push message to the NDN design, and collecting content by pushing data directly from mobiles to collectors without *Interest* messages, similar as a one-way UDP traffic. However, the little control of traffic uploading speed decreases data collection efficiency when the number of mobiles is large. Moreover, the ease of launching DDos attacks to data collectors makes the system greatly vulnerable to attackers.

We also explored adding *Capacity Counter* in Interest messages so that one Interest can retrieve back multiple data packets in order to further increase data collection efficiency. But this approach introduces delay into delivery of already available content, and also has the drawback of holding PIT slot for a longer time.

VI. Related work

It has long been realized the importance of collecting data from mobile devices and vehicles. Researchers have been designing systems to collect data from vehicles. In [16], S. Reddy*et al* designed a framework for mobile phone data collections that's used to identify participants appropriate for a collection process based on their geographic and temporal availability as well as participation habits. Their work focuses on collecting and selecting qualified participants based user's behavior, and it's supposed to be built on top of existing Internet protocols. J. LeBlanc *et al* [17] designed and evaluated a data distribution and collection system for police and highway patrol vehicles. The design require vehicles to stop by fuel or troop stations from time to time, where collected data is uploaded to a central data collection server. There is also a rich research literature of Disruption Tolerant Networks(DTN) [18], where continuous end-to-end connectivity cannot be assumed. The basic idea of DTN protocols is to store and forward data without establishing a complete path between a source to a destination beforehand. The DMND design is different from existing projects above in that it's the first exploration to build a data collection system via NDN. While the aforementioned existing work are designed to achieve the task of end-to-end content delivery. our work and NDN proposal is centered at named data as a first-class citizen in network communications rather than its containers.

VII. Conclusion

In this paper, we explore the direction of NDN to build a system collecting data from vehicles. By assigning human-readable names to data and decoupling it from communication channels, our DMND design is able to utilize available physical channels to the maximum extent. Caching of *Interest* and being able to be broadcasted from multiple base stations solves the problem of high-speed mobility, and masks intermittent connectivity from the application layer collection process. Moreover, rich expressiveness of hierarchical names gives data collectors great flexibility to specify scope of a collection process. Finally, our design is able to secure collected content by attaching data packets with mobile devices' signatures and employ public key encryption before delivery in public Internet. Evaluation of the DMND design in Qualnet shows high data collection efficiency, even when mobile devices are moving at a high speed.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *CONEXT*, Rome, Italy., 2009.

[2] T. M. Kurihara, "DSRC Application Layers – IEEE P1609," in *IM P1512 WG Meeting*.

[3] M. Meisel, V. Pappas, and L. Zhang, "Ad Hoc Networking via Named Data."

[4] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system."

[5] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenke, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," in *SIGCOMM 07, Kyoto, Japan*.

[6] N. Fotiou, G. C. Polyzos, and D. Trossen, "Illustrating a Publish-Subscribe Internet Architecture."

[7] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, and B. Ohlman, "Design Considerations for a Network of Information," in *ReArch 08, Madrid, SPAIN*.

[8] R. Cadenhead, G. Smith, J. Hanna, and B. Kearney, "The application/rss+xml Media Type."

[9] D. Gelernter and N. Carriero, "LINDA IN CONTEXT."

[10] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," in *RFC 3775*.

[11] J. Saltzer, "On the Naming and Binding of Network Destinations," in *RFC 1498*.

[12] "A Border Gateway Protocol 4 (BGP-4)," in *RFC 4271*.

[13] "DSRC," in *http://en.wikipedia.org/wiki/DSRC*.

[14] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Dns Security Introduction and Requirements."

[15] "Qualnet," in *http://www.scalable-networks.com/products/qualnet/*.

[16] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections."

[17] J. LeBlanc, T. E. Hurton, W. T. Miller, and A. L. Kun, "Design and Evaluation of a Vehicle Data Distribution and Collection system."

[18] K. Fall, "A delay-tolerant network architecture for challenged internets."