A Thesis Submitted to the
Department of Computer Science and Communications Engineering,
the Graduate School of Fundamental Science and Engineering of
Waseda University
in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering

January 30th, 2018

Vehicular communication based on Named Data Networking (V2VNDN)

By
Mahendra Sapkota
(5116FG03-9)

SATO Laboratory
Advisor: Prof. Takuro Sato

Acknowledgments

I would like to express my sincere gratitude to professor Takuro Sato for his advice, support, and guidance throughout my master's degree program. I have greatly benefited from his ideas and motivation towards research activities and thankful for providing me an opportunity to involve in 5G pagoda project. I would like to express the deepest appreciation to all the Sato laboratory members for their suggestions and help to accomplish my task. I want to thank professor Toshitaka Tsuda for guiding me throughout the 5G pagoda project. Finally, lots of love to my parents, siblings, and wife for their continuous support.

Abstract

Vehicular communication has been receiving a lot of attention to enhance the road safety and traffic management systems and is one of the leading approach for many organizations and industries. Similarly, Named Data Networking (NDN) is one of the proposed new internet architecture in which network use named data for content distribution rather than host to host communication. In this paper, we have proposed vehicle to vehicle communication using Named Data Networking (V2VNDN) which is mainly designed to share road traffic information, information about emergency vehicles and safety messages on road condition between two or more vehicles. In V2VNDN, we have used unique naming strategy for each content and assigned the routable prefix which can be broadcasted to all nearby connected nodes where each of the vehicles acts as a node. Since the number of contents is fixed, so it's necessary to broadcast all three contents by each of the nodes. We have compared our proposal with vehicular communication based on TCP/IP protocol stack and regular NDN; and shown the scalability, strong security and feasibility of V2VNDN.To make our proposal effective, we have shown the simulation results and verified the experimental results of V2VNDN for vehicular communication.

Key words: Vehicular communication, Named Data Networking (NDN), Unique naming strategy, Broadcast, Routable prefix**,** NDN contents.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NDN	Named Data Networking
CS	Content Store
PIT	Pending Interest Table
FIB	Forwarding Information Base
V2VNDN	Vehicle to Vehicle Named Data Networking
TCP/IP	Transmission Control Protocol/ Internet Protocol
CCN	Content Centric Networking
ID	Identifier
ndnSIM	Named Data Networking Simulator
ns-3	Network Simulator 3
GNU	GNU's Not UNIX
ICN	Information Centric Networking
SDN	Software Defined Network
VANETs	Vehicular Ad Hoc Networks

Chapter 1

Introduction

1.1 Background

Vehicular communication system is the information exchange between vehicles including road side access points to enhance the road safety and traffic management system. According to World Health Organization (WHO) [1], road accidents annually cause approximately 1.2 million deaths worldwide, one-fourth of all deaths caused by injury. Also, about 50 million persons are injured in traffic accidents. If preventing measure are not taken, the number of deaths and accidents keep on increasing in the future. So, the vehicular communication is the leading approach to make the transportation system safe and intelligent. Similarly, Named Data Networking (NDN) [2] is the networking scenario which deals with the content retrieval instead of point to point communication. So, we proposed V2VNDN [3] for the vehicular communication with the new internet architecture i.e. NDN. Named Data Networking (NDN) is mainly divided into three parts [2] [4] i.e. Pending Interest Table (PIT), Forwarding Information Base (FIB) and Content Store (CS). Each protocol has a different function based on the interest from the consumer side to the producer.

TABLE 1.1

Parts of Named Data Networking (NDN)

Content Store (CS)	It caches and stores received data packets at the router and If the content ID is matched in the content store (CS); data will be forwarded by CS otherwise it will look up at pending interest table (PIT).
Pending Interest Table (PTI)	It will update an existing PIT entry if matching found otherwise it will look up at FIB.
Forwarding Information Base (FIB)	It will create a new PIT entry if matching found otherwise it will discard the interest.

1.2 Issues in existing network

Although TCP/IP is one of the widely accepted networking scenarios, it still has many issues of latency, security, and mobility. An IP address is based on location and provides identity of the user or publisher which is the main cause of security risk. Similarly, user want fixed IP for many systems to keep the session stable which may also cause the security risk and scalability problems. Also packet loss occurs due to high latency and long delay time during the movement of source and destination on wireless environment. Fake source address and spam are the major problems in TCP/IP which are causing many problems to the internet users.

To overcome these issues, NDN is one of the best networking architecture which depends on "what" are the contents instead of "where" are the contents. So, we have shown the comparison of both the networking scenario (i.e. TCP/IP and NDN) and analyze the best networking option for vehicular communication.

1.3 Architecture of V2VNDN

For the vehicular communication, we prepared the three important contents which are necessary to reduce the road accident and to prevent from the emergency. The first is road traffic system, it will provide all the traffic information as well as the required time information for the requested user; and the second is safety message on the road condition like construction or damage on the road; and the last is information about emergency vehicles like ambulance, police car, fire truck, and disaster management vehicles.

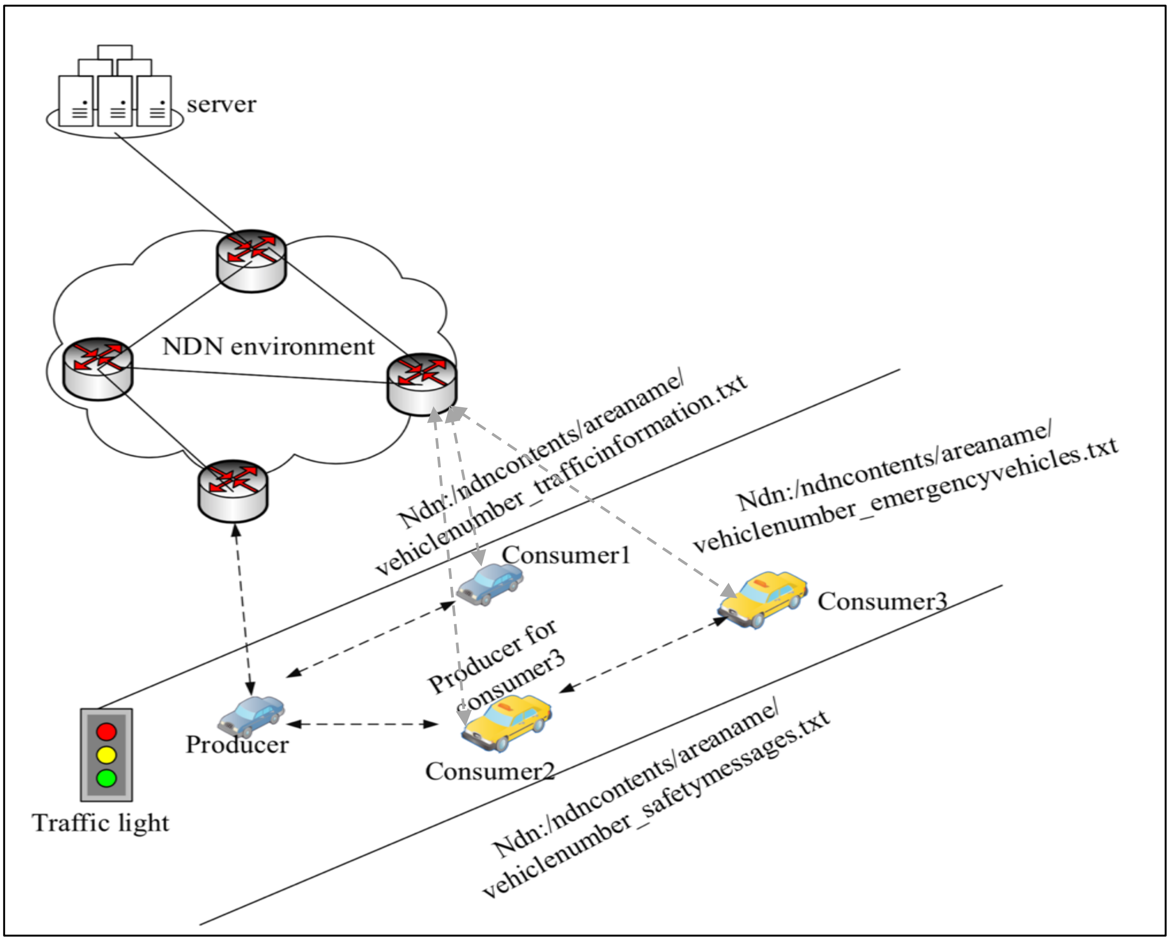

server
NDN environment
Ndn:/ndncontents/areaname/
vehiclenumber_trafficinformation.txt
Ndn:/ndncontents/areaname/
vehiclenumber_emergencyvehicles.txt
Consumer1
Consumer3
Producer for
consumer3
Producer
Consumer2
Traffic light
Ndn:/ndncontents/areaname/
vehiclenumber_safetymessages.txt

Fig. 1.1 Architecture of V2VNDN [3]

Figure 1 shows the architecture of V2VNDN [3] which shows the communication between vehicles with the necessary contents ID. To get the contents from a producer, the consumer can request with a content ID i.e. with repository, area name, and vehicle number followed by a content name. If the content ID is matched in the content store (CS) [5]; data will be forwarded by CS otherwise it will look up at pending interest table (PIT); it will update an existing PIT entry if matching found otherwise it will look up at FIB; it will create a new PIT entry if matching found otherwise it will discard the interest. This is the actual procedure to exchange the contents and interests between producer and consumer. In V2VNDN, the communication takes place between two or more vehicles where the vehicle which sends an interest is a consumer and another which provides the content is a producer.

TABLE 1.2

Interests requested by different consumers

Consumers	Interests
Consumer1	ndn:/ndncontents/areaname/vehiclenumber_trafficinfromation.txt
Consumer2	ndn:/ndncontents/areaname/vehiclenumber_safetymessages.txt
Consumer3	ndn:/ndncontents/areaname/vehiclenumber_emergencyvehicles.txt

In the above table, All the interests are requested by the different consumers to the producers based on their location. In figure 1.1, we have shown that the consumer 1 and consumer 2 are requesting the content to the producer while consumer 3 is requesting to consumer 2. In case of consumer 2 and consumer 3, consumer 2 becomes the producer of consumer 3 based on the area where they are present.

1.4 Organization of thesis

The thesis is organized as follows:

Chapter 1

In this chapter, the background, issues in existing network (i.e. TCP/IP), and architecture of V2VNDN are discussed.

Chapter 2

This chapter summarizes the Literature review including the introduction, vehicle to vehicle communication, Named Data Network (NDN), and related work to V2VNDN.

Chapter 3

Chapter 3 summarizes the vehicular communication on NDN environment.

Chapter 4

Chapter 4 discusses the simulation results and discussion.

Chapter 5

Chapter 5 describes the experimental results of the propose system.

Chapter 6

Chapter 6 summarizes the conclusion of the propose system and the future work.

Chapter 2

Literature Review

2.1 Introduction

At the beginning of this chapter, vehicle to vehicle communication and Named Data Networking (NDN) has been described. Each part is described with their architecture, functions, and classifications. This chapter also include the two important explanations: "why vehicular communication is important?" and "how NDN is better option for V2V communication?" Finally, the related work and comparison of V2VNDN with other modules are discussed.

2.2 Vehicle to vehicle communication

The communication of two or more vehicles by establishing the wireless links between them is called vehicle to vehicle communication. Vehicle to vehicle (V2V) [6] is also defined as the wireless network where automobiles send messages to each other with necessary information. The main motivation of this communication is to reduce the number of accidents and to make the vehicles intelligent. In this system, road side access points and vehicles act as the communication nodes in which one vehicle shares the information of safety messages and traffic information to the another.

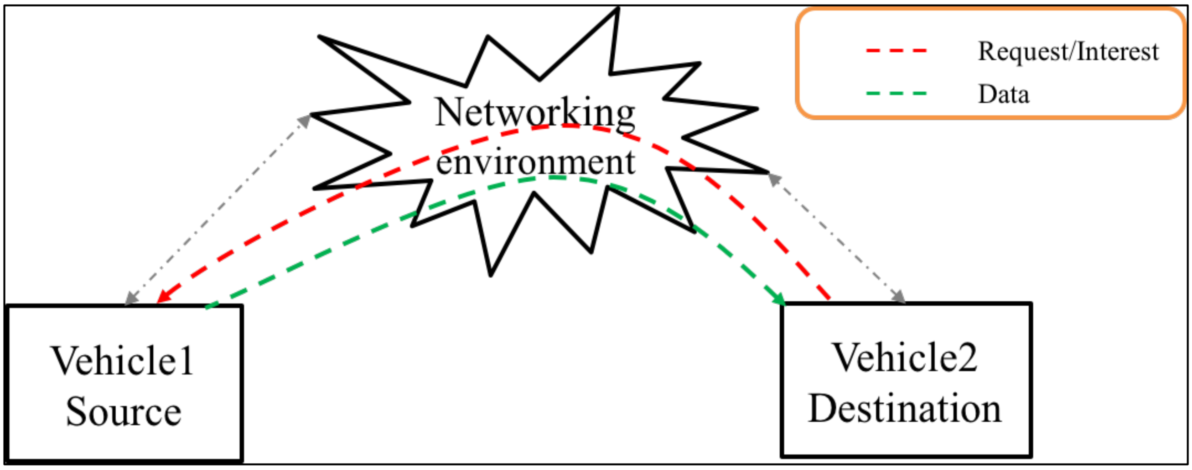


Fig 2.1 vehicle to vehicle communication

Figure 2.1 shows the communication between vehicle 1 and vehicle 2 using networking environment. Vehicle1 and vehicle 2 are the source and destination respectively where destination will request the data or content to the source and source will provide the content based on the request type. This shows the normal communication between vehicles.

2.3 Named Data Network (NDN)

Named Data Network is the new internet architecture which can reduce the weakness of the existing host to host communication architecture and based on the name of data instead of location. NDN is based on the content Centric Networking (CCN) [7] which was presented by Van Jacobson in 2006. In NDN, the contents provider is a producer and the contents receiver is a consumer. At first, all the contents are assigned with unique content ID to each and published by the producer. Then the routable prefix is broadcasted to the different NDN intermediate nodes and edge nodes. Finally, the contents ID table is provided to the consumers and consumers request the content to the nearest edge node. At the initial request, the response time is more for the specific content but if the user request the same content again then the response time is very less. This means NDN is the in-network caching system which stores content in the CS.

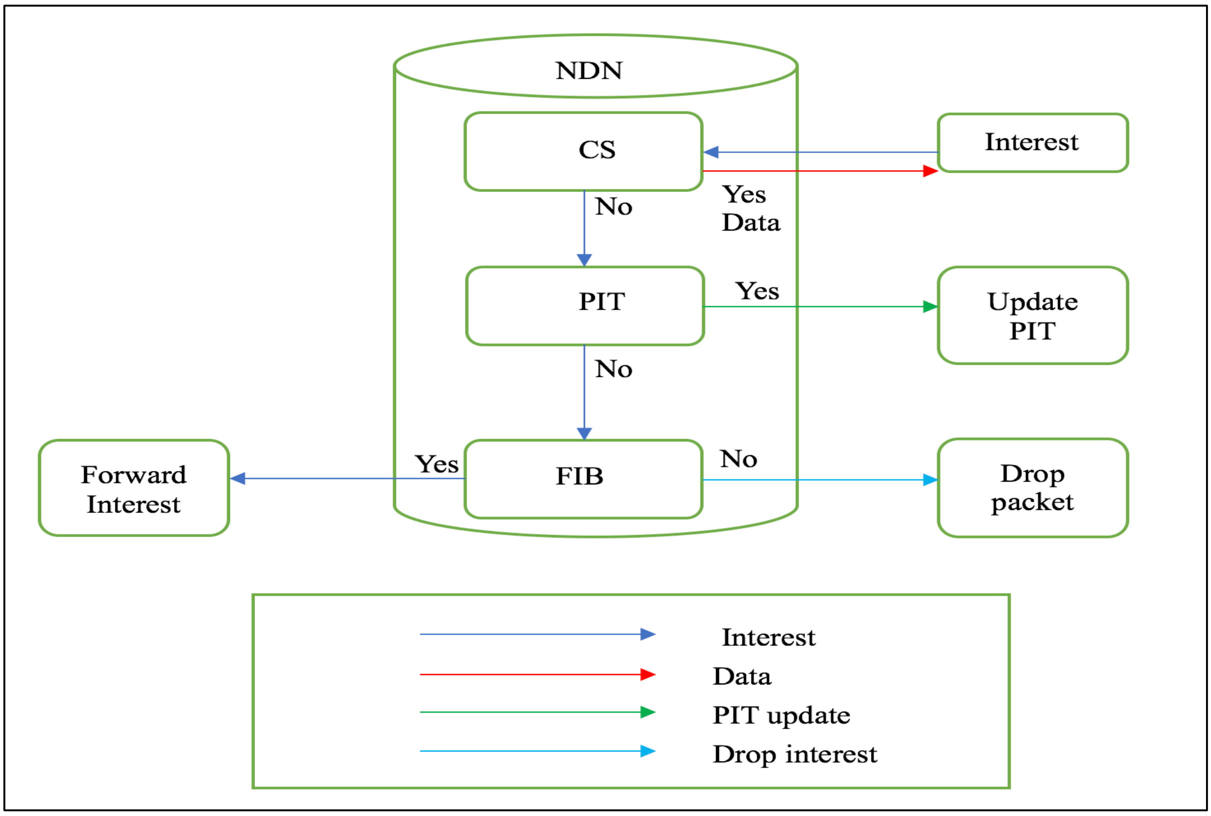


Fig.2.2 NDN node and its forwarding strategy

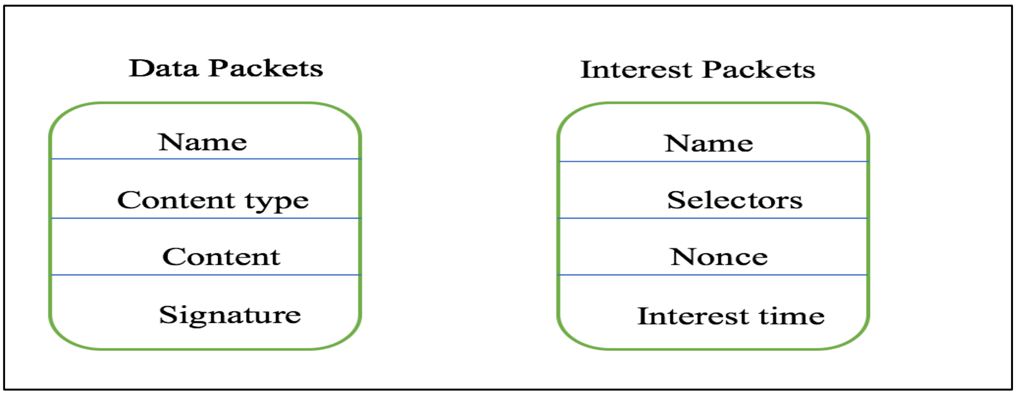


Fig2.3 Interest and data packets of NDN

2.4 Related work to V2VNDN

When we go through the past research papers [8][9], a lot of research has been done for vehicular communication based on existing networking system called TCP/IP but none of them are implemented practically and holds many issues of security, latency, and mobility. Most of the vehicular communication is based on host centric designed IP stacks [10] where the node mobility is complex and mobility issues in vehicular communication and it is hard to predict the movement of producer and consumer. In [11], researcher have proposed peer to peer cooperative caching scheme using Markov chain model and focused more on caching system. They have also discussed about the congestion control and network management using caching scheme. They have evaluated the proposed scheme based on the network congestion, delay and hit ratio but the overall performance is slightly better than the existing schemes and focused more on only caching scheme rather than overall vehicular communication. Similarly, in [12] the author has discussed the security system to secure the vehicular communications and proposed a hybrid security framework. However, the overall discussion and proposed scheme is based on an idea to secure the vehicular communications that occur in vehicular ad-hoc network. To solve this kind of issues, NDN plays an important role since it is location independent network scenario and provides the in-network caching system.

Similarly, emergency application for vehicle to vehicle has been proposed in [13] where the authors have adapted eVNDN for vehicle to vehicle communications which enables vehicles to communicate based on the autonomously distributed contents. The main goal was to provide the emergency applications for communication and they have also changed the PIT and CS to support their system. But the naming strategy, overall network scenario, and interest based on required information has not been shown clearly. In [14] [15], the authors have focused both on NDN and SDN (Software Defined network) and present the use cases for vehicular NDN scenarios and SDN. SDN has been used to solve the problems in VANETs (Vehicular Ad Hoc Networks). They have considered the VANETs that use the NDN architecture and presented the existing problems. To solve the existing issues, they have proposed the SDN and based on the assumptions that the SDN controller having knowledge of the network and vehicular traffic information like density of the networks, location, number of vehicles and their direction and location of contents producer, they have proposed the forwarding strategy at the road side access points. They have proposed this scenario for both vehicle to vehicle and vehicle to infrastructure communications. All the proposed schemes and idea are based mostly on theoretical assumptions rather than simulation and experimental results. So, the implementation of the proposed system is difficult and the system is complex. In [16], authors have presented VANET scenario using NDN architecture and shown the simulation result of throughput, delay and routing overhead. They have used geo-based location strategy for packet forwarding and highlighted the limitations but they have not proposed any specific forwarding strategy to replace the existing issues.

So far, the research community focused mostly on forwarding strategy, pending contents on NDN and host centric networking. In this paper, we have focused towards the overall network scenario of NDN and new naming strategy. We found that interest packets and data packets can travel from each of the vehicles with the content ID which consist of vehicle number to identify the prefix name. Also, we proposed the possible solution of broadcasting the routable prefix of one vehicle to the nearby vehicles. So, our proposal will clearly show the overall network scenario for vehicular communication using NDN.

Chapter 3

Vehicular Communication on NDN Environment

3.1 Introduction

In V2VNDN, we have assumed that each vehicle acts as a NDN node and the vehicular communication takes place between any two vehicles based on their location and position. Following operations take place to transfer interests and receive data:

- The receiving node or the consumer will send an interest to the specific vehicle from which it wants the information. Since each content ID is assigned with the vehicle number, area name, and fixed contents name. so, it’s easy to find out the content ID of a producer by the consumer.

- Once the interest is received by a producer, it will upload and publish the current information about traffic or safety messages immediately for the first user and first time.

- After publishing the data by producer and send to the consumer, the data is cached into the edge node of that area. In that area, the other vehicle can request the same data with the same content ID and get response immediately. So, the time taken to receive the published content is very less than the initial request.

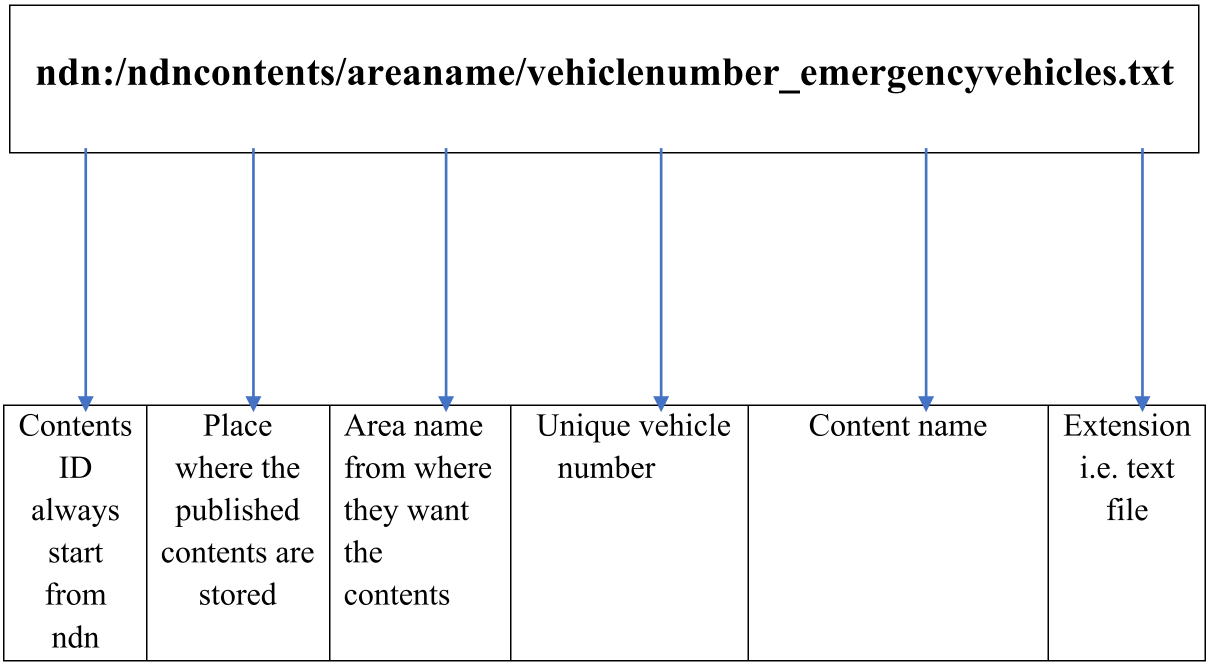
- If the contents are not found in the CS, the PIT table will be updated based the received interest with the face ID.

3.2 Naming process

In V2VNDN, we are mainly focusing on three important contents of different location. The naming of contents is divided into several parts. Each part is important to receive the data from producer (i.e. vehicle). Suppose the vehicle ‘A’ is waiting at some area due to the passing of emergency vehicles and vehicle ‘B’ want to know about the situation which is about 500m far away from ‘A’; ‘B’ will send a request based on the broadcasted routable prefix and receive the data about that area. The structure of naming process is shown below:

TABLE 3.1

Naming strategy in V2VNDN



ndn:/ndncontents/areaname/vehiclenumber_emergencyvehicles.txt

Contents ID always start from ndn	Place where the published contents are stored	Area name from where they want the contents	Unique vehicle number	Content name	Extension i.e. text file

Table 3.1 shows the naming strategy in V2VNDN and description of each part. Publishing the contents and the format of contents ID are listed below:

- The published contents are saved into the repository.
- The producer will publish the content using “ndnpublish” command and consumer will receive the content using” ndnget”.
- After publishing the contents by the producer, the contents ID are distributed to NDN environment and assigned timer to each content. Since the structure of contents ID look same but the contents vary from one area to another based on the condition of that area.
- Since the code base is written in java script, so the user terminal can access the contents easily from one of the browser or browser like environment which is installed in the vehicle.
- According to area or location, the consumer will update the PIT table and receive

data immediately once the data is published by producer.

The below table shows the status of producer and consumer after publishing and receiving the contents:

TABLE 3.2

CONTENTS ID [3]

Producer	Consumer
ndn:/ndncontents/Waseda/24**_trafficinfo.txt var/www/traffic.txt	ndn:/ndncontents/Waseda/24**_trafficinfo.txt received.txt
Status: Inserted file	Status: Retrieved file
Time: published time (in ms)	Time: Received time (in ms) File size: In bytes

In the above table 3.2, the consumer first requests the content of traffic information of Waseda (i.e. area name) and then the producer publishes the file. After publishing the file, the consumer receives the content immediately with the status of retrieved file and producer receives the status of inserted file and time taken to publish the file. The total file size is shown in bytes to the consumer side.

3.3 Publishing content in NDN

At first the original content is divide in several segments and then further divided into several chunks. The size of each chunk is about 4096 bytes which is default size. Time to divide the file into segment and chunks vary with the size. While publishing the file in NDN, the segment is automatically divided into several chunks and named in order. In the experiment scenario, when we publish the content for the first time, we can see all the segments and chunk division but if the same content is published by the same publisher, then the content is inserted very fast and shown the notifications of “replacing the file”.

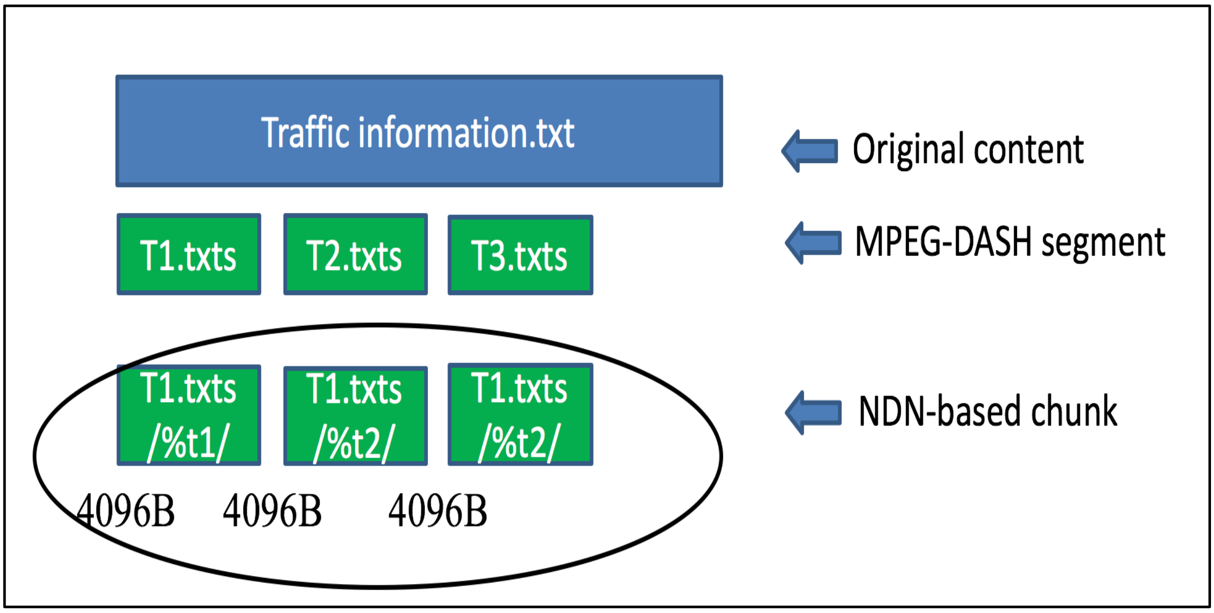


Fig. 3.1 Publishing the content in NDN

Chapter 4

Simulation Results and Discussion

4.1 Introduction

In this chapter, we discuss about the simulation results of proposed system and compare the TCP/IP, regular NDN, and V2VNDN. we have shown the simulation results of V2VNDN by using ndnsim1.0 [17] and ns-3 [18]. ndnSIM is the network simulator used to carry out the simulations of Named Data Network and other network related simulation. ndnSIM is the free software available under the GNU license for research and development. The simulated vehicular network consists of 4 vehicles where node 0 is a producer and other nodes are the consumer. For the experimental scenario, we used NDNx [19] which was developed by NDN project team.

4.2 Simulation environment

In this section, simulation parameters, simulation environment, and discussion of simulation parameters are shown. For the V2VNDN simulation, we have used ndnsim and the codebase is written in C++. We tested the simulation of V2VNDN by using different forwarding strategy like BestRoute, Flooding, and SmartFlooding. By comparing the different forwarding strategy, we choose BestRoute where the interest will be sent to the highest ranked face. We also varied the communication between nodes ranging from one to one communication and one to three communications. The results were obtained from the 6 simulation runs. Rest of the parameters are same as the table 4.1.

For simulation, we introduced the following nodes with description:

- Node 0: producer node which publishes the contents based on the user interest. Upload all three contents: traffic information, safety messages, and emergency vehicle of some locations. And broadcast all the important information to the nearby vehicles as well as distant vehicles of that area.
- Node 1,2, & 3: consumer nodes which request the necessary information of some area to the producer.

TABLE 4.1

SIMULATION PARAMETERS [3]

Parameters	value
Number of Nodes	4
Number of Interests	1-3
Data Rate	1 Mbps
Maximum packets	100
Forwarding strategy	BestRoute
Producer	0
Consumers	1,2, & 3

4.3 Results and discussion

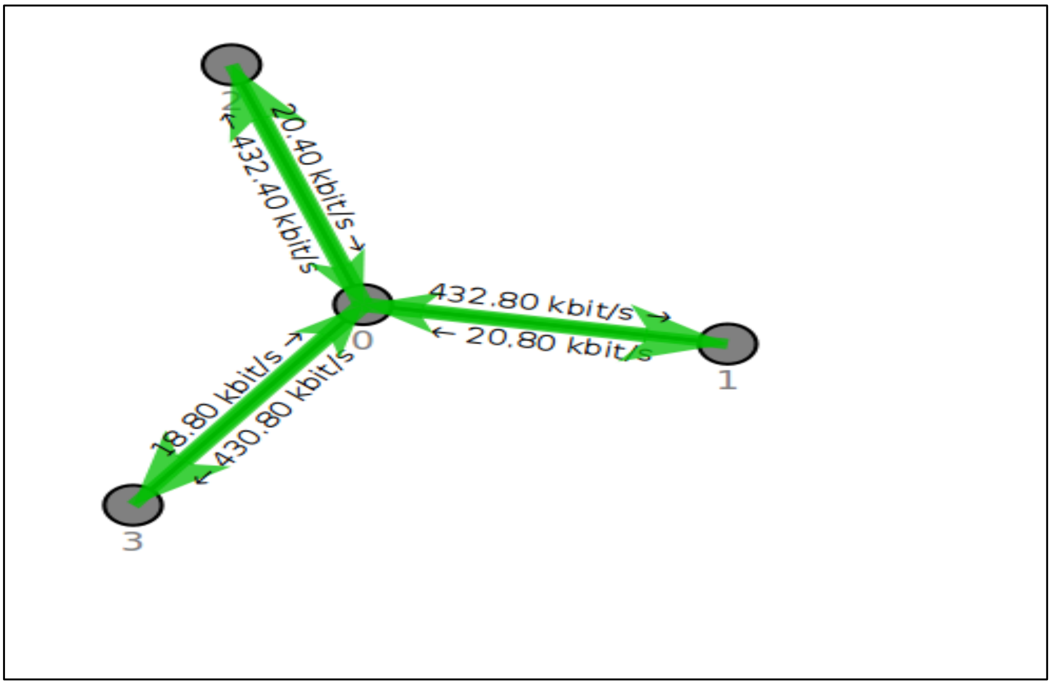


Fig.4.1 Simulation using python visualizer

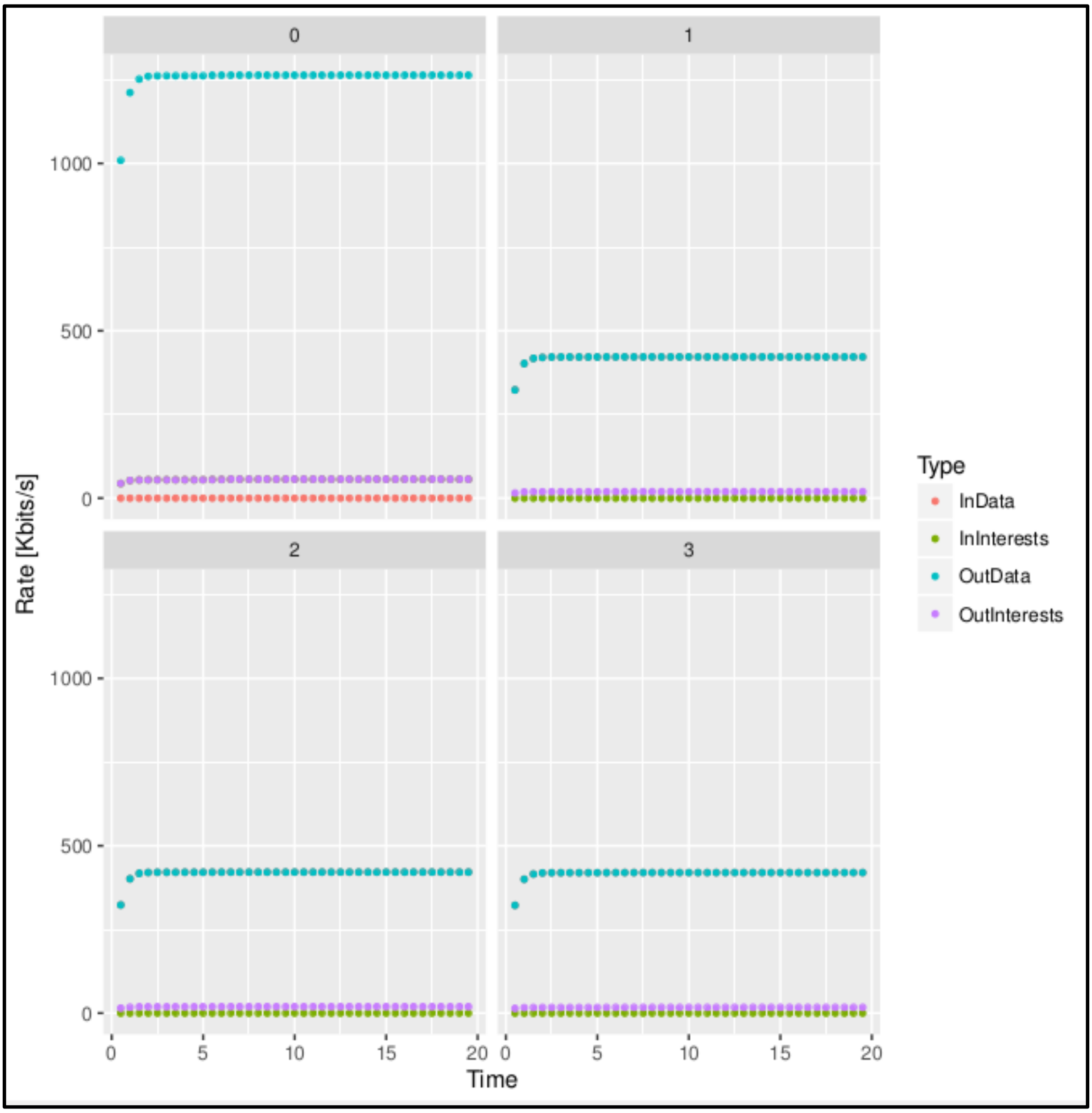


Fig.4.2 Time (sec) vs. data rate (kbps)

Figure 41. shows the simulation of V2VNDN nodes where node 0 is the producer node and the other nodes are consumer nodes. The data rate of interest and content is clearly shown in the graph. Similarly, the time vs. data rate graph is shown in figure 4.2 where we have shown all the possible interests and data from producer to consumer. The above graph is numbered based on the producer and consumer nodes. In the above graph the incoming data (i.e. "InData") in node 0 is null and "InInterest" in node 1, 2, and 3 is null. The interest requested by the consumers and the total interest received by the producer is denoted by "OutInterest". Similarly, the data published and provided by the producer (i.e. node 0) and received by the consumers is denoted by "OutData".

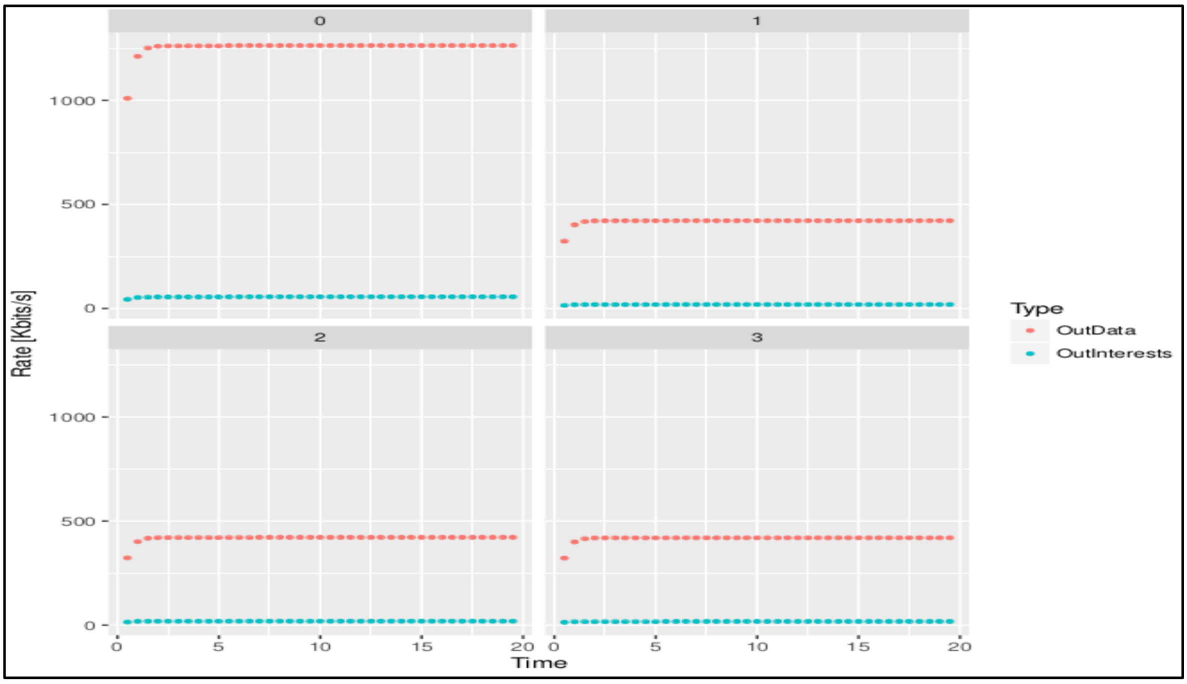


Fig.4.3 Time (sec) vs. data rate (kbps)

Figure 4.3 is the simplest form of figure 4.2, in this graph we have shown only the interest requested by the consumers and the data packets provided by the producer.

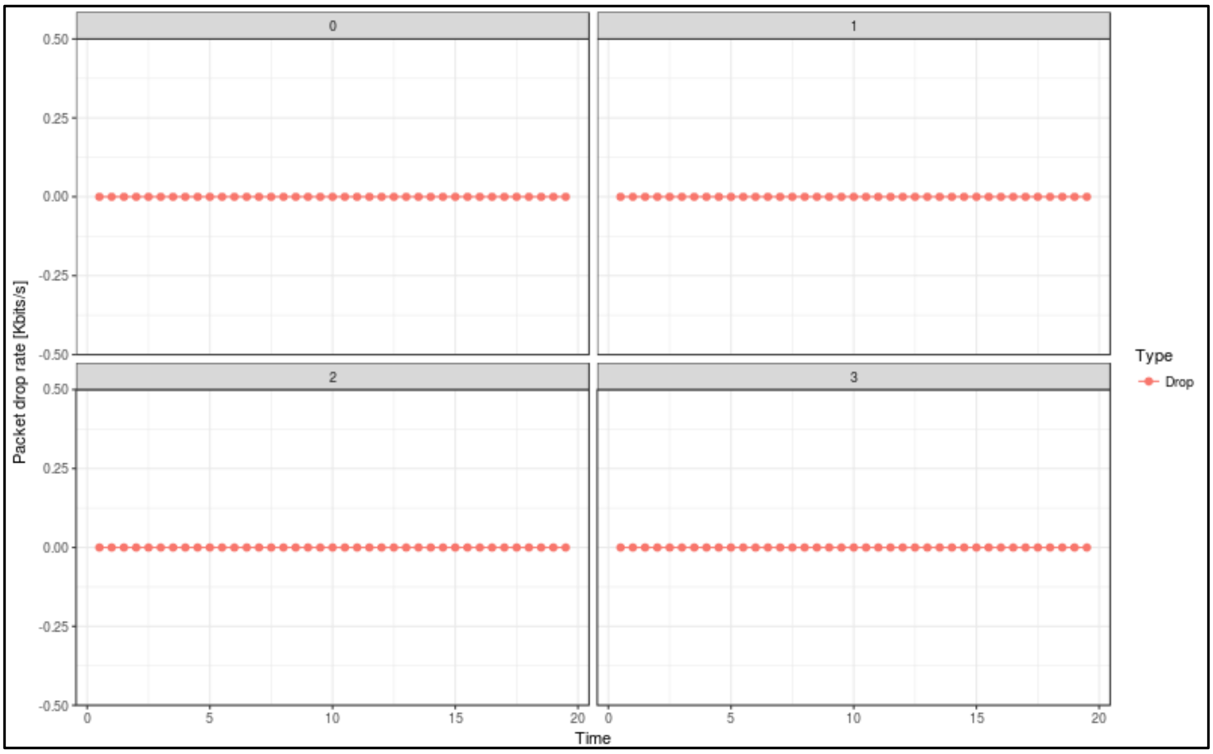


Fig.4.4 Time (sec) vs. packet drop rate (kbps)

Figure 4.4 shows the packet drop rates in the different time interval. Since the packet drop rate in V2VNDN is null. To be specific, V2VNDN can reduces the drop rate (i.e. almost null) in both the case of interest forwarding and data retrieval.

4.3.1 Evaluations of two nodes communication

In this section, we have shown the one to one communication between a producer and a consumer. From these results, we can also analyze the difference between many to one and one to one communication between nodes.

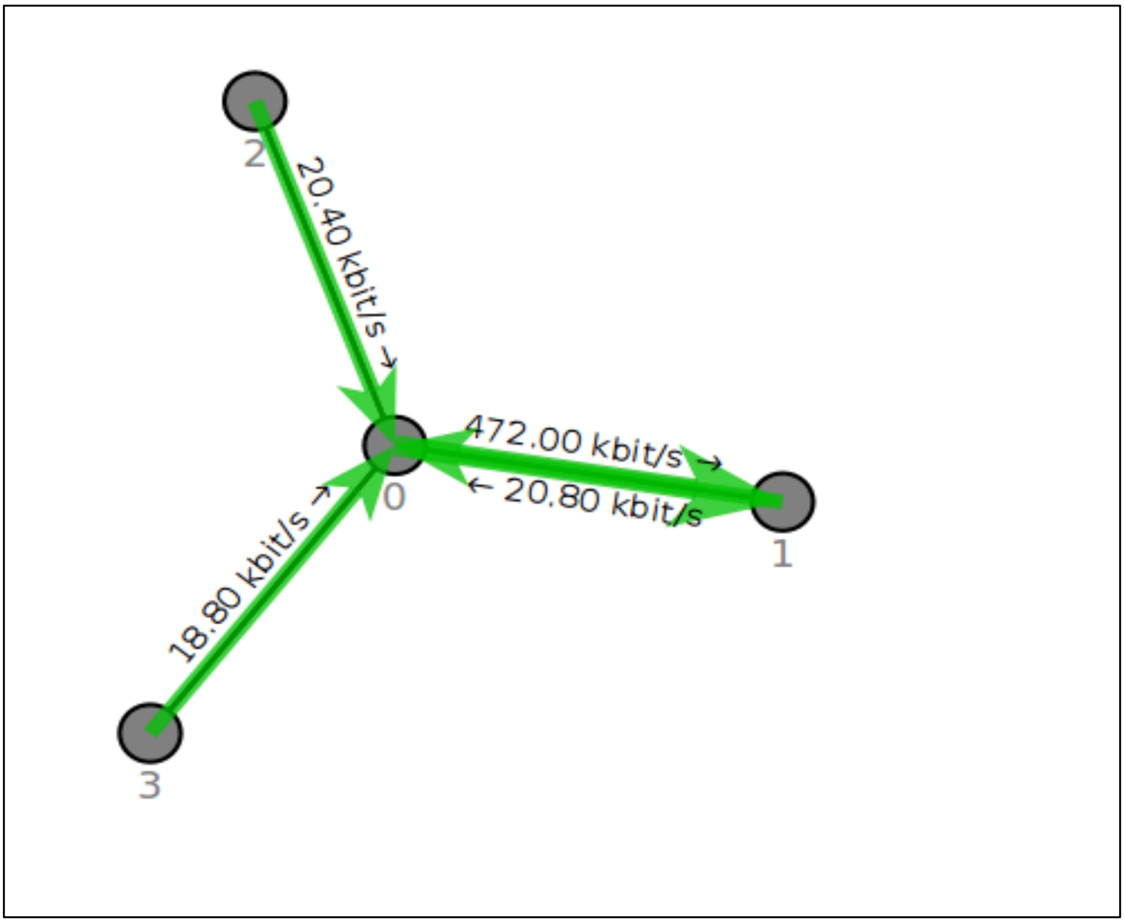


Fig.4.5 Simulation to show the communication between two nodes

Figure 4.5 shows that the communication between node 0 and node 1 but receiving interests from all the nodes. This simulation was done to evaluate the performance between two nodes. This simulation also shows that the interest request by the node 1 is available in node 0 but for other two interests, the contents need to be published within some time interval. So, the node 1 receives the data immediately but the node 2 and 3 will receive after some delay. The average throughput of the data packets and interest packets are 472 kbps and 20 kbps respectively.

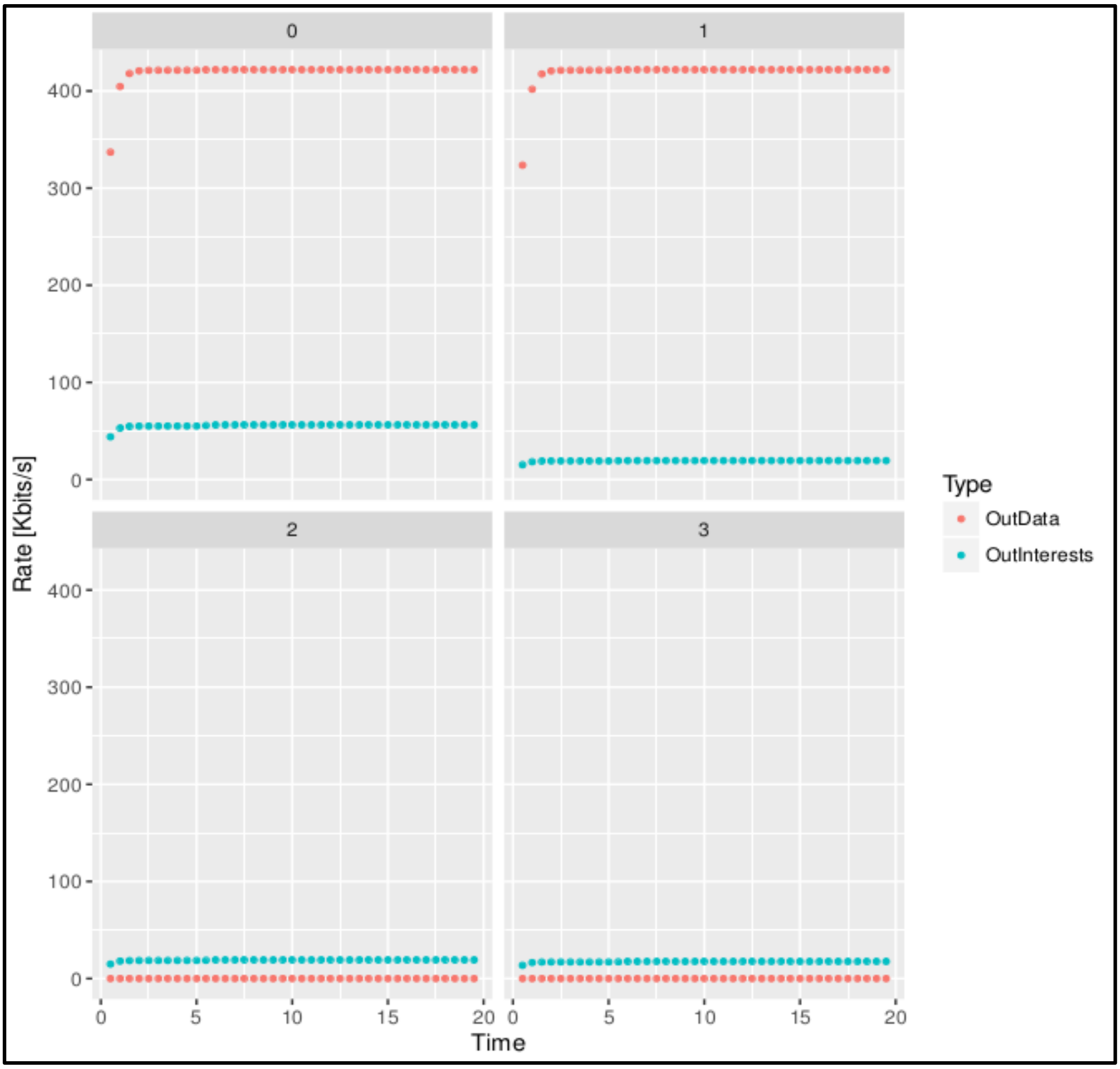


Fig.4.6 Elapsed time (sec) vs. Rate (Kbps)

Figure 4.6 shows the simulation results of V2VNDN when the communication takes place between any two nodes. From figure 4.3 and 4.6, we can analyze that the data packet rate doesn't vary more between one to one and many to one communications between nodes. This proves that many vehicles can request the data packets at the same time to one node (i.e. Vehicle as a producer).

4.3.2 Average data rate and delay of V2VNDN

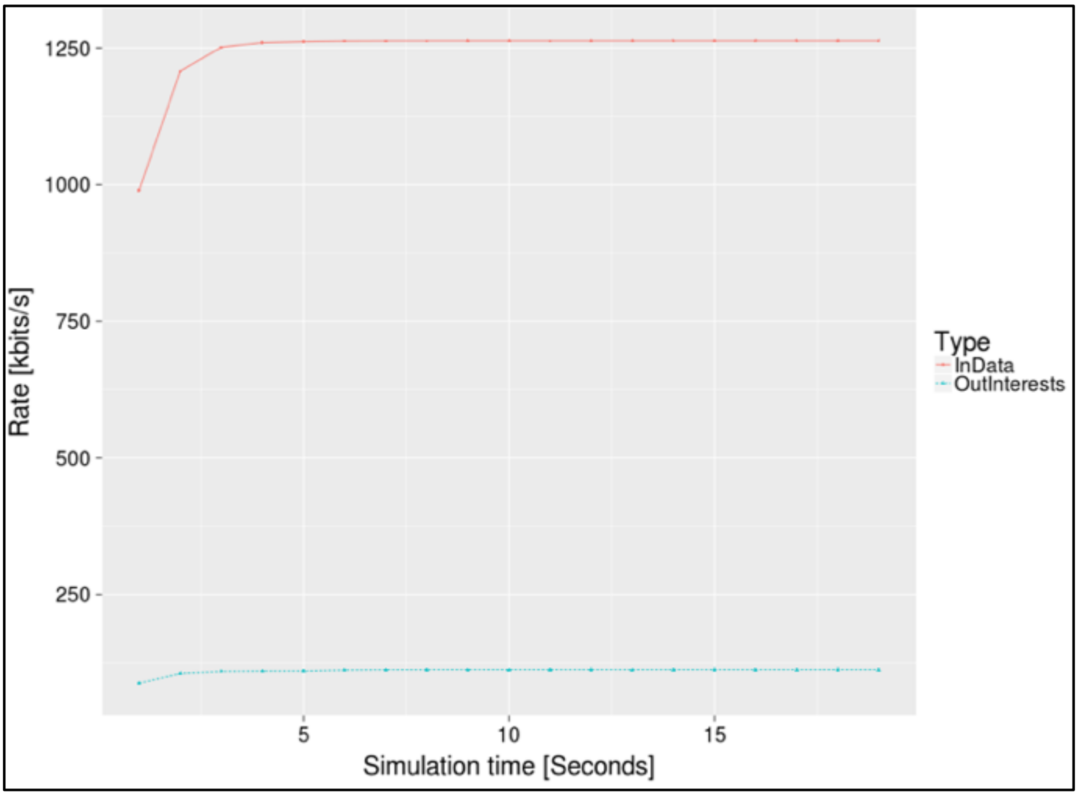


Fig.4.7 Simulation time vs. Data rate [3]

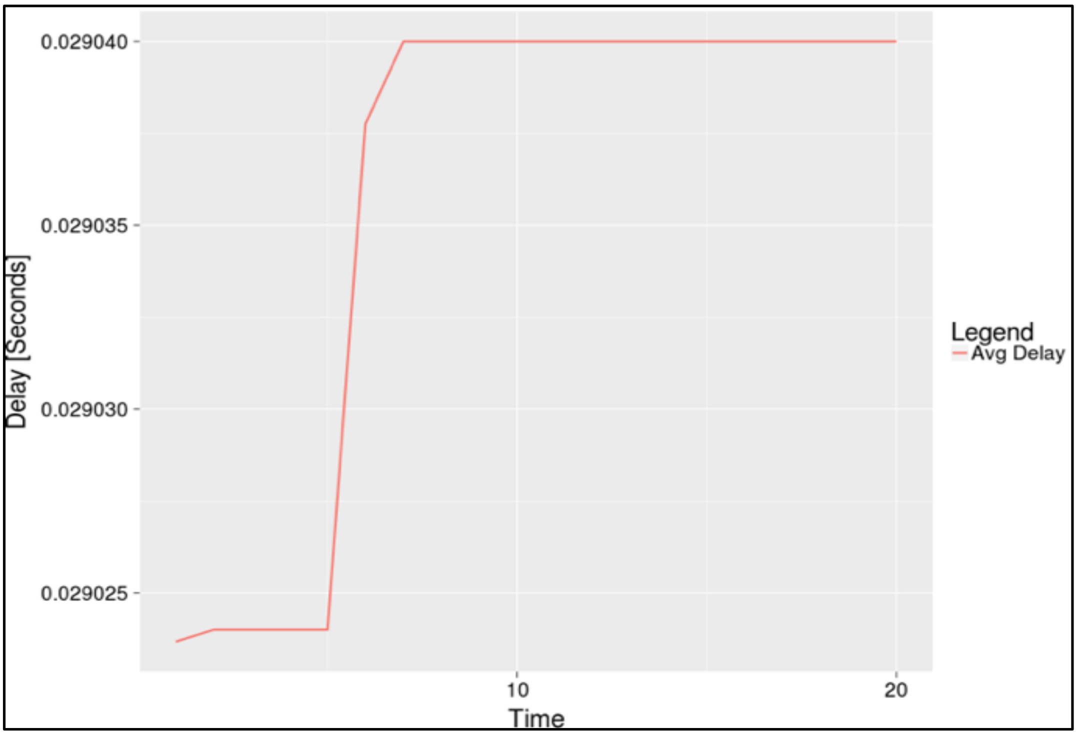


Fig.4.8 Time vs. Delay in seconds [3]

Figure 4.7 and 4.8 shows the average data rate and delay in V2VNDN respectively. Similarly, figure 4.7 shows the average incoming interest and outgoing data in V2VNDN environment. When we analyze the figure 4.8, the initial delay and final delay of V2VNDN is very less. The above simulation results were also verified by using NDNx.

4.4 Performance evaluations of V2VNDN

For performance evaluation of the proposed V2VNDN, we have shown the comparison results of V2VNDN, regular NDN and TCP/IP. For TCP/IP, we have considered the IEEE 802.11p [20] [21]. The simulation shows the delay and average data rate (kbps) off all three scenarios.

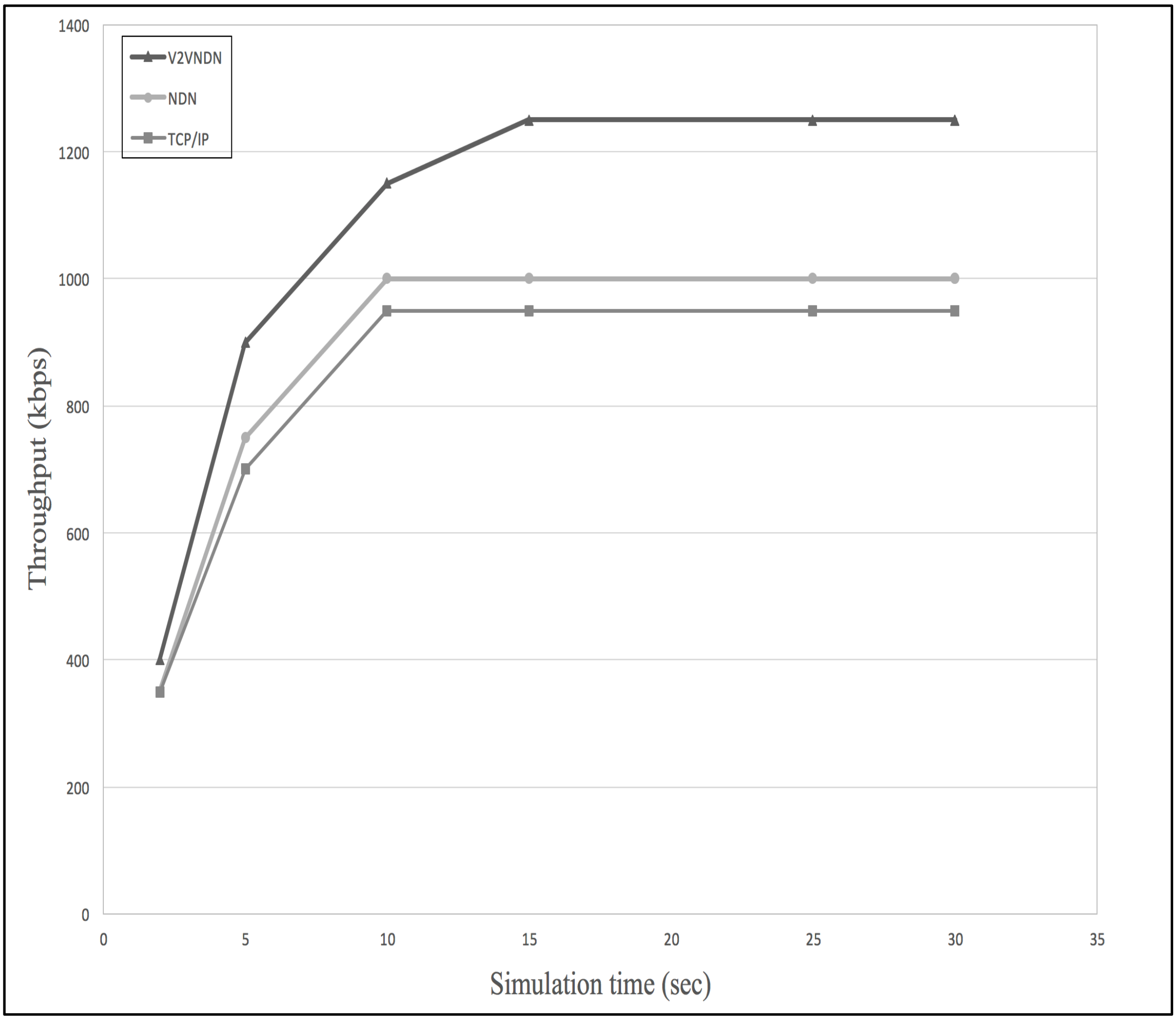


Fig.4.9 Simulation time vs. Throughput (Kbps)

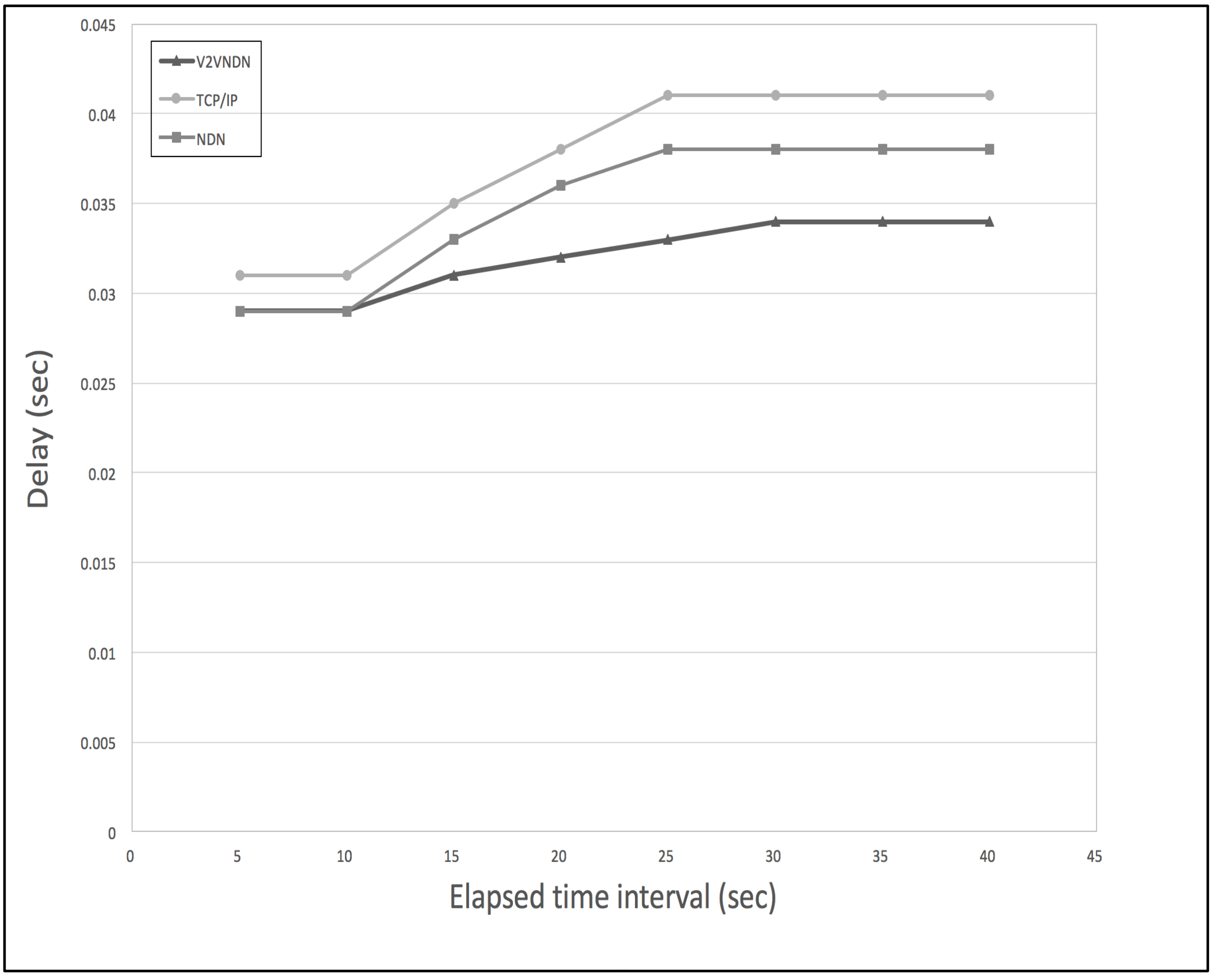


Fig.4.10 Elapsed time interval (sec) vs. Delay (sec)

Figure 4.9 and 4.10 show the average throughput and delay graph between V2VNDN, IEEE802.11p and regular NDN. In both the simulation, the V2VNDN shows the better performance than the other networking scenario. Since the average speed during the obstacles or emergency conditions is very low. So, the mobility and handover for vehicular communication will be almost same as the normal communication between nodes using NDN environment. From this scenario, we can conclude that there will be no mobility and handover problem in V2VNDN. So, it is worth to mention that, the V2VNDN can become one of the best proposals for vehicular communication and is easy to conclude that NDN is the best networking scenario for vehicular communication.

Chapter 5

Experimental Results

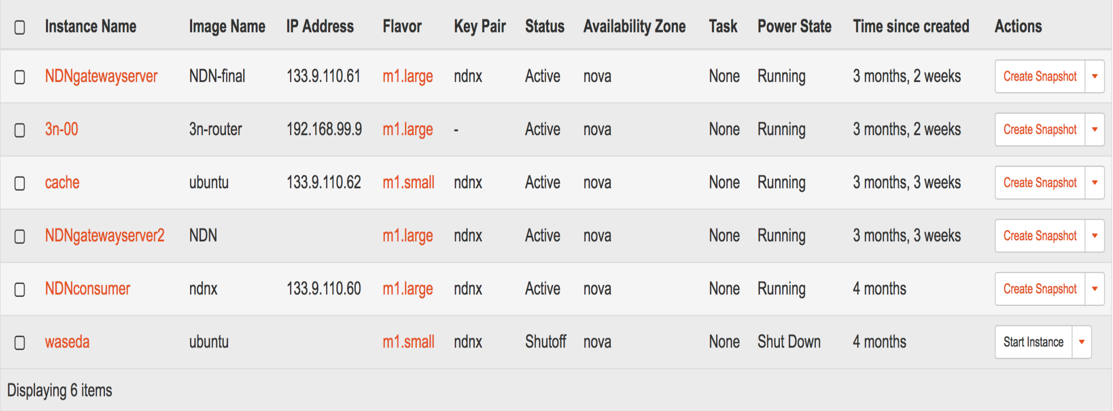
5.1 Introduction

In this chapter, we discuss about the experimental results of a proposed system using NDNx with some modifications. Since we built and configured NDNx for vehicle to vehicle communication with some modifications, we have named it as V2VNDN. We also used openstack [22] and Ubuntu 14.04 LTS operating system for the configuration of the total system. At first, we configured and built two Linux servers; one as a producer and the other as a consumer of V2VNDN and uploaded both the images in the openstack system in VDI (Virtual Disk Image) format and finally, assigned global IP address to each. This experimental result shows the communication between two nodes i.e. two vehicles.

5.2 Openstack instances

Openstack [22] is the open source software for cloud computing that controls the storage, computing, and networking resources through a datacenter. In our case, we used openstack to manage and control V2VNDN producer and consumer. It also helps to control the instances remotely and makes the total process to control the V2VNDN system easier.

For the openstack management: We created two nodes; controller node and compute node. We then successfully installed openstack components and service to these two nodes using Ocata. For V2VNDN transmission, we have successfully transmitted the real video file (about 130 MB), text files, and image files between two machines (one as the producer and one as the consumer).



□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	NDNgatewayserver	NDN-final	133.9.110.61	m1.large	ndnx	Active	nova	None	Running	3 months, 2 weeks	Create Snapshot ▾
□	3n-00	3n-router	192.168.99.9	m1.large	-	Active	nova	None	Running	3 months, 2 weeks	Create Snapshot ▾
□	cache	ubuntu	133.9.110.62	m1.small	ndnx	Active	nova	None	Running	3 months, 3 weeks	Create Snapshot ▾
□	NDNgatewayserver2	NDN		m1.large	ndnx	Active	nova	None	Running	3 months, 3 weeks	Create Snapshot ▾
□	NDNconsumer	ndnx	133.9.110.60	m1.large	ndnx	Active	nova	None	Running	4 months	Create Snapshot ▾
□	waseda	ubuntu		m1.small	ndnx	Shutoff	nova	None	Shut Down	4 months	Start Instance ▾

Displaying 6 items

Fig.5.1. Openstack instances of V2VNDN

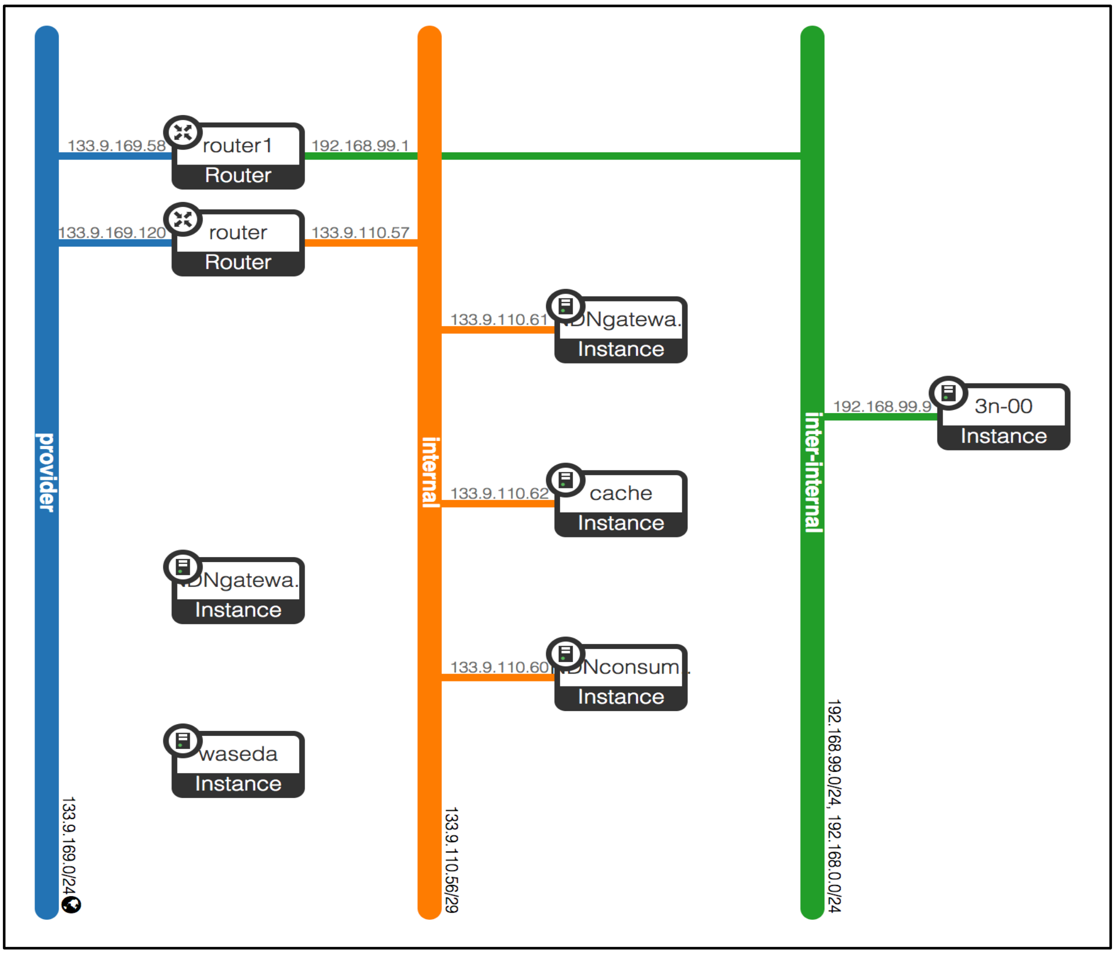
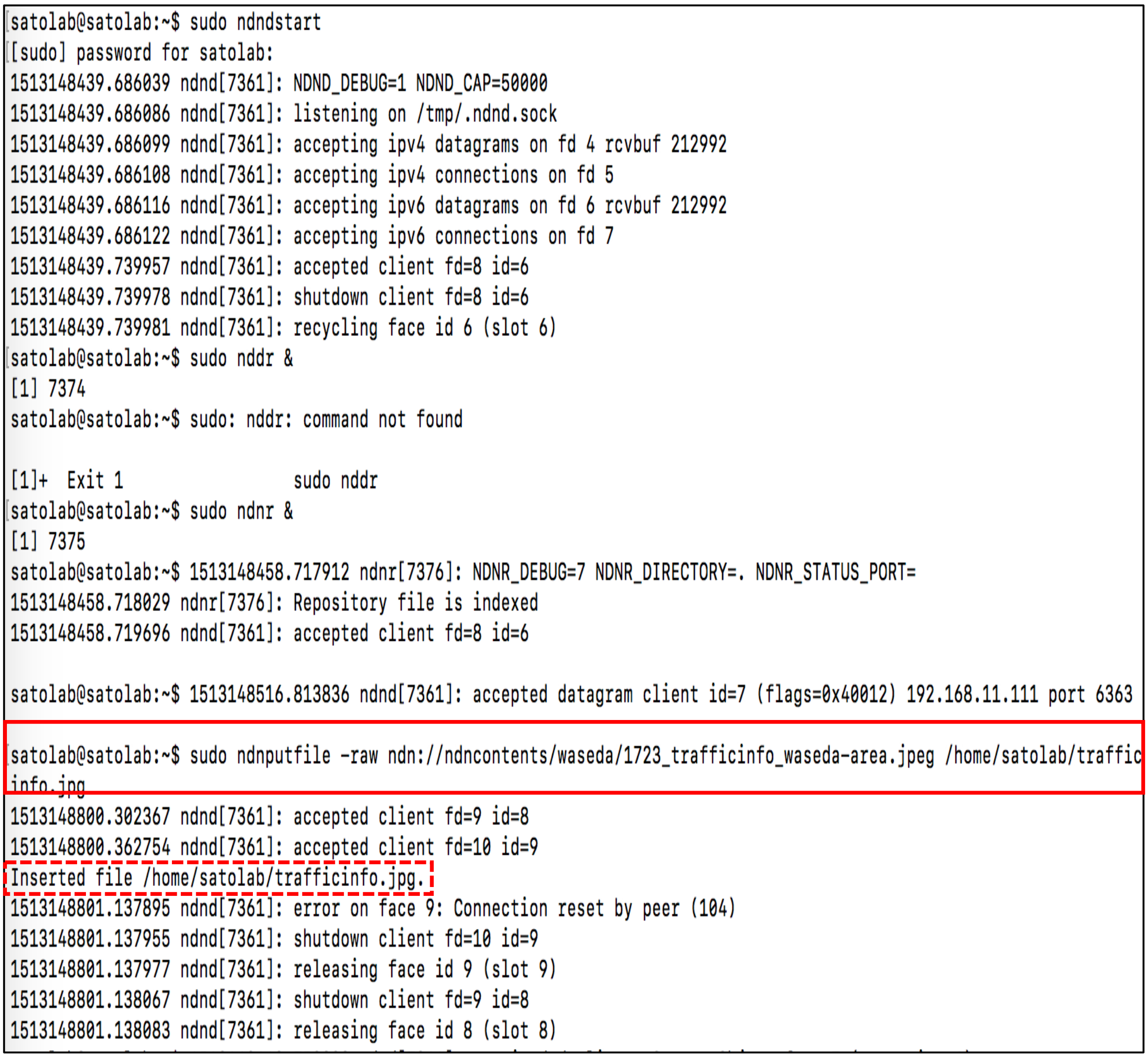


Fig.5.2 Openstack configuration for V2VNDN nodes.

Figure 5.1 shows the openstack instances of V2VNDN where we use NDNgatewayserver and NDNconsumer for communication between two nodes. In this figure we can see, both the images are in active state and has a large flavor size. Similarly, figure 5.2 shows the openstack configuration for V2VNDN nodes.

5.3 V2VNDN producer

In this section, we discuss about the experimental result of a producer side in V2VNDN. In the producer side, first we start NDN and then the repository where the published files are uploaded and broadcasted in the NDN network. After completing the initial process, we accept the client request (i.e. consumer request) by initializing the face ID. Then we publish and send the requested file to the user. In V2VNDN, we have used the area name and vehicles number as a routable prefix. So, it's easier for consumer to send an interest to the nearby producer.



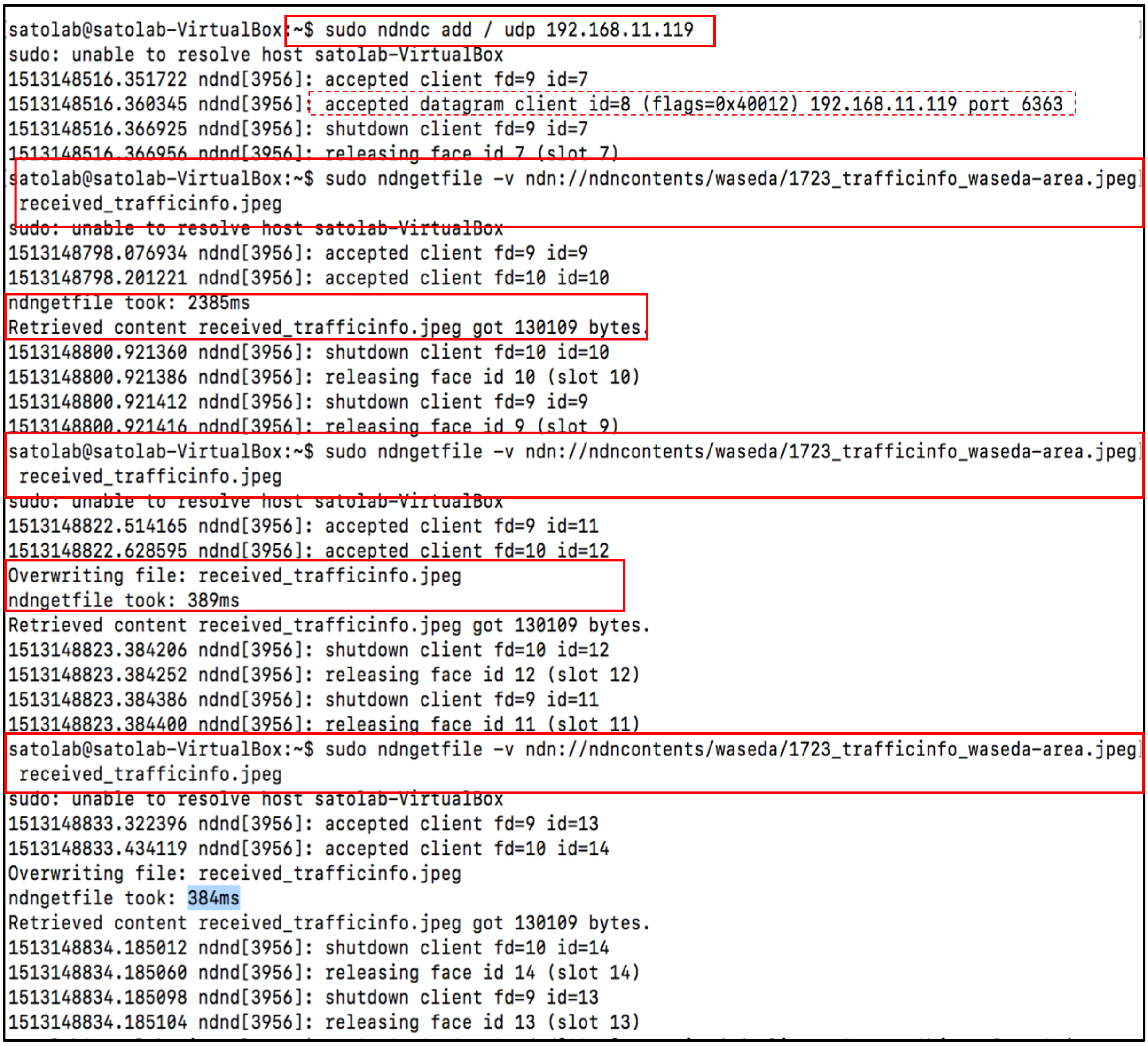
```
satolab@satolab:~$ sudo ndndstart
[sudo] password for satolab:
1513148439.686039 ndnd[7361]: NDND_DEBUG=1 NDND_CAP=50000
1513148439.686086 ndnd[7361]: listening on /tmp/.ndnd.sock
1513148439.686099 ndnd[7361]: accepting ipv4 datagrams on fd 4 rcvbuf 212992
1513148439.686108 ndnd[7361]: accepting ipv4 connections on fd 5
1513148439.686116 ndnd[7361]: accepting ipv6 datagrams on fd 6 rcvbuf 212992
1513148439.686122 ndnd[7361]: accepting ipv6 connections on fd 7
1513148439.739957 ndnd[7361]: accepted client fd=8 id=6
1513148439.739978 ndnd[7361]: shutdown client fd=8 id=6
1513148439.739981 ndnd[7361]: recycling face id 6 (slot 6)
satolab@satolab:~$ sudo nddr &
[1] 7374
satolab@satolab:~$ sudo: nddr: command not found

[1]+  Exit 1                  sudo nddr
satolab@satolab:~$ sudo ndnr &
[1] 7375
satolab@satolab:~$ 1513148458.717912 ndnr[7376]: NDNR_DEBUG=7 NDNR_DIRECTORY=. NDNR_STATUS_PORT=
1513148458.718029 ndnr[7376]: Repository file is indexed
1513148458.719696 ndnd[7361]: accepted client fd=8 id=6

satolab@satolab:~$ 1513148516.813836 ndnd[7361]: accepted datagram client id=7 (flags=0x40012) 192.168.11.111 port 6363

satolab@satolab:~$ sudo ndnputfile -raw ndn://ndncontents/waseda/1723_trafficinfo_waseda-area.jpeg /home/satolab/traffic
info.jpg
1513148800.302367 ndnd[7361]: accepted client fd=9 id=8
1513148800.362754 ndnd[7361]: accepted client fd=10 id=9
Inserted file /home/satolab/trafficinfo.jpg.
1513148801.137895 ndnd[7361]: error on face 9: Connection reset by peer (104)
1513148801.137955 ndnd[7361]: shutdown client fd=10 id=9
1513148801.137977 ndnd[7361]: releasing face id 9 (slot 9)
1513148801.138067 ndnd[7361]: shutdown client fd=9 id=8
1513148801.138083 ndnd[7361]: releasing face id 8 (slot 8)
```

Fig.5.3 Producer side of V2VNDN

Figure 5.3 shows the producer of V2VNDN which includes all the necessary commands to start, upload, and published the file in NDN repository. The highlighted part shows the uploaded and published "trafficinfo.jpeg" file of "Waseda area" into NDN environment based on the user request. We have used the "raw" format to publish the file after getting the request from consumer side. In the above highlighted sections, first part shows the necessary command to upload and publish the file of traffic information and the second part shows the inserted message in the NDN environment.

5.4 V2VNDN consumer

In this section, we discuss about the experimental result of a consumer side in V2VNDN. In the consumer side, first we start NDN and the repository. After completing the initial process, we establish the link between producer and consumer through link layer. Since a producer and a consumer are the openstack instances assigned with the global IP, a consumer can add the global IP address of a producer to establish the link. Finally, a consumer can request the data by sending the necessary interest to the producer.



```
satolab@satolab-VirtualBox:~$ sudo ndndc add / udp 192.168.11.119
sudo: unable to resolve host satolab-VirtualBox
1513148516.351722 ndnd[3956]: accepted client fd=9 id=7
1513148516.360345 ndnd[3956]: accepted datagram client id=8 (flags=0x40012) 192.168.11.119 port 6363
1513148516.366925 ndnd[3956]: shutdown client fd=9 id=7
1513148516.366956 ndnd[3956]: releasing face id 7 (slot 7)
satolab@satolab-VirtualBox:~$ sudo ndngetfile -v ndn://ndncontents/waseda/1723_trafficinfo_waseda-area.jpeg
 received_trafficinfo.jpeg
sudo: unable to resolve host satolab-VirtualBox
1513148798.076934 ndnd[3956]: accepted client fd=9 id=9
1513148798.201221 ndnd[3956]: accepted client fd=10 id=10
ndngetfile took: 2385ms
Retrieved content received_trafficinfo.jpeg got 130109 bytes.
1513148800.921360 ndnd[3956]: shutdown client fd=10 id=10
1513148800.921386 ndnd[3956]: releasing face id 10 (slot 10)
1513148800.921412 ndnd[3956]: shutdown client fd=9 id=9
1513148800.921416 ndnd[3956]: releasing face id 9 (slot 9)
satolab@satolab-VirtualBox:~$ sudo ndngetfile -v ndn://ndncontents/waseda/1723_trafficinfo_waseda-area.jpeg
 received_trafficinfo.jpeg
sudo: unable to resolve host satolab-VirtualBox
1513148822.514165 ndnd[3956]: accepted client fd=9 id=11
1513148822.628595 ndnd[3956]: accepted client fd=10 id=12
Overwriting file: received_trafficinfo.jpeg
ndngetfile took: 389ms
Retrieved content received_trafficinfo.jpeg got 130109 bytes.
1513148823.384206 ndnd[3956]: shutdown client fd=10 id=12
1513148823.384252 ndnd[3956]: releasing face id 12 (slot 12)
1513148823.384386 ndnd[3956]: shutdown client fd=9 id=11
1513148823.384400 ndnd[3956]: releasing face id 11 (slot 11)
satolab@satolab-VirtualBox:~$ sudo ndngetfile -v ndn://ndncontents/waseda/1723_trafficinfo_waseda-area.jpeg
 received_trafficinfo.jpeg
sudo: unable to resolve host satolab-VirtualBox
1513148833.322396 ndnd[3956]: accepted client fd=9 id=13
1513148833.434119 ndnd[3956]: accepted client fd=10 id=14
Overwriting file: received_trafficinfo.jpeg
ndngetfile took: 384ms
Retrieved content received_trafficinfo.jpeg got 130109 bytes.
1513148834.185012 ndnd[3956]: shutdown client fd=10 id=14
1513148834.185060 ndnd[3956]: releasing face id 14 (slot 14)
1513148834.185098 ndnd[3956]: shutdown client fd=9 id=13
1513148834.185104 ndnd[3956]: releasing face id 13 (slot 13)
```

Fig.5.4 Consumer side of V2VNDN

Figure 5.4 shows the consumer of V2VNDN which shows all the necessary commands to communicate between a producer and a consumer. After starting the link, a consumer gets the accepted acknowledgment form a producer. Then, a consumer requests the necessary contents to the producer with the suitable content ID. If the ID is matched in the content table broadcasted by a producer, a producer will upload and publish the requested contents and sent to the consumer. In the above figure, highlighted parts show the step by step procedure on a consumer side to request and get the contents from a producer side. We have also shown the repeated request (i.e. repetition of the same request) and time take by each to receive and download the contents. The first request to receive the traffic information took 2385ms but the second and third request took 389ms and 384ms respectively. The above results also verify that the time taken to receive the same contents is very less after a first request. This also proves the NDN as an in-network caching network scenario.

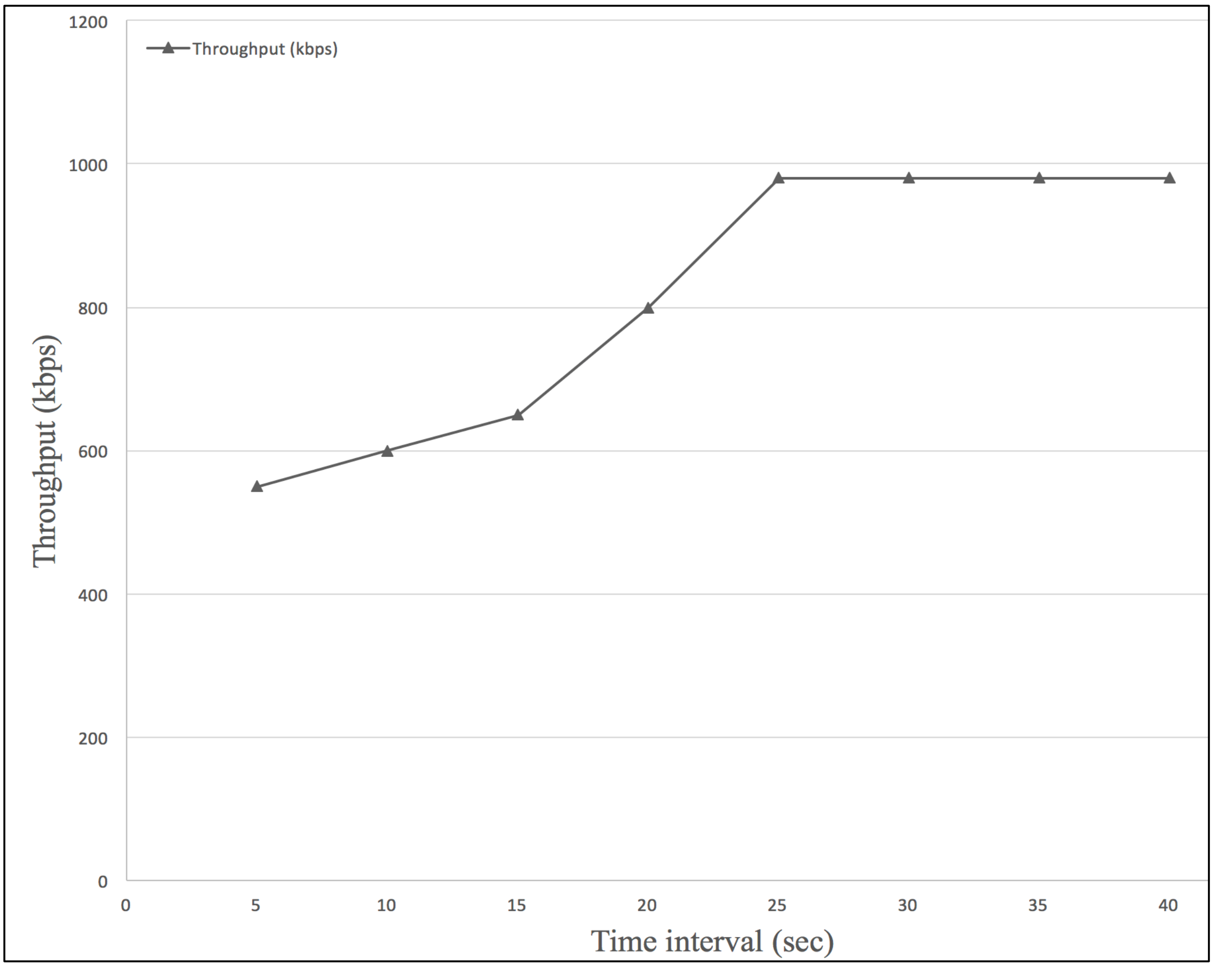


Fig. 5.5 Throughput graph of V2VNDN

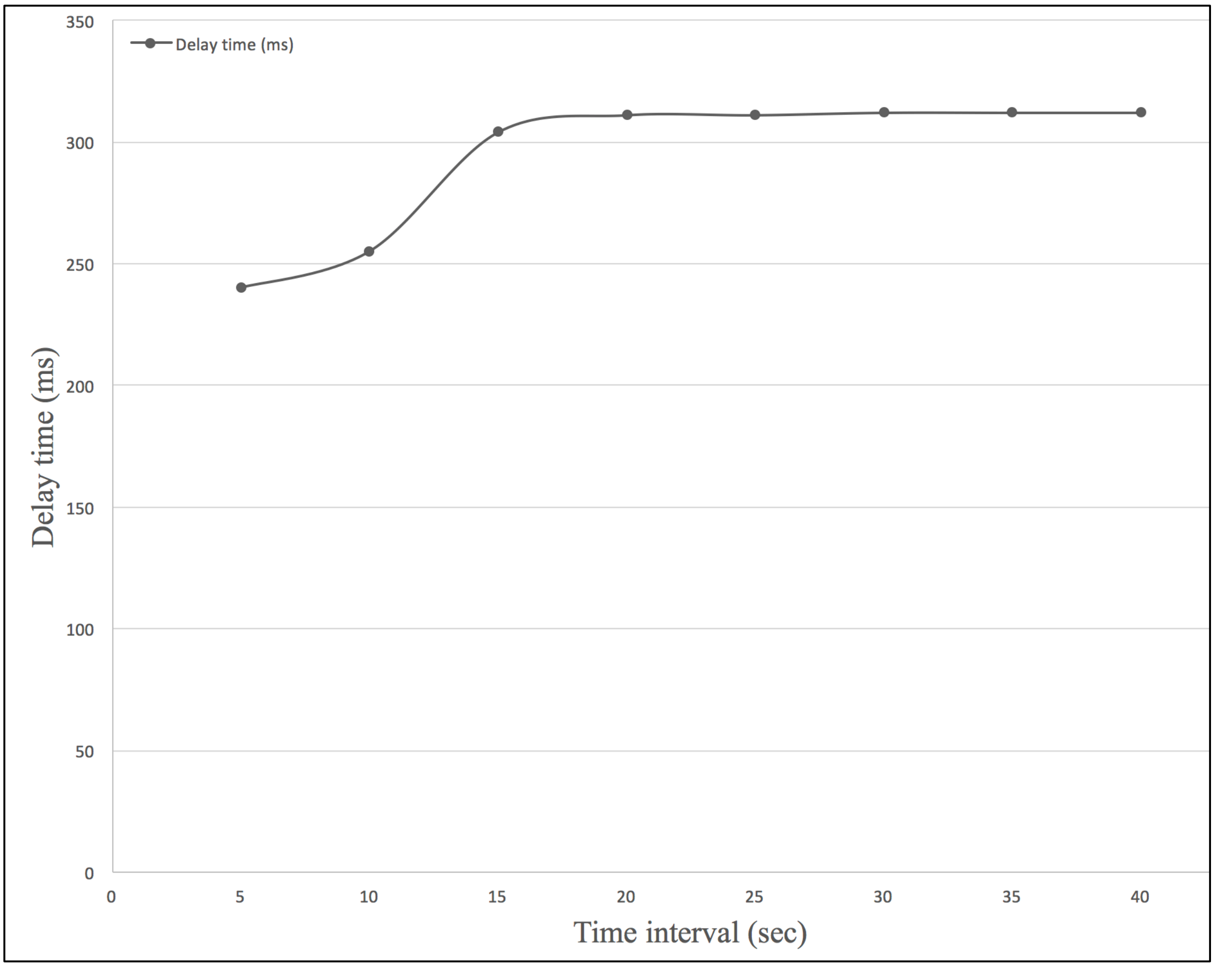


Fig. 5.6 Delay time graph of V2VNDN

Figure 5.5 and 5.6 show the experimental results of throughput and delay time of V2VNDN respectively. We can conclude from both the simulation and experimental results of V2VNDN that, the throughput is higher and delay is very less than the other networking scenarios.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this paper, we proposed the vehicular communication using named node networking (V2VNDN) and shown the naming, publishing and retrieving strategy of NDN environment. Through the simulation, we found that the delay of V2VNDN is less and throughput is higher than the regular NDN and the TCP/IP. Since the number of contents is fixed and each contents ID is assigned with vehicle number and area name, it's very easy to know the content ID of nearby vehicles and received the required information. The main goal we achieve via V2VNDN is to share the important information between two vehicles with high throughput and less delay. It is also worth to say from the data-drop graph that the data drop rate is null. We can conclude by stating that the V2VNDN is easy, accurate, highly efficient, cost-effective, and strong security networking system which can be implemented soon to make the transportation system intelligent and accident-free.

When we analyze the experimental results, the time take to publish the file in the repository is very less. Also, the consumer side clearly shows the V2VNDN as an in-network caching system. It also describes the evaluations of performance and naming strategy.

The conclusion of each chapter is listed below:

Chapter 1:

This chapter introduces the background of vehicular communication and Named Data Networking. It also answers about "why the vehicular communication is important" and how NDN is the suitable networking scenario for V2V. We have also discussed in brief about the issues in existing network. At last, we have introduced our main proposal (i.e. V2VNDN) and shown the architecture of V2VNDN.

Chapter 2:

In this chapter, we have described the introduction of vehicle to vehicle communication and Named Data Networking. This chapter mainly focuses on the related works with V2VNDN and issues in the past research papers as well as the discussion about similar research papers and how our proposal is different from other.

Chapter 3:
In this chapter, we have described about the vehicular communication in NDN environment. It focuses mainly on naming process, contents ID, and the procedure to publish the contents in NDN. It also describes the step by step procedures to publish and retrieved the files in our proposed system.

Chapter 4:
This chapter shows the simulation results and discussion on each simulation results. We have carried out the number of simulation to show the functionality and performance of the propose system. The simulation results are divided into two types; one is pure simulation for V2VNDN and the other is comparison of V2VNDN with the TCP/IP and regular NDN. The simulation results mainly focus on throughput, delay and drop rate of the propose system. In all the simulation, the performance of V2VNDN is better than the other networking scenario.

Chapter 5:
This chapter shows the experimental results and discussion on the producer and the consumer side. It shows the necessary steps to upload and publish the contents in the repository and time to insert the file in the producer side. Similarly, it also describes the necessary steps in the consumer side and shows the time taken to retrieve the data and size of the data.

Chapter 6:
This chapter describes the conclusion of the propose system and the future work that are necessary to improve and implement the vehicular communication system.

6.2 Future work

For future work, we will improve the existing difficulties in Named Data Networking. Since the NDN is a new internet architecture and holds many challenge to implement in the real environment. So, we will overcome the challenges and difficulties in the future. Also, we will consider the energy consumption of the propose system. We will also try to implement the system in the real environment and note down the performance evaluations.

In V2VNDN, we discussed only about the fixed contents and low speed vehicular environment. In the future, we will evaluate the mobility and handover for high speed vehicles and we'll also increase the number of contents by managing the routing table and routable prefix.

Research Achievements

[1] Automated Cruise Model car based on the Information Centric Networking Mahendra SAPKOTA, Masaru SAWADA, and Takuro SATO; Proceeding of the 2017 IEICE General Conference; BS-1-1.

[2] Vehicular communication using Named Data Networking (V2VNDN) Mahendra Sapkota, Rio Komaji, and Sato Takuro; Technical Committee of ICN; http://www.ieice.org/~icn/ .

References

[1] "Vehicular communication systems" https://en.wikipedia.org/wiki/Vehicular_communication_systems

[2] "Named Data Networking (NDN) Project" http://named-data.net .

[3] Vehicular communication using Named Data Networking (V2VNDN) Mahendra Sapkota, Rio Komaji, and Sato Takuro; Technical Committee of ICN; http://www.ieice.org/~icn/ .

[4] "Named Data Networking (NDN) Project" https://named-data.net/codebase/platform/moving-to-ndnx .

[5] Automated Cruise Model car based on the Information Centric Networking Mahendra SAPKOTA, Masaru SAWADA, and Takuro SATO; Proceeding of the 2017 IEICE General Conference; BS-1-1.

[6] "V2V: what are vehicle-to-vehicle communications and how do they work?" https://www.extremetech.com/extreme/176093-v2v-what-are-vehicle-to-vehicle-communications-and-how-does-it-work

[7] By Van Jacobson, Diana K. Smetters, James D. Thornton, Michael Plass, Nick Briggs, Rebecca Braynard Communications of the ACM, Vol. 55 No. 1, Pages 117-12410.1145/2063176.2063204

[8] C. H. Lee, K. G. Lim, B. L. Chua, R. K. Y. Chin and K. T. K. Teo, "Performance evaluation of IEEE 802.11 for vehicular communication," *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, Seoul, 2016, pp. 1-4. doi: 10.1109/ICCE-Asia.2016.7804783

[9] L. Kristiana, C. Schmitt and B. Stiller, "V2VUNet — A filtering out concept for packet forwarding decision in three-dimensional inter-vehicular communication scenarios," *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, 2016, pp. 1-6. doi: 10.1109/PIMRC.2016.7794600

[10] S.H. Ahmed, S.H. Bouk, M.A. Yaqub, D.KIM, H. Song and J. Lloret, "CODIE: Controlled Named Data Networks," in IEEE Transaction on Vehicular Technology, vol. 65, no. 6, pp. 3954-3963, June 2016. doi: 10.1109/TVT.2016.2558650.

[11] N. Kumar and J. H. Lee, "Peer-to-Peer Cooperative Caching for Data Dissemination in Urban Vehicular Communications," in *IEEE Systems Journal*, vol. 8, no. 4, pp. 1136-1144,

Dec. 2014. doi: 10.1109/JSYST.2013.2285611

[12] T. Karimireddy and A. G. A. Bakshi, "A hybrid security framework for the vehicular communications in VANET," *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2016, pp. 1929-1934. doi: 10.1109/WiSPNET.2016.7566479

[13] Y. H. Ho, Chun-Han Lin, Shih-Jie Sun and Ling-Jyh Chen, "Emergency application for Vehicle-to-Vehicle Communication using Named Data Networking (eVNDN)," *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, 2016, pp. 611-616. doi: 10.1109/ICUFN.2016.7537105

[14] E. Kalogeiton, Z. Zhao and T. Braun, "Is SDN the solution for NDN-VANETs?" *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Budva, 2017, pp. 1-6. doi: 10.1109/MedHocNet.2017.8001643

[15] S. H. Ahmed, S. H. Bouk, D. Kim, D. B. Rawat and H. Song, "Named Data Networking for Software Defined Vehicular Networks," in *IEEE Communications Magazine*, vol. 55, no. 8, pp. 60-66, 2017. doi: 10.1109/MCOM.2017.1601137

[16] V. Jain and R. S. Kushwah, "NDN architectures in VANET scenario," *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, 2017, pp. 178-182. doi: 10.1109/ICICCT.2017.7975183

[17] "NS-3 based Named Data Networking Simulator" http://ndnsim.net/1.0/index.html# .

[18] "Network simulator (ns-3)" https://www.nsnam.org/ .

[19] "Moving to NDNx" https://named-data.net/codebase/platform/moving-to-ndnx/ .

[20] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," VTC Spring 2008 - IEEE Vehicular Technology Conference, Singapore, 2008, pp. 2036-2040. doi: 10.1109/VETECS.2008.458

[21] "IEEE 802.11p Wikipedia" https://en.wikipedia.org/wiki/IEEE_802.11p .

"Openstack environment" https://www.openstack.org/ .