Tampere University

MD NAZMUS SAQIB

PERFORMANCE EVALUATION OF NESTED INTERESTS IN NDN SYSTEMS

Faculty of Information Technology and Communication Sciences
M. Sc. Thesis
October 2021

ABSTRACT

Md Nazmus Saqib: Performance Evaluation of Nested Interests in NDN Systems
M.Sc. Thesis
Tampere University
Master’s Degree Programme in Information Technology
October 2021

Key words and terms: NDN, future internet, edge computing, ICN

The evaluation of Internet-of-Things (IoT) makes way for the enhancement of Edge computing for faster communication in the wireless mobile network to enable the use of shared external resources and performing computational tasks. To perform this type of tasks latency can be a major bottleneck and the conventional host-based network cannot provide the best solution for that. Named Data Networking (NDN) possess the possibility to improve this kind of scenario where the dynamic users can share the external resources in the edge servers to perform their time-sensitive tasks. To demonstrate the proof of this idea, here it is considered the two basic features of NDN, Interest Aggregation and Caching in a mobile wireless network scenario. The outcome of this research supports the claim demonstrating the advantages of NDN over the traditional network architecture.

PREFACE

With the dream of having a Master's degree, I came to Finland. I have always been mesmerized by the Finnish education system. I am very fortunate to be a part of it. I am also very much grateful for having the opportunity to be a part of Finnish society.

I want to express my sincere gratitude to all the faculty members, staff and students of Tampere University. Especially I want to thank my supervisors Dr. Dmitri Moltchanov and Prof. Yevgeni Koucheryavy for their constant support. Without the support, my journey would never be such a smooth one.

Also, I am immensely grateful to my family members for their unconditional support and inspiration, which led me to attain my goal.

Tampere, October 27, 2021

Md Nazmus Saqib

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS & ABBREVIATIONS

NDN	Named Data Networking
NFD	NDN Forwarding Daemon
FIB	Forwarding Information Base
PIT	Pending Interest Table
CS	Content Store
ICN	Information-Centric Networking
CCN	Content-Centric Networking
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
ACK	Acknowledgement
IoT	Internet of Things
ACT	Audio Conference Tool
WLAN	Wireless Local Area Network
VNDN	Vehicular Named Data Networking
SDN	Software Defined Network
VANET	Vehicular Ad-hoc Network
NFN	Named Function Networking
CSMA	Carrier-Sense Multiple Access
DDOS	Distributed Denial-Of-Service
UCLA	University of California, Los Angeles
RFCOMM	Radio Frequency Communication
PAN	Personal Area Network

1 Introduction

The evaluation of today’s network communication system and internet architecture has been possible because of the ubiquitous proliferation of TCP/IP. It is the reflection of years of researches and improvement of TCP/IP architecture. It has been the first consideration in establishing a communication framework for a long time. The architecture has great potentiality and capability in supporting almost all modern services and applications. Despite having this much potentiality, it is still not perfect and can not be a solution to all the problems in the network system. It possesses a lot of limitations in such a situation where all the services are distributed in the network and the environment consists of dynamic users.

The unwavering quality and congestion control components of TCP don't manage high latency and loss of transmitted packets on significant distance networks, consequently making the transmission of packets unrealistically sluggish. Except if the distributed network server is exceptionally close from a network viewpoint, a standard TCP stream doesn't give adequate throughput. A typical way to deal with taking care of the throughput issue, one dependent on boundless misguided judgments, is to buy more bandwidth. When the objective is to help an enormous number of clients accomplish higher total throughput, some improvement might be figured out. However, a greater communication channel doesn't address the hidden impediment of TCP that outcomes in greatly wasteful use of bandwidth in high-latency situations. In reality, more bandwidth in a real sense has no effect in numerous reasonable circumstances: over a base bandwidth-latency product threshold, TCP throughput is an element of latency and loss of packets and is not dependent on bandwidth. So almost certainly, the exchange time for a given record will be the very same over the higher bandwidth association.

Researches and studies kept on going to find alternative ways to minimize the addressed limitations of TCP/IP network. Researchers proposed and went for different approaches to rethink the principle of network architecture. They have been able to come up with different solutions and some of them have positive outcomes over the traditional approach. These outcomes encourage the frontrunners in this aspect to look for new and different solutions.

The Named Data Networking (NDN) has been brought to attention as the future of internet architecture with a goal to overcoming the challenges of today’s network system. The strength of this paradigm is that content or data is the most important element of it. In NDN, an Interest packet is sent from the Consumer in the network for a specific data element, and the Producer replies with the requested data. The main advantage of NDN over the TCP/IP is, it doesn't have to resolve hostnames while the communication is happening. This approach immediately eliminates the overhead in the network, which can be found in the case of the traditional ones, and makes use of bandwidth more efficient. The two most integral features of NDN are Interest Aggregation and Caching. Interest Aggregation combines similar types of Interests allowing better resource utilization. Caching policy enables NDN to store recently acquired data in temporary local storage to satisfy future Interests for the same Data which makes the communication mechanism faster. This approach is also highly effective when there is a failure in the transmission link and the topology is not fixed.

The main idea behind the development of NDN architecture is to improve the efficiency and structure of the transport layer and the network layers. But NDN can also provide a positive impact on the efficacy of service layers. Specifically, at a scenario when there are services involved where the resources are changing dynamically along with the demands. In an edge network system, the services are provided by the edge server and NDN can be useful in this case. In this kind of network, there is a complex trade-off between the requirements of the application requirements and available resources where the topology of the network is highly dynamic [1]. Research on this topic provides an insight that NDN is capable of solving critical computation problems and it can re-use the past results [2]. But this study does not contain proper numerical results in terms of providing dynamic service by Named Data Networking.

The research in this paper demonstrates an augmented computing system using NDN in a mobile wireless network system. In this computing system service is provided in the edge node. The scenario considered here includes several mobile users who need to perform computing tasks and they offload them to the edge server. This approach takes full advantage of the NDN capabilities and does not interfere with the default NDN functionalities. The main idea of this research is to demonstrate that, in an augmented computing environment, NDN can significantly improve the computational abilities of mobile devices by using resources available on outside servers (ex. edge or cloud servers).

We have gathered some research questions to move forward with our proposed approach. Such as:

- What are the issues with traditional IP network?
- How NDN can overcome the limitations of the traditional IP network in certain cases?

Phases of the thesis

In chapter 2, we discussed the motivation behind our experiment. We also discussed NDN and its architecture. We have explained the fundamental components of NDN and their functionalities We have also made a comparison between the conventional IP network and NDN to show the benefits of NDN over IP. After that, we discussed several applications of NDN, edge networking and their use cases. In chapter 3, we explained the methodology of the experiment. Here we discussed the tool needed for the experiment and made necessary modifications to perform simulation of our desired scenario and acquire the simulation result. In chapter 4, we demonstrated the numerical result and analyzed it to show the performance of NDN. We have also made a comparison with the traditional network deployment which is IP. In chapter 5, we discussed the research goal and made the conclusion. The last part of the thesis contains the references and appendices.

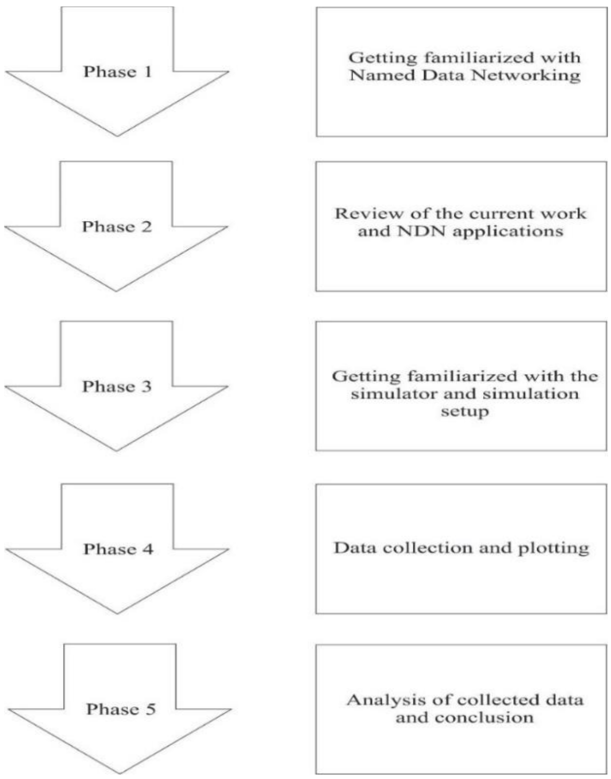

Figure 1. Phases of the thesis

The thesis is divided into the following parts:

Phase 1: In this phase, we gained the basic idea of Named Data Networking. We came to know about the architecture and also learned the functionalities of each component. We also gained the required knowledge to understand how the traditional network system is designed. After that, we got familiarized with the edge network and edge computing system and got to know the challenges of using IP in this architecture.

Phase 2: In this phase, we have studied the researches on NDN and came to know what people are doing right now. This phase helped me to open the mind towards the huge potential of NDN in different sectors.

Phase 3: In this phase, we came to know about the tools to simulate our intended scenarios. Here we came to know about Network Simulator 3 and its functionalities. We have discovered that, to simulate the NDN network, there is a simulator called ndnSIM and it comes as an extension of the Network Simulator 3. Here we have made a simulation of the traditional IP network and then simulated our desired NDN network scenarios.

Phase 4: After the simulation, we achieved some numerical results in this phase. We have plotted those gains in the graph to analyze the effects of different characteristics of NDN. This phase demonstrates the outcome of our research and makes a conclusion of this study.

Phase 5: Here in this phase, we have made our necessary analysis with the collected data from our simulations. We made a comparison with the traditional IP network scenario and also analyzed the effects of different aspects of NDN networks.

2 Background and Overview of NDN

In this section, we broadly discussed NDN. We have explained the functionalities and key features of NDN that make it different from the traditional network system. In addition to that, we have provided some application areas of NDN and got to know how NDN is solving so many problems in different fields. Edge computing is also described in this section to get a broader idea of our research goal. Lastly, we have made a discussion on our considered scenario and how NDN can be useful in certain cases in the network communication system.

2.1 Named Data Networking

The invention of the Named Data Networking (NDN) was done based on an intention to create a new design of Internet architecture. It will overcome the shortcomings that arise because of host addresses that are an integral part of today's communication system. NDN changes the information by putting a name to it and removes the idea of the destination address of it that makes the information the most prioritized element of that system. [3] The traditional Internet architecture puts the security mechanism on the communication channel. But NDN puts the security inside the content and removes the need for security in the communication channel, empowering a few fundamentally versatile communication techniques, for example, caching the data inside the network for efficient use of bandwidth. The venture examines the specialized difficulties that should be addressed to approve NDN as a future Internet design: routing versatility, quick forwarding, trust models, secured network, content assurance and protection, and crucial communication hypothesis. The task utilizes start to finish testbed arrangements, recreation, and hypothetical investigation to assess the proposed design, and is creating determinations and prototype usage of NDN protocols and applications.

When designing the architecture of a communication system, there need to be two things under consideration [4]. The first thing is the namespace choice for data delivery and the second thing is to find an effective technique for delivering the data. Now, for the first consideration, there remain two choices to transfer data from source to its destination. One option is to assign an identifier to the source and destination and another option is to name the data itself.

Today's internet architecture follows the TCP/IP protocol which uses the first choice for communication, which is naming or putting identifiers to the communication endpoints. But NDN [5] [6] takes the second design choice. As a proposed Internet engineering, NDN is intended to organize the universe of computational science, going from IoT sensors to the cloud system, by naming information bits. Named and secured information pieces make the focal point in the NDN network design, and the NDN network layer utilizes application information names to convey. This plan enables the organization to recover named information by whatever methods available; treating network systems, caching and empowers one to get information safe and in a secured way.

Principles of NDN Architecture

NDN provides a new internet architecture that creates an hourglass shape. The architecture provides great compatibility with the current internet infrastructure that is widely established for communication. NDN provides a layer that can work on top of anything, even the IP, and anything can also run over the NDN.

NDN has six principles in terms of architecture. Among them, three of them are motivated by the Internet architecture and the other three are the result of the research and observation from the current solutions of the communication system.

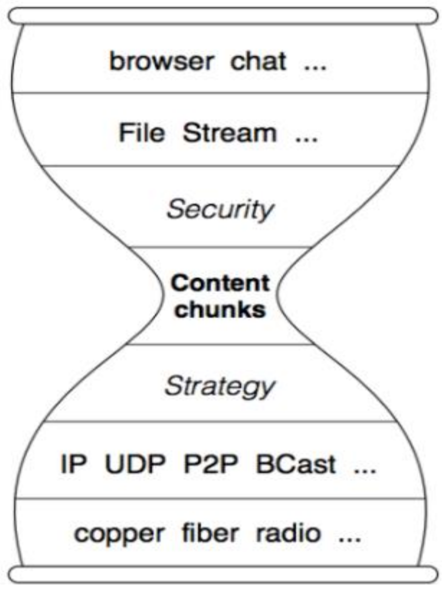

Figure 2. NDN hourglass architecture

The hourglass architecture is the thing that makes the first Internet system rich and incredible. It fixates on an all-inclusive network layer (IP) actualizing the insignificant usefulness essential for worldwide interconnectivity. This "thin waist" has worked as a key factor for the rapid development of the Internet. NDN keeps a similar hourglass-molded architecture.

Routing and forwarding plane detachments have demonstrated essential for Internet improvement. It uses the forwarding technique to send the desired data following the best possible route to the destination. And the routing techniques keep on evolving and research is going on to find the best route for data transmission.

Network traffic should act naturally managing. Flow balanced information conveyance is vital for stable network activity. Since IP performs open circle data conveyance, transport conventions have been corrected to give unicast traffic balance. NDN puts the flow balancing into the thin waist of the architecture.

Security should be incorporated into the design. Security in the current Internet engineering is a reconsideration, not gathering the requests of the present progressively antagonistic environment. NDN gives a fundamental security building block directly at the thin waist by signing all named information. The end-to-end standard empowers the advancement of powerful applications despite disruption in the network. NDN holds and grows this planning rule.

The design ought to encourage user decision and competition where conceivable. Though not a significant factor in the first Internet system, the worldwide organization has instructed us that architecture can't be made from a neutral point of view. NDN puts forth a cognizant attempt to enable end-users and empower competition among the solutions.

The architecture of Named Data Networking

In NDN the communication is initiated by the receivers that are known as Consumers. The involved entities exchange two types of data known as Interest and Data. Both the packets include the name of the desired data.

Types of Packets

- **Interest**: The consumer sends the interest packet and it contains the desired data-name and sends the packet inside the communication network. Then the network router forwards the packet to the desired producer depending on the forwarding strategy
- **Data**: When the interest packet arrives at the designated producer, it will send back the data packet to the requesting consumer along with the name, desired content

and producer key for security purposes. For effective transmission, the packet regarded as data, uses the same route as the interest.

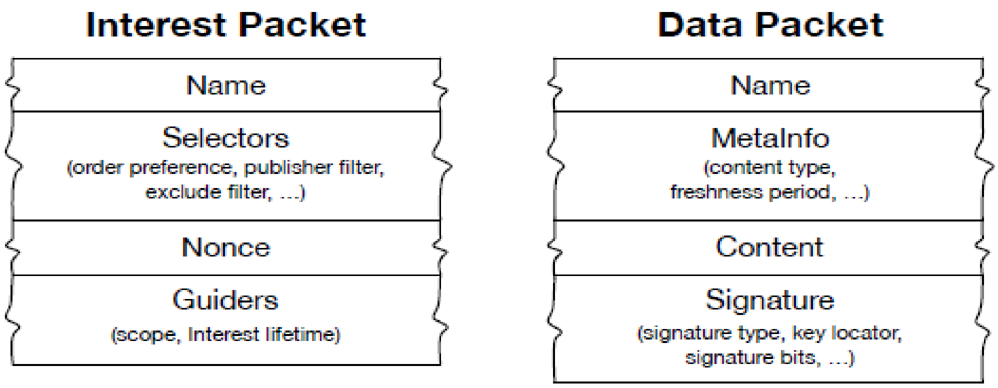

Figure 3. Interest and Data packet structures in NDN

Names

In NDN the data names are structured in a hierarchical manner. It allows the application to name the data according to its need. The name is meaningful to the application and it enables a relationship among the different chunks of data. The NDN router does not know the meaning of the data name but of course, they are aware of the components residing in the name. For example, if we consider a scenario where there are several sensors inside different rooms of different buildings and each of the sensors supplies some data to the server. In NDN the data having the name /building1/floor2/room3 will indicate that it is coming from the third room of the second floor of the first building.

To acquire the information that is produced progressively, consumers should have the ability to develop the name in a deterministic manner for an ideal piece of information by not having recently seen the information. There can be two ways. One can be a deterministic method enables both the producer and consumer to show up at a similar name dependent on data accessible to both. Another way is consumers can acquire information dependent on fractional names. For example, the consumer may start looking for /building1/floor2/room3/data1 and get back information named /building1/floor2/room3/data1 /1/1. After that the consumer would have the ability to indicate later sections and send a request to get that piece of information.
When designing and determining the names for the data in NDN, all the data names should not be unique from a global perspective. The part of the name which is responsible for finding the desired piece of information should be unique. The name which is designed for using at a local scope is used in a local context and for finding the required data, a local routing mechanism can be utilized.

Architecture for Routing

For effective transmission of data and interest packets, NDN routers follow some defined data structures and forwarding mechanisms:

- **Pending Interest Table (PIT):** When a consumer sends the interest, the NDN router forwards them to the desired producer. If the interest is not satisfied et, they are stored in PIT. PIT stores the interest along with the incoming and outgoing interfaces.

- **Forwarding Information Base (FIB):** FIB is a routing table inside the NDN network. It contains information about data names and interfaces. The records in the FIB table are created based on the routing protocol. There may be several outgoing interfaces registered for the same name prefix.

- **Content Store (CS):** In NDN, data is meaningful regardless of its origin and its destination. And it can satisfy any future interest. So the data is stored inside CS. CS is originally a temporary cache, where the data is store after receiving by the router.

- **Forwarding Strategy:** There are several strategies and set of rules that are used to forward interest and data packets in the NDN communication system. These strategies may decide to drop any packets in unfavorable situations, for example, the network is highly congested and there is no more traffic allowed or there is some DDOS attack suspected. The strategy is usually assigned to the data name prefixes. The default strategy is the Best route Forwarding Strategy.

At the point when Interest arrives, firstly NDN router look into the Content Store to see if there is any data available that matches the requirement. If the desired information is there in the router, then NDN router saves that Data in that interface from where the Interest arrives. In any case, the router performs the look up procedure in the PIT to see if there is any Data exists that fulfill the requirements. If actually looks for the name and if same name is found there it stores the interface name of the receiving Interest. If no name is found there, that is matching, it will forward the Interest to the destination, that will produce the desired Data. To forward the data it will look into the FIB and follow the path stored there that was determined by the Forwarding Strategy of the NDN router. There are situations when the router gets Interest with a similar name. In this case, it will advance only the first Interest towards the producer.

At the point when NDN data arrives at the router, it looks for the records that have matching names. When matching names are found, the Data is sent to all the requesters for that Interest. After that, the entry for that Data is erased from the PIT. But the Data will be there in the Content Store to meet the needs of any future requirements. In the communication mechanism in NDN, the route of Data transfer is always the same as the route of the Interest transfer and reducing packet loss, giving a balance of the stream. On the present Internet, TCP ACKs are implemented to break the deadlock caused by traffic. NDN Interests have an impact like this in controlling traffic.

Security

In NDN, security is incorporated into the information itself, instead of being a component of where, or how, it is acquired. Each piece of information is marked along with the name of it, restricting them in a safe way. The signatures on the information are obligatory. There is no way for an application to be there without security. The signature is an integral part of data in NDN. So when it is embedded inside the NDN data, it can guarantee the originality of that data. Like that, it creates a sense of trust in the consumers to move with the new approach of NDN rather than securing a channel for sending the same data that is implemented in the current settings.

Nonetheless, there is a need for a lot of research and study to make the data secured according to the specification of NDN. Current settings of communication system use a mechanism called public-key cryptography which is not a happily accepted approach because there are some difficulties in using the mechanism. Other than the advanced and special signature, there is a need for adaptable components to supervise the trust of the users. Fundamental research on this security approach shows that NDN has already thought of a way to exploit data security. In NDN, there is a way to send the key as NDN data. So, dissemination of the keys is more efficient in this mechanism.

Difference Between IP and NDN

During the 1960s and 70s when the center of thoughts are hidden the Internet were created, telephony was the lone illustration of fruitful, compelling, worldwide scale communication. Hence while the communication system offered by TCP/IP was extraordinary and pivotal, the difficulty it settled was telephony's: a point-to-point communication between the involved parties. IP has surpassed all assumptions for encouraging pervasive interconnectivity thus empowered sensational changes on the planet that we associate with the Internet.

IP, regardless of being intended for discussions between conveying endpoints, is presently overwhelmingly utilized for content appropriation, both to fixed hosts and progressively to cell phones. Similarly, as telephony would be a helpless vehicle for the transmission content dissemination done by TV and radio, Internet engineering is a helpless match to its essential use today. Likewise, pernicious assaults, pulled in by the huge financial estimation of Internet applications, have become day by day occasions. Following the conversational model of the IP model, numerous endeavors have been dedicated to getting communication channels secured, yet security breaks keep on expanding.
In the traditional IP network, the datagram contains information about the communication endpoints that are IP source and IP destination. But the motivation behind Named Data Networking is naming the datagram. The name of the datagram is structured hierarchically, and it does not contain the source or destination address.

In today's IP architecture, datagram sending is motivated by "where" it is sending rather than "what" it is sending. After that, the intermediate entities take care of the mapping between the application and network. In contrast to the conventional architecture, the NDN datagram removes all the communication inefficacy because there remains no need for the middleware.

Communication channels can contain different types of data and the data can be about anything. In the IP network, the same security approach is applied and the communication channel between the involved parties is secured by that. This convention often is not enough to meet the requirements. In the case of NDN, all the data sent by the producer is signed by the producer itself and also verified by the consumer. That provides sufficient security to the information. For example, Ip can't tell from the data, if this was sent by a bank or not. But NDN can easily identify this.
NDN provides a hierarchical and unique naming to the data and the naming is not dependent on its source. That unique naming scheme provides meaning in the network system that can be cached there. It is important to satisfy future requirements. Also, this facilitates the routers to maintain the data plane state which eventually opens a door of new opportunities in contrast to IP. The unique naming also removes the problem of data looping inside the network and the nodes can distribute data effectively.

NDN provides a lot of benefits over the IP network and it can show a lot of opportunities ahead also with some issues. One of the issues is the naming of the datagram.
It is not still well established how the naming could be done efficiently by the application to ensure uninterrupted data delivery.

Named base routing creates an issue in the aspect of scalability. As we know IP uses IP addresses and the address is finite. But with NDN we can have the infinite naming of the packets. In today's internet, IP prefix aggregation is used to scale the internet routing state. NDN can also control the name space in the global routing technique in an effective manner.
ASICS are studied to see how the packet can be forwarded via wire. It is discovered that IP packets can be forwarded by them at a wire rate. In the case of NDN, packets are different from the IP packet and they have the data name of variable lengths. So methods for effective transmission at a wire-rate are still under investigation.
Security of the communication channel is decoupled from the data itself in the IP network. In NDN, the data signing mechanism makes it more user-centric. There is still a lot of research to be made to find an effective mechanism to implement security on the named data.

2.2 Application of NDN

Named Data Networking has been invented with some impactful and distinctive features that allow using this network architecture in so many sectors. [7] Among all other characteristics of NDN naming is an important aspect that can be designed in an application-specific manner and different types of applications can use different kinds of naming schemes. Other than that, other functional features such as caching inside the network, security in the data itself, interest aggregation, etc. can support different types of applications and provide benefits over the traditional TCP/IP network infrastructure.

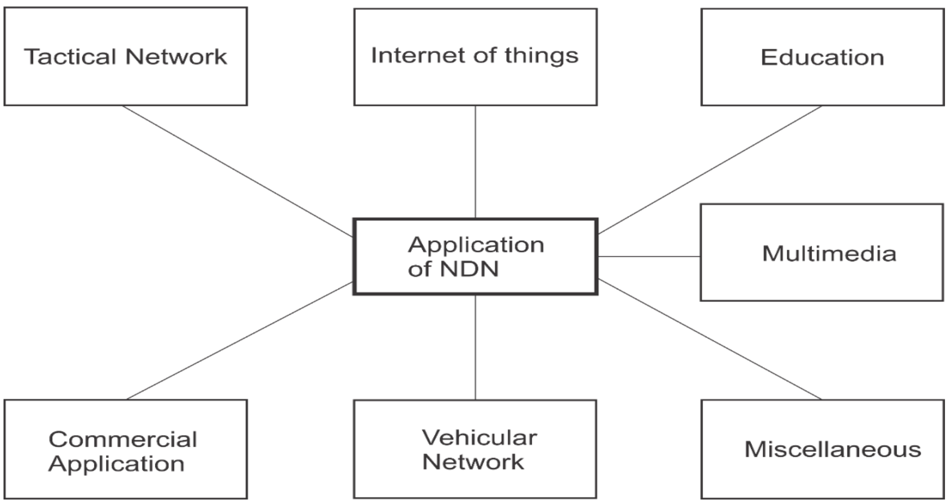

Figure 4. Applications of NDN

Tactical Network

Normally, data transfer happens in inheritance tactical networks through circuit-switched techniques that save segments of accessible data transfer capacity. These techniques typically bring about underutilization. In the present military networks, generally, all these frameworks are moving towards net-centric structures, making packet-based usefulness imperative to these networks. The packet-based systems anyway experience the ill effects of their own disadvantages too, like huge control overhead and variable start to finish delays. Whenever utilized accurately NDNs offer incredible advantages to the presentation of tactical networks as far as data transmission preservation and effectiveness. In conventional packet-based networks, information is sent across joins that have been orchestrated into start to finish ways. The networks utilize either TCP or UDP for message transmission. Both of these protocols work in an end-to-end style, communicating and accepting information and affirmations among sources and destinations. In NDN, information is communicated distinctly in light of an interest message. These interest messages indicate the name of the content the client is interested in and not really the physical location of the node putting away/delivering this information. This once more is the key distinction from conventional IP networks. While IP networks use the address of source and destination to route and forward information, NDNs utilize carefully the name of the content for information sending.

Data packet construction and delivery in NDN is significantly easier than interest packet delivery. Data packets are not steered however essentially follow the way of the interest messages back to the first requester(s). A longest-match query of an information packet's content name is done upon packet appearance. Moreover, information recovery can happen in more than a few ways. By using its cache, nodes can store and advance their messages, which advance correspondence over irregular connections.

In this paper [8]. This framework is intended to supplement conventional communication techniques, as it can uphold correspondence in tested combat zone conditions where the dynamicity of the radio network system and the shortfall of end-to-end availability once in a while block constructing ad hoc communication systems.

This methodology depends on a mix of the Publish/Subscribe paragon with the standards of an astute and disturbance-tolerant network system. The model they used is document-arranged instead of being essentially message-arranged. A document is an organized snippet of data a portable host can either publish in the network or get from the communication network. Its fundamental components are a descriptor and a payload. The descriptor is intended to give meta-data about the payload, like its starting point as well as its objective, its kid, a list of features, and so forth As indicated by the Publish/Subscribe model, a host is as it was intended to get documents for which it has subscribed for. An intriguing outcome of this model is that it yields a reasonable decoupling between data producers and consumers. A document can be published even at the point when no subscriber is free to get it, and it will be gotten later by subscribers even after the publisher has moved away or left the organization out and out (time decoupling). A document can likewise be published without respect to the character of expected subscribers (tending to space decoupling).

They built up a middleware platform. This platform is named DoDWAN (for Document Dissemination in disconnected mobile Wireless Ad Hoc Networks). DoDWAN furnishes a top-level application facility with a publish/subscribe API. Through this API an application system can publish a document and subscribe to get explicit sorts of documents.

The model they characterized is viable at spreading particular data in an incompletely associated ad hoc system. Most prominently the middleware they created dependent on this model can proficiently drive strategic radios in a combat zone climate.

Internet of Things (IoT)

The plan of a networking design that interconnects a colossal ecosystem, where things might be asset compelled or potentially portable and furthermore the traffic designs are profoundly heterogeneous, presents extraordinary difficulties. Agile network arrangement, security, versatility, robustness, dependability are a couple of the necessities that IoT should uphold. Because of its intrinsic characteristics, NDN can address a few IoT prerequisites by straightforwardly overseeing a few functionalities at the network layer [9]. Because of the utilization of hierarchical, application-explicit namespaces and the shrewd forwarding dependent on PIT and FIB structures, NDN can offer simple, hearty and versatile information recovery.

NDN-BMS [10] is an application-driven undertaking that plans also, actualizes an NDN-based structure the board framework to be utilized by the office the board staff. The prototype sent on the UCLA grounds catches files, also, pictures time-series information created by industry-standard sensors situated in grounds structures. In NDN-BMS, the sensor information namespace depends on naming the things being estimated, for example, electrical momentum and chilled water stream, as per the actual chain of importance of the structure. For instance, the prefix "/bms/building1/floor2/room2 covers the information produced on the first floor of Room number 2 on Building number 1, including such kid Information protests as ".../current/201602101210" for the current draw estimated for the room at 12:10 p.m. on February 10, 2016. This namespace encourages steering and reserving: each node could enroll the actual area name it addresses with its upstream node, hope to get Interest for information produced without help from any other node or its downstream node, and reserve information from downstream for later access. The framework utilizes a fundamental encryption-based admittance access control inside a gathering of legal users.

A smart city improves the nature of its residents' lives by giving straightforward entry to universal administrations through incorporation utilizing correspondence frameworks at the establishment. Moreover, ITS assumes a significant part in making a metropolitan region into a smart city. The current IP-based arrangements for ITS have skewed the exhibition because of high interest for information progressing, particularly when the consumers become the producers. In the interim, NDN has advanced as a promising future Internet design and is being researched broadly. The paper [11] centers around the materialness of NDN in ITS for smart urban communities to enhance applications, satisfy portability, and guarantee security. ITS applications, regardless of whether focused on security or non-wellbeing, require information without concern with respect to the character area of its supplier. That information can be spatial, fleeting, or spatiotemporal in nature. For example, an application may require spatiotemporal data, for example, street condition, climate, parking accessibility, traffic volume data, condition of dumpsters for waste vehicles, a patient's current/past fundamental signs data for a rescue vehicle, etc., for a particular period and additionally area.

NDN utilizes progressive naming to the course and distinguishes the information or chunk of information in the communication system. There are a couple of naming plans regularly proposed for vehicular organizations [12] furthermore, smart city [13] sector. Here they proposed the adjusted naming construction for ITS. The smart city (SC:) is the identifier for the naming plan for the smart city climate. The help identifier name segment implies the kind of method or system offered by ITS. Then they another type of information is used to provide the three-dimensional location information, time and date information and information about the scope. After that their attribute-value pair information is used to construct the data perfectly. They also reconstructed or modified the current caching policy which defines the time of data exists in the network cache. Face management is also another part of their proposal where they monitor all the data transmission using the faces. The faces are given some weights depending on different communication statistics like the number of satisfied interests and so on. The faces with higher weights are prioritized for communication of the prioritized data. This approach decreases the number of lost messages and timely delivery. In any communication architecture, segmentation of transferred data is inevitable. Here, they use an effective numbering strategy for the data segments that are given by the provider node or the adjacent nodes. In the traditional NDN, there is no connection establishment that occurs before sending the interest packet. In this case, there is an acknowledgment and negative acknowledgment method used to guarantee the successful delivery of data and interest packets. In contrast to the traditional NDN where the interest is broadcast to all the faces, this approach reduces the broadcast of messages depending on different network parameters. The approach also modifies the PIT mechanism. NDN is usually more consumer-centric. Their proposed approach introduces a push-based forwarding strategy for the mission-critical data.

A smart home architecture equipped with different sensors and actuators is introduced here [14]. Their implementation contains a Home Server (HS), different sensors and actuators, private cloud, push and pull service and it uses an implementation-specific naming scheme. HS gathers all the information from different sensors introduced in the various zones (e.g., rooms, back yard). For greatest accessibility and particular utilization, they expect to be that the cached information/input can be put away in a PC. In generally existing gadget-driven smart home correspondence plans, there is one HS that collaborates with all the detecting gadgets for the required information and will in general send criticism to the client utilizing Bluetooth, WLAN (Internet), and show units introduced inside the home. If there should be an occurrence of any communication blunder between any sensor/actuator and the HS, noxious application conduct is anticipated. They decouple the HS and sensors/actuators, where the client can recover criticism straightforwardly from a particular sensor, by communicating a packet containing Interest for specific data, given that the client is right up-front vicinity of the sensor. In this proposed approach, sensors can speak with the HS and with the client at the equivalent time. In the previous case, the HS broadcast Interest bundles intermittently to acquire the detected data and show it on an appended screen. In the last case, the client can communicate a similar Interest parcel and gather the information from the closest accessible source, instead of piggybacking the IP address of the HS and the traffic crossing from one room of the home right to arrive at the HS or PC. Consequently, NDN makes it simpler to give local and universal admittance to home computerization. Here an application running over a PC with restricted access can recover information from the HS through an Interest packet. At the point when the client is outside the home, the PC application can send basic information to the client through push-based sending in NDN [15]. Through the PC information assortment, it can figure measurements, anticipate basic occasions, alert the client, and let the HS settle on choices for actuators even without the client. The most broadly utilized naming plans are flat and hierarchical naming. Flat names are commonly gotten through hash calculations applied to contents and are not handily appointed to dynamically detected information substances that are not yet detected/distributed. Accordingly, it considers progressive naming in this work. The naming plan should have the option to distinguish either assignment to perform by the actuator(s) or then again, the sort of data needed from the sensor(s). Subsequently, they distinguish two namespaces that can be utilized by the HS for these reasons.

Interest Satisfaction Rate (ISR) was compared against the distance of different sensors and also against the time intervals of the interest generation. Their result showed a 90% better result when the time interval is 15 seconds. It is also found that the range of the sensors do not affect ISR

Education

The fundamental point of these systems is to introduce an essential presentation of the communication system improvement throughout the time. The significant applications are to set up instructive organizations like, gathering rooms and to facilitate ad-hoc networks during gatherings, meetings, and talks. For the better agreement of two communication system improvements, IP-based systems architecture and information-driven network, an elaborated investigation is introduced in [16]

In this paper [17] the plan of an audio conference tool is introduced, which is their push to investigate application plans on top of Named Data Networking. Instead of depending on brought together administrations as current executions do, Audio Conference Tool (ACT) adopts a named information strategy to find progressing meetings just as speakers in every meeting, and to bring voice information from each speaker.

ACT permits every client to recover a rundown of booked or progressing conferences and to report another conference to likely members. To recover conference data, a client should know the name for the conference data information deduced before she has any information about the conferences to be found. Subsequently, conference initiators and members should concede to the name prefix for conference depiction information by following some settled application naming methodology. Following the predetermined naming methodology, a conference initiator declares the conference by making the conference portrayal information with a legitimate name under the conference list prefix. The members of a conference fall into two classes, the speaker(s) who creates the voice information, and the audience members who demand the voice information. ACT just has to know the speakers in a conference to recover information from them. For limited scope conferences where each member is likely to talk, every client is both a speaker and an audience. For huge conferences, for the most part, talking, there are fewer speakers than audience members. ACT scales well by following just a rundown of dynamic speakers, rather than every one of the members. The discovery of speakers is made when a client joins a conference. The method of the discovery of the speakers is done similarly to the conference discovery.

A topology-dependent name prefix (e.g. /ucla.edu/cs/zhenkai) is used by each speaker for its voice information. Because a speaker may generate multiple information streams, further name components like device ID and sound codec can be appended to the name prefix to identify each stream. A device ID represents the actual device that produced the information stream, and it ought to be unique inside the local scope of the network with the goal that the access router realizes where to advance Interests for a specific information stream. To ensure uniqueness, a 32- bit irregular string can be used for the device ID. A speaker ought to include all name prefixes for all the information streams it produces in the speaker description information.
Packets created by ACT are digitally endorsed accordingly of implicit security crude in NDN: each NDN packet name is bound to the content in it with a mark, guaranteeing the genuineness of voice information.

ACT uses the named data approach to remove the repeated retransmissions of information packets. It exploits NDN to shape a totally appropriated plan that is strong against the failure of components. If a failure occurs, all associated members can proceed with their conferences. In the ACT, nothing should be changed if the moving node is an audience: it basically begins sending Interests for voice information when it is re-connected. Also, a conference declaration is as yet reachable in any event, when the initiator moves, as the Interest is broadcasted in this approach.

Multimedia

NDN-RTC [18] developed as a videoconferencing library that utilizes Named Data Networking (NDN), which is regarded as a proposed future Internet architecture. It was intended to give a facility to experimental exploration in low-latency, continuous media communication over NDN. It expects to give a user experience like Skype or Google Hangouts. It implements a recipient-driven methodology that exploits NDN's name-based forwarding methodology, signed data, caching technique and so on. As executed, NDN-RTC employs broadly utilized open source libraries, including the WebRTC library, VP9 codec, and OpenFEC for forward error rectification.

There are two tasks to be performed in the NDN-RTC library: producer as well as consumer. In bidirectional data transmission, applications utilize the library to assume the two entities. The library is based on a consumer-driven methodology straightforwardly following NDN's Interest-Data trade model. Conversely to the sender-driven methodology of traditional IP-based RTC, the producer distributes information to the storage inside the network at its own speed, while the consumer demands information on a case-by-case basis and deals with the connection between active Interests, approaching information clusters.

The producer's fundamental undertakings are to procure video and sound information from media inputs, encode them, marshal the information into parcels, name those bundles, sign them, and store them in an application-level store that will non-concurrently react to approaching Interests. The NDN-RTC namespace defines names for media (fragmented video packets and packaged sound samples), information on error correction, and metadata. The namespace is intended to effectively uphold a consumer-driven network. NDN-RTC handles sound and video transfers freely, which empowers the library to help sound-only streaming, and for applications to focus on sound over video for expanded nature of involvement for intuitive discussions in bursty or low-bitrate situations. The Producer takes data from input media and signs them. Then it puts the information into a network storage and cache. The input data contains the data from the input stream as well as metadata.

In NDN-RTC's beneficiary driven design, the consumer expects to pick the most suitable media stream data transmission from those given by the producer, e.g., by observing the condition of the network; get and, if vital, reassemble media in the right sequence for playback; moderate, as far as could be expected, the effect of network latency and bundle drops on the watcher's nature of involvement. The consumer executes pipelining of the Interests and information buffering.

NDN-RTC has shown that communication in a real-time scenario utilizing NDN is reasonable with the current open-source execution and on the current NDN testbed.

Commercial

The primary things upheld under these applications are electronic payment support from any place, consistency support among various datasets, record facilitating administrations for document synchronization, vehicular network like information assortment from moving nodes, information dispersal, e.g., news, street conditions, climate, and so on, among vehicles.

File synchronization is a valuable application that gives information replication over an assortment of hosts to help sharing, reinforcement, and far-off access. FileSync/NDN [19] is a dispersed application that carries out file synchronization utilizing Named Data Networking. Its motivation is to offer support like DropBox, with upgrades. In contrast to DropBox, FileSync/NDN receives a peer-to-peer (P2P) model that doesn't require centralization of information, and it utilizes NDN to diminish the traffic that would be available in a TCP/IP-based P2P approach. FileSync/NDN exploits a few highlights of NDN to improve the dispersion and security of the data it gives. CCNx Synchronization Protocol (CCNx-SYNC) synchronizes named information saved to repositories under guaranteed name prefix, empowering applications to synchronize content by basically saving it to their store under that prefix. It utilizes the CCNx-SYNC to keep up the consistency of its data across numerous nodes on the organization.
To arrange FileSync/NDN to synchronize a directory with different host machines, a client gives (1) a local directory to hold the files to be shared; (2) the topological and content prefixes related with the ideal common directory, subsequently characterizing the slice for CCNx-SYNC. Different clients wishing to share this folder should comparably give their local occurrence of FileSync/NDN with the indistinguishable topological and content prefixes with the goal that a similar slice can be saved to each of their local stores. Each file in the common directory is composed by FileSync/NDN to a local CCNx store as divided, formed contents sticking to the naming methods of repositories.

Each host advises any remaining hosts regarding local changes to the synchronized assortment by producing a preview addressing the total condition of the local common directory, just as a record of cancellations, and adding it to the assortment utilizing a novel name. Its substance is synchronized in a similar way as different files in the common directory by the fundamental CCNx-SYNC system.

The experiment was done to observe the time it takes to accommodate a common directory and the number of file refreshes contained in a snapshot. It was observed that the method developed for the reconciliation of common directory performs linearly. From this experimentation, all things considered, it took the application 0.35 seconds to proliferate the worldwide depiction to all nodes. The standard deviation was 97 milliseconds.

Vehicular Network

In the traditional VNDN arrangements, a huge number of vehicles with high mobility are engaged with sending messages, so it is difficult to accomplish demand total that is the principal benefit of NDN and can help lessen the data procurement delay. [20] [21]Additionally, the arrangements utilize flooding to accomplish information recovery without consumer mobility uphold, which builds the information securing cost and diminishes the information obtaining achievement rate. In this paper, [22] they tried VNDN system furthermore, expect to decrease the information securing delay and cost and improve the information obtaining achievement rate. In this system, the cluster chain vehicular system is developed to improve the steadiness of the system. Considering cluster chains, consumers can utilize demand total and unicast to secure information from the closest provider. In addition, consumer mobility is upheld to ensure fruitful recovery of information regardless of the high mobility of vehicles. This system is quantitatively assessed. As per the test results, contrasted and the VNDN approaches, their structure lessens the information securing cost and delay by almost 54.6% and 12.3%, individually, and improves the information recovery achievement rate by almost 7.9%.
In NDN, IP-based network has been handled by naming data packet in the network, SDN proposes to decouple the control and data planes without the need of actual interfaces with switches and routers. Both NDN and SDN additionally support data transmission through heterogeneous interfaces and have been as of late examined for vehicular networks. Traditional Vehicular Networks (VN) are in light of the IP-based system, which is inclined to a few issues because of the powerful network topology among different components. In this article [23], they present a design that joins SDN functionalities inside VNs to recover the necessary content utilizing NDN. The architecture consists of several components such as SDN Controller, Caching, Naming of Contents, Intelligent Forwarding, Push-based Forwarding, Intrinsic Data Security, Congestion Control, Topology Indicator, Content Caching Manager, State Information.

The SDN controller is a network element that has a general network view to adequately coordinate the network components to effectively play out the NDN activities including content caching, Interest and Data message forwarding and so forth. It chooses the achievable caching location(s) to dodge superfluous duplicates of the substance to adjust the CS limit furthermore, query overhead in the network. For the situation of an enormous substance, every Data message conveys a chunk of the data. In the VN situation, each name segment characterizes the relationship between data or data fragments. As the content looking and forwarding choices rely upon its name, the name contains essential segments to absolutely and productively get the information from the network. FIB contains name prefix, active face ID(s), and some essential values including the rank of the face(s). Each time an Interest is fulfilled through a particular face, its rank is expanded, which makes it more reasonable for Interest forwarding later on. This rank number is utilized to focus on the face and its inclination to forward Interests. Push-based forwarding is also an important part here which is much impactful in case of a hazardous situation. For the security of the transmitted data, the SDN controller provides the security policies that are agreed upon between the Interest and Data provider.

While cellular networks have been viewed as the lone worldwide wireless framework, in actuality they experience the ill effects of spectrum shortage and coverage constraints. Simultaneously vehicles are being equipped with computational power, storage, and multiple correspondence interfaces via various correspondence advances that can be misused to exploit pioneering connectivity with different vehicles and actually moving information over distance. The paper [24] delineates NDN's promising potential in providing a bringing together design that empowers organizing among all computing devices autonomous from whether they are associated through wired framework or ad hoc.
Though Named Data Networking (NDN) has been thought of as a promising network system design for Vehicular Ad-Hoc Organizations (VANETs), interest sending in NDN endures severe issues in a vehicular environment. Broadcast storm brings about much loss of packets and colossal transmission overhead. Likewise, connect separation brought about by exceptionally powerful geography prompts low packet delivery proportion. Then again, traffic information is assuming huge parts in VANETs since they are fundamental in varieties of Intelligent Transportation System (ITS) applications. In this way, a proficient NDN sending procedure utilizing topographical attributes to retrieve traffic data is earnestly required. In this paper [25], they demonstrated Density-Aware Delay-Tolerant (DADT) Interest sending procedure to retrieve traffic data in vehicular NDN to improve packet delivery percentage. DADT explicitly addresses data retrieval during network disturbances utilizing Delay-Tolerant Networking (DTN). It makes retransmission choice dependent on directional organization thickness. Additionally, DADT mitigates broadcast storm by utilizing a rebroadcast conceding timer. The outcome at DADT is

compared against different procedures through recreation and the outcomes show that it can achieve a higher rate of satisfaction while keeping low transmission overhead.

Dissimilar to the host-based communication system, ICN advances information names as a first-class resident in the communication system. However, the conventional ICN name-based data routing requires Interests to be steered by name to the closest reproduction, inferring the Interests are overflowed in VANET. This presents a huge overhead and subsequently corrupts remote network execution. To keep up the efficacy of ICN execution in VANET, [26] propose an entrepreneurial geo-inspired content-based directing strategy. They propose Last Encounter Content Routing (LER), which monitors content areas utilizing last experience data also, performs shrewd topographical directing. To diminish content discovery overhead, the content areas are kept up at every node. At the point when two vehicles experience each other, they trade their content records and the content areas known to them. Along these lines, the neighborhood content area data is refreshed each time when a vehicle experiences another. Afterward, if this vehicle gets an Interest with a name that matches an entry in the list of content areas, it forwards this Interest by geo-directing as opposed to flooding, which is broadly utilized in current ICN. The Interest is possibly flooded from the outset when the area is obscure, and just until a transfer discovered coordinating area information. With this data, the Interests can be geo-routed rather than being flooded to decrease the blockage level of the whole communication system. The reenactment results show that the proposed strategy decreases the extent of flooding to under two jumps and improves retrieval rate by 1.42 occasions over flooding-based techniques.
Innovation progresses in computational science and wireless communication have made it financially plausible for producers to gather information from all the vehicles to screen their activities and identify any expected issues. To make this a reality requires another engineering that can adequately deal with vehicle mobility, interrupted network, what's more, information security, just as scale to a huge number of vehicles. This study [27] addresses these plan difficulties by investigating the course of Named Date Networking. Following the NDN course and growing the underlying plan of conventional content naming, they planned a profoundly proficient, solid and secure vehicle information assortment framework, DMNDI. They directed the starter assessment of the DMND plan through reproduction in Qualnet and contrasted the DMND execution and that of a framework running MobileIP. The outcome shows that, when data distributers (vehicles) are fixed, over 99% of assortment solicitations can effectively pull information bundles back; in any event, when vehicles move at a rapid of 40-50 meters each second (89.48-111.8 miles/hour), DMND can, in any case, hold its high effectiveness of 97% of information

answers. Conversely, under a similar reproduction test setting, the demand answer proportion of MobileIP drops from 97.9% for static distributers to 9.6% when distributers are moving at a speed of 10-20 meters/second (22.37-44.74 miles/hour).

Miscellaneous

Numerous scientists have contemplated the utilization of NDN in various areas, for example, home networking, gaming, multi-sensor synergistic detecting, texting, traffic control, crisis correspondence support during a catastrophe, creating APIs for communication, taking care of error-prone hubs, environment exploration and high energy and atomic material science, and so on.

Named Data Networking is a network architecture that in the future can dislodge IP from its present role of a global inter-networking layer and furthermore to have an edge in such regions of networking as PAN (Personal Area Networking) where IP has never been generally utilized. This task investigates the issue of fitting the Named Data Networking request-response model into compelled Bluetooth networking stack for both mobile and desktop use cases. The paper [28] gives a cross-platform intermediary layer, which works between NDN stack also, Bluetooth stack to accomplish NDN network over Bluetooth network.

NDNBlue performs on devices running o Linux and Android. NDNBlue is viable with BlueZ which is the implemented Bluetooth protocol stack on the Linux platform. NDNBlue needs the NDN software router and its C library to be introduced on Linux and a similar application on Android. To add a Bluetooth interface to the NDN programming router, NDNBlue follows a comparable methodology as NDNLP. NDNBlue is carried out as an NDN application acting as a proxy between Bluetooth platform subordinate stack and NDN autonomous stack and runs on Linux and Android. Then again, NDNBlue could be coordinated into the software router code base of NDN, however, this choice was ignored because of continuous changes of the NDN daemon, which would prompt huge upkeep expenses and strength issues of this arrangement. As an outcome, by working between the Bluetooth stack and the NDN programming router, NDNBlue transfers data and empowers a client to add a Bluetooth face to the router. As a transport layer protocol, NDNBlue takes advantage of RFCOMM. Moreover, Bluetooth Service Discovery Protocol (SDP) is used here for registering a server application. For this approach clients that are intended to connect to the server can discover the RFCOMM channel. NDNBlue is confined to just a single Bluetooth connection per node. Since SDP is upheld, it is conceivable to utilize progressively assigned RFCOMM channels in worker nodes. That is how it is possible to mitigate this restriction.

In another research [29] the author demonstrated that mobile network efficacy can be increased by including Named Data Networking as an internet architecture. To support their cause, they included a forwarding strategy named Listen First, Broadcast later (LFBL) [30] which is developed for wireless communication. The motivation behind this strategy is to include Named Data Networking in the wireless network. LFBL incorporates a little modification over NDN's three-way exchange for ad hoc mobile networks, which are, name prefix announcement, interest forwarding, data return.

In this approach, A node N floods a packet request that conveys the name of the mentioned application information. Any node or on the other hand nodes that end up having that named information send an acknowledgment, which is sent to N utilizing data gathered by the transitional nodes during the request stage. At long last, N sends an affirmation as encouragement or debilitate these reactions. NDN broadcasts name prefix like other routing protocols, while LFBL saves the prefix engendering by paying the cost of flooding demands. At intermediate nodes, the obligation regarding forwarding choices is set unequivocally in the possession of the receiving node, as opposed to the sender. Subsequent to getting a packet, a potential forwarder stops to listen to the channel, holding back to check whether a more ideal node forwards the packet first. Something else, it forwards the actual packet. Still, the method (receiver-based forwarding strategy) is not well examined under the condition, where the nodes are highly dynamic.

2.3 Edge Computing

Edge computing is regarded as a data handling design in which customer information is handled at the outskirts of the network system, as near the starting source as could really be expected.

Information is the soul of present-day business, giving significant business knowledge and supporting continuous power over basic business cycles and activities. The present organizations are inundated with an expanse of information, and enormous measures of information can be regularly gathered from sensors and IoT gadgets working continuously from distant areas and inhospitable working conditions anyplace on the planet. In any case, this virtual surge of information is additionally changing the manner in which organizations handle computing. The standard computing system based on a unified server and the regular web isn't appropriate to moving interminably developing ocean of information. Data transfer capacity impediments, issues of transmission latency and eccentric network interruptions would all be able to scheme to hinder such endeavors. Organizations are reacting to these information challenges using edge computing engineering.

Benefits of Edge Computing

The advantages and needs of edge networks are undeniable in the field of modern communication system

Speed of Data Transmission

By and large, speed is totally imperative to the core business. The monetary area's dependence upon high-frequency exchanging calculations, for example, implies that a slowdown of simple milliseconds can have costly results. In the medical services industry, losing a small amount of a second can even involve decisive. Furthermore, for organizations that give information-driven administrations to clients, slacking rates can baffle clients and cause long-haul harm to a brand. Speed is not, at this point simply an upper hand—it is a best practice.

The main advantage of edge computing is its capacity to enhance the performance of the network by diminishing latency. Since IoT devices in edge computing measure information locally or in close by edge servers, the data they gather doesn't need to travel close to the extent it would under a cloud design. It's not difficult to fail to remember that information doesn't travel promptly; it's limited by similar laws of physical science as all the other things in the known universe. Current business fiber-optic innovation permits information to go as quick as 2/3 the speed of light. While that sounds quick, it neglects to consider the sheer measure of information being communicated. With the world expecting to create a huge amount of data, computerized gridlocks are practically ensured.

By handling information nearer to the source and lessening the actual distance it should travel, edge computing can enormously diminish latency. The outcome is higher velocities for end clients, with latency estimated in microseconds as opposed to milliseconds. Taking into account that even a solitary snapshot of latency can cost organizations a lot of money, the speed benefits of edge computing can't be disregarded.

Data Security

The multiplication of IoT devices in edge computing creates an attacking opportunity for the networks, it likewise gives some significant security benefits. Cloud computing system is characteristically unified, which makes it particularly helpless against Distributed Denial of Service (DDoS) assaults and blackouts. Edge computing circulates preparing, storage, and applications across a wide range of servers, which makes it hard for any single interruption to bring down the network.

One significant worry about IoT edge computing gadgets is that they could be utilized as an entry point for cyberattacks, permitting malware or different interruptions to contaminate a network from a specific part. While this is a certified danger, the distributed idea of edge computing design makes it simpler to carry out security conventions that can close undermined parts without closing down the whole network.

Since more information is being prepared on nearby nodes instead of communicating it back to a focal server, edge computing additionally lessens the measure of information really in danger at any one time. There's less information to be caught during travel, and regardless of whether a node is undermined, it will just contain the information it has gathered locally instead of the stash of information that could be uncovered by an undermined worker. Regardless of whether an edge-computing system consolidates particular edge servers, these often give extra safety efforts to prepare for devastating DDoS attacks and similar kinds of cyberthreats. An edge server should offer an assortment of instruments customers can use to get and screen their networks progressively.

Scalable Network

As organizations develop, they can't generally expect their IT foundation needs, and building a committed server farm is a costly recommendation. Notwithstanding the considerable in advance development costs and continuous upkeep, there's additionally the subject of the upcoming necessities. Conventional private offices place a fake requirement on development, securing organizations in conjectures of their future computing needs. If business development surpasses assumptions, they will most likely be unable to exploit openings because of deficient computing assets.

Luckily, the improvement of cloud-based innovation and edge computing has made it simpler than any time in recent memory for organizations to scale their tasks. Progressively, computing, storage, and investigation abilities are being packaged into devices with more modest impressions that can be arranged closer to end clients. Edge frameworks permit organizations to use their devices to extend their edge network's range and capacities. Extending information assortment and examination no longer expects organizations to set up concentrated, private servers, which can be costly to assemble, keep up, and supplant when it's an ideal opportunity to develop once more. By joining colocation services with provincial edge computing servers, associations can grow their edge network reach rapidly and cost-successfully. The adaptability of not depending upon a centralized foundation permits them to adjust rapidly to developing business sectors and scale their information and computing needs more effectively. Edge computing offers an undeniably more affordable course to adaptability, permitting organizations to extend their computing limit through a mix of IoT devices and edge servers. The utilization of edge computing nodes likewise facilitates development costs. Because new device added doesn't force generous transfer speed requests on the center of a network.

Versatile Network

The adaptability of edge computing additionally makes it extraordinarily flexible. By joining forces with neighborhood edge servers, organizations can undoubtedly target attractive business sectors without putting resources into costly framework extensions. Edge servers permit them to service end clients proficiently with minimal actual distance or latency. This is particularly important for content suppliers hoping to convey continuous real-time features. They likewise, permit them to agilely move to different business sectors in case of change in economic conditions.

Edge computing likewise enables IoT devices to assemble exceptional measures of noteworthy information. As opposed to trusting that individuals will sign in with devices and communicate with unified cloud workers, edge computing devices are consistently on, consistently associated, and continually creating information for future examination. The unstructured data assembled by edge networks can either be handled locally to convey fast services or conveyed back to the center of the network where incredible examination and AI projects will analyze it to distinguish patterns and outstanding information focuses. Furnished with this data, organizations can settle on better choices and meet the genuine necessities of the market proficiently. By fusing devices into their edge network design, organizations can offer better services to their clients without totally updating their IT foundation. Service-oriented devices give an energizing scope of conceivable outcomes to associations that esteem development as a method for driving development. It's an enormous advantage for ventures hoping to grow network venture into districts with restricted availability.

Reliable Network

Because of the security advantage of edge computing, it is likely to have reliability in network architecture. With IoT edge computing devices and edge servers situated nearer to end clients, there is less possibility of a network issue in an inaccessible area influencing nearby clients. Indeed, even in case of a close-by server blackout, IoT edge computing devices will keep on working adequately all alone because they handle essential processing capacities locally.

By handling information nearer to the source and focusing on traffic, edge computing decreases the measure of information streaming to and from the essential network, prompting lower latency and quicker speed. The actual distance is critical to efficacy too. By finding edge frameworks in servers topographically nearer to end clients and disseminating handling in like manner, organizations can significantly diminish the distance information should go before services can be conveyed. These edge networks guarantee a quicker, consistent experience for their clients, who hope to approach their content and applications on request anyplace whenever. With so many edge computing devices and edge servers associated with the network, it turns out to be considerably harder for anyone to close down service.

Use cases of Edge Computing

Edge computing has great potential in several sectors of modern life. Below we have discussed the sectors, where edge computing is used in a widespread manner.

Autonomous Vehicle

Driverless vehicles are required to assume control over the highways at any point soon. To work securely, these vehicles should accumulate and examine immense measures of information relating to their environmental factors, climate conditions and contacting different vehicles. They will likewise have to take care of information back to originators to follow use and alerts with nearby city networks. Lamentably, this deluge of sent information will go into a similar progression of traffic created by cells, PCs, and a scope of other associated devices. With such countless extra vehicles assembling and sending information, transfer speed strains are unavoidable if makers don't embrace new computing arrangements. It's one thing for an office PC to encounter inconvenient slack while getting to a network; it's very another for a self-driving vehicle to slack when it's going at a high speed on an open highway.

Edge computing design makes it feasible for autonomous vehicles to gather, interact, and divide information among vehicles and to more extensive networks progressively with basically no latency. Joined with a network of edge servers geologically situated to gather and transfer basic information to municipalities, crisis reaction services, and car producers, edge-empowered vehicles will offer unmatched unwavering quality without modifying network frameworks.

Smart Cities

Metropolitan territories are rapidly turning out to be huge data gathering focuses, with sensors gathering information on traffic designs, utility utilization, and key foundation consistently. While that information permits city officials to react to issues quicker than any time in recent memory, the entirety of that data should be gathered, put away, and broken down before it tends to be put to utilize. Conventional cloud arrangements can't give quick reaction times to the large number of devices working on the edges of the network.

Edge computing design makes it feasible for devices managing utilities and other public services to react to changing conditions in close to real-time. Combined with the rising number of autonomous vehicles and the always extending IoT, smart cities can possibly change how individuals live and use services in a metropolitan environment. Since all edge computing use cases depend upon devices gathering information to do fundamental handling tasks, future cities can respond powerfully to changing conditions as they happen.

Healthcare

The healthcare sector struggled to embrace the most recent IT infrastructure, however edge computing offers energizing additional opportunities for conveying patient care. With IoT devices equipped for conveying huge measures of patient-produced wellbeing information, medical services providers might actually approach basic data about their patients continuously as opposed to interfacing with moderate and inadequate data sets. Clinical devices themselves could likewise be made to assemble and deal with information throughout treatment.

Edge computing could have a critical effect on the conveyance of medical care services to remote areas. Patients in these areas are often at a long distance from the closest healthcare professionals and regardless of whether a healthcare professional assesses them on location, they will most likely be unable to get to significant clinical records. With edge computing, devices could accumulate, store, and convey that data continuously, and even utilize their handling abilities to suggest medicines.

Industrial Manufacturing

Maybe no industry stands to profit more from IoT devices than the manufacturing area. By fusing information storage and computing into modern gear, producers can accumulate information that will consider better prescient maintenance and energy proficiency, permitting them to diminish expenses and energy utilization while keeping up the better unwavering quality and gainful uptime. Smart manufacturing procedures implemented by continuous information assortment and examination will likewise assist organizations to more readily fulfill customer needs.

Edge computing can likewise give extraordinary benefits to ventures working where transmission capacity is low or non-existent. Offshore oil rigs, for example, can use edge computing design to accumulate, screen, and process information on an assortment of natural elements without relying on an inaccessible server.

Financial Sector

Banking companies are receiving edge computing related to smartphone applications to target services more readily to clients. They're likewise fusing similar standards to furnish ATMs and booths with the capacity to accumulate and handle information, making them more responsive and permitting them to offer a more extensive set-up of features.
For high-volume account firms managing in hedge reserves and different business sectors, even a millisecond of delay in an exchanging calculation can mean a huge amount of loss. Edge computing design permits them to put servers close to stock trades all throughout the planet to run asset escalated calculations as near the wellspring of information as could really be expected. This furnishes them with the most precise and exceptional data to keep their business moving.

Challenges of Edge Computing

Edge computing is still at an early stage of its development and a structure to encourage this isn't yet accessible. Such systems will require to fulfill necessities, for example, application advancement to measure demands continuously on edge hubs. Current cloud computing systems, for example, the Amazon Web Service11, Google App Engine13

Microsoft Azure12, etc. can uphold information concentrated applications, however, carrying out constant information preparation at the edge of the network is yet an open exploration zone. Also, the necessity of sending application responsibilities on edge hubs should be well perceived. Deployment techniques - where to put a responsibility, association arrangements - when to utilize edge hubs and heterogeneity - how to manage various kinds of nodes need to be considered for conveying applications on the edge. According to [31] edge computing faces some challenges like General-purpose computing in edge nodes, Discovering edge nodes, Partitioning and offloading tasks, Uncompromising Quality of Service and Quality of Experiences and Use of edge nodes publicly and securely.

In principle, edge computing can be encouraged on a few hubs that are situated between the edge device and the cloud. The nodes can include bae station, router, access point and so on. If we consider base station, for example, it uses Digital Signal Processors (DSPs) that are tweaked to the responsibilities they handle. Practically speaking, base stations may not be appropriate for taking care of logical jobs basically because DSPs are not intended for general reason computing. There is research in updating the assets of edge hubs to help broadly useful computing. For instance, a remote home switch can be moved up to help extra responsibilities. [32]
Getting advantage of the edge of the network needs discovery systems to discover fitting hubs that can be utilized in a decentralized cloud setup. These components can not be manual because of the huge volume of devices that will be accessible at this layer. Additionally, they should provide supply for heterogeneous devices from various generations just as current jobs, for instance, huge scope AI assignments, which were beforehand not thought of. Benchmarking techniques should be essentially fast in spreading the word about the accessibility and capacity of assets. These instruments should consider consistent joining (and expulsion) of hubs in the computational work process at various progressive levels without increasing delay or trading of the client experience. Dependably and proactively managing deficiencies on the hub, automatically recuperating from them will be expected. Existing strategies utilized in the cloud will not be feasible in this setting for the discovery of edge hubs.

Advancing distributed computing systems have come about in the advancement of various procedures to encourage the dividing of tasks that can be executed at different geographic areas [33]. For instance, work processes are divided for execution in various areas. Assignment dividing is normally communicated unequivocally in a language or the management system. Nonetheless, utilizing edge hubs for offloading computational processes represents the test of not just parceling computational job effectively, yet doing this in a mechanized way without fundamentally needing to expressly characterize the abilities or area of edge hubs. The client of a language that can use edge hubs may expect adaptability to characterize a calculation pipeline - progressively in a grouping for instance, first at the server then at the edge hubs or first at the edge hub and afterward at the server or possibly over numerous edge hubs at the same time. Eventually, there emerges the need for creating schedulers that send apportioned assignments onto edge hubs.

Quality conveyed by the edge hubs can be caught by QoS also, quality conveyed to the client by QoE. One rule that should be embraced in edge computing is to not over-burden hubs with the responsibilities that are computationally intensive [34]. The limitation here is to guarantee that the hubs accomplish high throughput and are solid while conveying for their expected responsibilities in the event that they accommodate extra jobs from a server or from edge nodes. Whether or not an edge hub is exploited, the client of an edge device or an information focus anticipates a base degree of service. For instance, when a base station is over-burden, it might influence the service gave to the edge devices that are associated with the base station. An intensive knowledge of the pinnacle long periods of the utilization of edge hubs are required so that jobs can be divided and booked in an adaptable way. The job of a management architecture will be alluring yet raises issues identified with observing, booking and re-planning at the foundation, platform furthermore, application levels.

2.4 Considered Use-case

The goal of this research is to implement a dynamic computing service for resource-heavy tasks. In this service, there will be two types of tasks. One is the data itself and the other is the software that needs to be used to process and compute the data. For example, we consider a scenario, where a mobile user wants to do a 3D rendering of pictures of any object. For that job, there involve two entities. One is the data itself and the other is the software needed for the processing of data. To perform that job, the user device contains a queue of required tasks. The complexity of those tasks is known to the device and the device CPU is aware of the time needed for performing each of the tasks.

Moreover, we assume that the network is designed to provide a remote computing framework for mobile devices to use. The remote computing infrastructure can provide edge or cloud computing services. To take advantage of that infrastructure, the mobile device dynamically and opportunistically offloads some of its queued tasks to the remote server. The offloading mechanism is dependent on the choice of the user device. The remote server processes the tasks according to the needs and sends them back to the mobile device. If the output of that task is sent back to the user before the user device is supposed to start the processing of that specific task, then it is regarded as a completed task and immediately it is erased from the process queue. Otherwise, the task will be performed on the user's device. We consider the Target time (T) as the time required to process the desired service.

We are considering NDN for this kind of scenario motivating from. It is assumed that, when a user performs some computing requests, it is common that there is a relation among the requests from the user. This approach is particularly helpful in the VANET scenario in hazardous conditions. For example, on a highway, there is a disaster that happened (climate condition or accident) and all the involved vehicles are reacting to that situation. They are eventually reacting to that situation with the same kind of data. But the problem

is, there is a limited software resource to process these data. In this case, NDN can potentially come in benefits because of the fundamental characteristic of NDN called Interest Aggregation and Caching. When there are more and more similar requests, the gain from NDN increases accordingly. In this research, we are motivated to demonstrate that, NDN provides a strong impact on performance for this kind of scenario, where there are more and more similar requests. We also considered the mobility of the users against their static condition as well as available resources at the edge servers

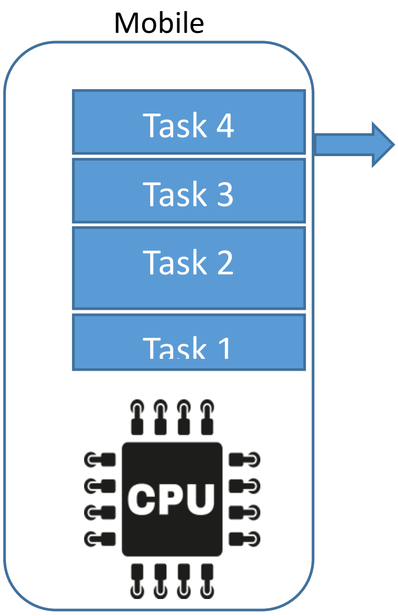

Figure 5. Queued task in the user device

In arising dynamic edge compute applications, delay tolerance is required even notwithstanding changing network structures, differing remote connections, and network resources in the presence of portability. Also, time-delicate analysis involves building the correct facility dynamically with the correct information continuously. In such circumstances, it is conceivable that the wellspring of the information or the destination may not be known beforehand. Further, traditional IP network systems don't have any attention to the network layer to misuse similitudes in the information or demands of the computation. Edge applications and computing mechanisms have a deep relationship with each other. NDN provides a way to support this relationship that leads to finding a solution in a more productive way

This paper [35] examined the plan of Pervasive Edge Computing (PEC) which is a democratized edge computing structure. It empowers end-client gadgets (e.g., cell phones, IoT gadgets, and vehicles) to powerfully partake in an enormous scope of the computing system. The democratized edge includes the real-time piece of services utilizing accessible edge assets like information, software, and computing equipment from different partners. They examined how the novel Named Data Networking design can encourage service distribution, revelation, and movement.

Named Function Networking known as NFN, utilizes the procedure of Information-Centric Networking to find the magnitude of computation to be done and information to be handled [36]. NFN empowers a Remote Procedure Call (RPC) and incorporates strategies for getting the outcome that is residing inside the network somewhere. RICE [37] further expanded NFN usefulness with consumer validation and approval and obliging non-trivial calculations. RPC functionalities, with the emphasis on edge computing, are stretched out in NFaaS [38]. In the event that the first compute worker that gets the solicitation needs more resources to execute the assignment, it advances the solicitation to an adjoining node. Such a methodology gives pioneering achievement of the compute task however doesn't ensure if the best server is determined or the time imperative is fulfilled.

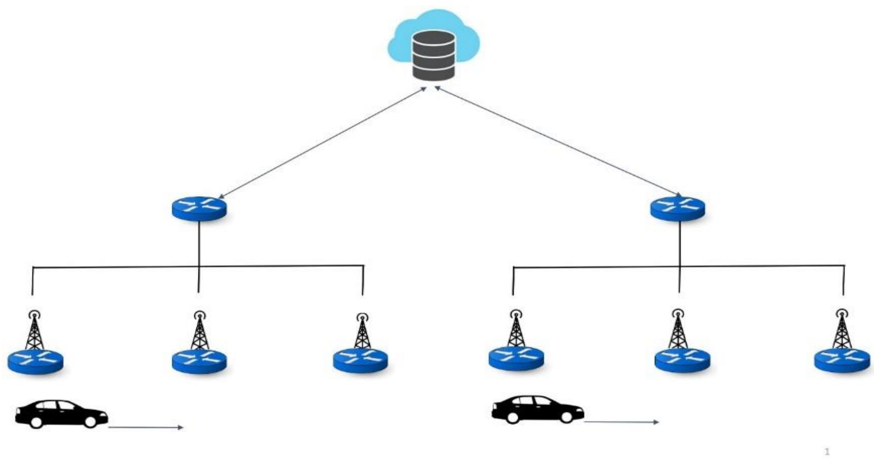

Figure 6. Cloud network deployment

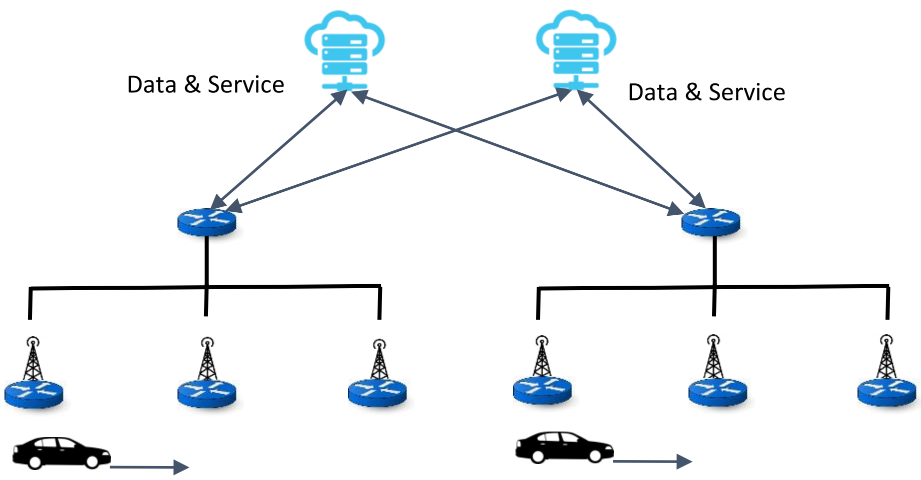

Figure 7. Edge network deployment (Data and Service in the Edge servers)

Named Data Networking involves two types of packets named Interest packet and Data packet. The Consumer provides the Interest packet and the Producer sends the Data. Whenever a Consumer needs a data packet, it sends the Interest packet in the network defining the name inside the Interest. The nodes inside the network which possess that specific data, send them to the requesting Consumer. In this case, all the neighboring nodes get the reply. But the Data packet that was got first, is delivered to the requester. The main advantage of this approach is it reduces the overall load of the involved network with efficacy. But there may appear a scenario, where there need multiple responses for a certain interest. For instance, a consumer can ask for multiple services for a specific kind of task.

In this research, a new type of Interest packet is introduced to enable multiple responses for a single Interest packet. The packet is named as "Discover" packet. This is a special keyword that is embedded into the Interest name and the keyword is determined in the network so that it cannot be included inside another type of interest. For instance, the start of a namespace can be assumed as "discover/service/" and it is solely responsible for leveraging multiple responses. This approach is useful to reduce the load of the involved network and also helpful in server selection. It will significantly decrease the number of Data packets that are deployed by the neighboring nodes. It depends on the timer and Data packet number that is forwarded downstream. When a Consumer deploys this kind of Interest packet in the network, it attaches the information about the lifetime of that interest as well as how many responses it wants back. This particular information is vital to determine when the Interest will be erased from the entry of the Pending Interest Table.

There is a mechanism to prevent the old entries in the PIT. When the "Discover" packet is traversing following a path, the NDN router makes a comparison between the parameters and the policies owned by them. In this case, if the mentioned parameters are compatible with the policies, those parameters will have the application on the actual entry of PIT. If that is not done, the NDN router will decrease the response number following it own mechanism. Here is an important factor is "InterestLifeTime". So, when this value goes to zero, there is no place for that entry inside the PIT. When Data is received inside the router, it looks for the matching name. Once found the data is delivered to the requesters and decrease the counter value. Like that, when the value of the counter is equals to zero, it means that, all the requests have been met. Then the entry in the PIT for that data is erased from there.

Data and Software in the Network

In the previous scenario, it is assumed that both the data and software will be located inside the edge nodes. Here it is considered that the requested information and the software required for processing that data will be inside the involved network. The data can be stored in one of the access points in the network and software can reside in a trusted repository like Application Storage.

In this case, the consumer will send an Interest packet specifying the name of the data as well as the type of service required to process the data inside the namespace. Whenever the node which has that specifically requested data listens to that request, it creates another Interest and sends it to the edge node where the necessary computation will be performed according to the need of the Consumer. For sending that type of request to the edge node, it will use a special reserved keyword inside the name prefix for example "/file/". We name it File Producer. Any access point in the network can act as a File Producer. Inside that network, the node which posses the application required to process and analyze the data according to the need, will produce another Interest and deploy it to the edge node. In this case, the Interest can also include a predetermined keyword as mentioned before for example, "/software/". We name this node as Software Producer. The edge node will take both the Interest from the File Producer and the Software Producer and performs necessary actions to analyze and compute the data. It will then send it ack to the Consumer.

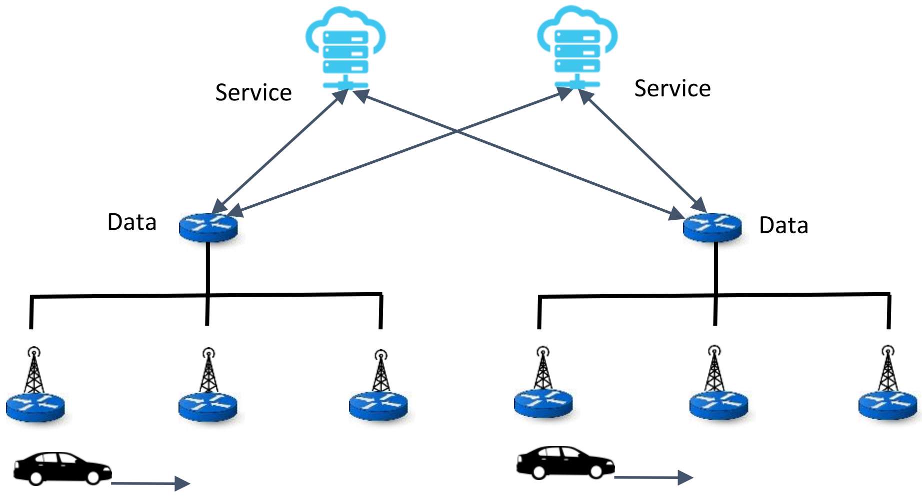

Figure 8.. Edge network deployment (Data in the access points and Service in the Edge servers)

3 Evaluation Methodology

This chapter reflects the functionality of the simulator used to perform our proposed approach. It also describes the main components of the used simulator and their characteristics. To simulate our proposed scenarios there were also some modifications needed in the simulator side which is also given here along with the parameters used in the simulation.

3.1 Simulator Used

To simulate the proposed approach using the behavior of Named Data Networking architecture a simulator was used named NDNSIM that is based on the Network Simulator 3 (NS-3). NS -3 uses IPv4 and IPv4 as network-layer protocols. NDNSIM introduced a new network layer protocol stack inspired by the architecture of Named Data Networking. This layer is capable of running on top of different link-layer protocols such as CSMA point-to-point and so on. It can also run on top of other conventional network protocols like IPv4 and IPv6 and other transport protocols like TCP and UDP. The main advantage of this simulator is that, it allows the user to simulate different types of heterogeneous network systems.

NS-3 is implemented using C++ and NDNSIM also follows the same path. NDN has different network-layer components named FIB, PIT, CS and network and application layers. NDNSIM is developed in such a way that, there are separate C++ classes for each NDN network layer component. The benefit of having separate classes for each component enables its user to modify or add different functionalities to NDNSIM with minimal effort and flexibility.

The core component of NDNSIM is NDN Forwarding Daemon (NFD). This component works as a forwarder in NDN architecture and implements all the abstraction layers for networking. The main objective of NFD is to make the Interest and Data move to the desired way. NFD consists of several inter-dependent modules such as ndn-cxx library, faces, tables, forwarding, management and Routing Information Base (RIB) management.

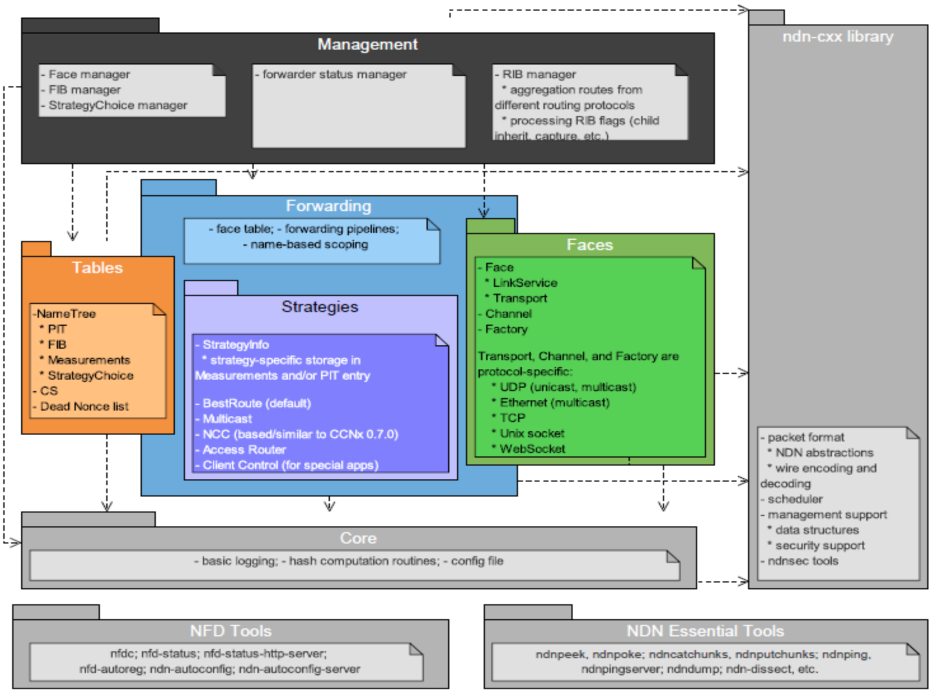

Figure 9 NFD Architecture

Packets show up at NFD by means of Faces. The face is speculation of the interface. It tends to be either an actual interface, in which NDN works straightforwardly over the Ethernet. It can also work as an overlay burrow meaning that NDN works as a surface above TCP as well as UDP. Furthermore, correspondence between NFD and a neighborhood application should be possible by means of a Unix socket. It is similar to a Face. LinkService and Transport make a Face. The LinkService offers undeniable types of assistance for the Face, as fragmentation and reassembly of data packets, error correction. Transport goes about as a covering over a fundamental network transmission protocol (TCP, UDP, and so on) and offers types of assistance same as link-layer counters. The approaching stream is examined by the Face through the API of the operating system, takes out network-layer bundles from link protocol bundles, and conveys these network-layer parcels such as Interests, Data and Nacks to the forwarding.

A network-layer bundle is prepared by forwarding pipelines. It includes a series of procedures that work on the parcel. The information plane of NFD is stateful. How NFD deals with a parcel relies not only upon the actual bundle yet additionally the forwarding state, which is kept inside the tables.

At the point when an Interest is got by the forwarder, an entry is made in the Pending Interest Table (PIT) for the approaching Interest, where every entry addresses a forthcoming Interest or an Interest that is satisfied recently. A query for coordinating with Data is done on the NDN router Content Store (CS). It is a cache of Data bundles residing inside the network. In the event that there is coordination with the Data bundle in CS, that Data bundle is gotten back to the entity that requested it; in any case, the Interest should be sent.

A forwarding procedure concludes the procedure to advance an Interest. NFD contains a mechanism for per-namespace system decisions. It chooses which procedure is answerable for Interest forwarding. For this purpose, the longest prefix match query is done on the Strategy Choice table. This table contains the configuration for all the strategies. The procedure is responsible to determine the destination of the interest. It also determines if there is any need to forward that interest. This method also retrieves information from Forwarding Information Base (FIB), which contains the routing data. It utilizes information that is strategy-specific. They are kept along with PIT entry.

When the methodology chooses to transfer an Interest to a determined Face, the Interest travels through a couple of more strides in forwarding pipelines. After that, it is advanced to the Face.

An incoming data is prepared in a different manner. Firstly, a query is performed to check in the PIT to find any entry that can be used to satisfy the Data bundle. All coordinated entries are then chosen for additional handling. If there is a situation that, this Data does not match any of the PIT entries, it is considered obsolete. So, it is dropped. In another case, the Data is kept inside the Content Store. All the PIT entries for the same data have associated forwarding techniques and a notification is sent to them. By means of this notification, the strategy can notice the reachability and efficacy of the forwarding path. At last, the Data is shipped off to all consumers. The way toward sending Data through a Face is like sending an Interest.

At the point when the forwarder gets a Nack, the steps differ relying on the forwarding technique being used.

3.2 Simulator Modification

To simulate the proposed scenario, several functionalities have been made in the ndnSIM simulator. There are four different entities involved in the implementation of the simulation. They are Consumer, File Producer, Software Producer and Compute node (acting as the edge server).

The simulation starts from the Consumer(s). To act as Consumer, a node has been assigned. The route for the Interest prefix has been added and also Interest lifetime is defined in the source code. NDN network stack is put inside the Consumer node. Then the Consumer starts sending the Interest for its desired data defining the name of the data inside the namespace. The functionality is implemented using the public member function *sendInterest()* of the class *nfd::Face* defined in ndnSIM. When the Consumer receives its requesting Data from the Producer(s), *OnData()* is called which is another public member function of class *ns3::ndn::App*.

To act as a File Producer, a node has been dedicated to receiving the Interest sent by the Consumer. As implemented previously, the NDN stack is also installed inside that node to perform according to the NDN architecture. When a new Interest packet is received by this node *OnInterest()* method is invoked from *ns3::ndn::App* class. Inside this method, Data is prepared for sending to the Compute node by setting up the freshness of the content. The Data is also digitally signed before sending to the destination node using a method from class *ndn::StackHelper*.

Another entity in the simulation is the Software Producer. The implementation of this node is almost similar to the File producer node except a slight difference is there. Unlike the File Producer, upon receiving the Interest from the Consumer, it gets the required software name from the name prefix and prepares for sending the software to the Compute node. Like File Producer, *OnInterest()* method is invoked from *ns3::ndn::App* class.

Compute node gets the response from both the File Producer and Software Producer. When it gets the required File and Software, it does the necessary computation and prepares the Data destined for the Consumer. This reply from the Compute node invokes the *OnData()* on the Consumer node and the Consumer gets its desired Data packet.

3.3 Simulation Setup

The goal of this research is to evaluate the performance of an augmented computing system using the Named Data Networking as a networking protocol. To reach this goal we considered a highway scenario where the vehicles are moving on the road and communicating with the edge and cloud servers to offload their computing task according to their needs. The scenario is thought of considering two different traffic models. One is the light traffic model and the other is the congested traffic model. In the case of the light traffic model, it is assumed that there are 10 vehicles moving per one kilometer and their mean velocity is 40 meters per second. On the contrary, in the case of a congested traffic model, there are 40 vehicles moving along the road and their average velocity is 10 meters per second. Wireless small cell deployment is considered as the connectivity model for the above-described traffic model and traffic scenario. To simulate the required scenario for the experiment, we have considered the ndnSIM, a network simulator using Named Data Networking.

With our thought of traffic model, two different kinds of deployment are tested. They are cloud-centric deployment and edge-centric deployment. In the case of a traditional network system, there is a handover procedure involved that provides seamless and uninterrupted connectivity among the base stations and mobile users. Our experiment depends on the unlicensed wireless network system and no handover procedure is considered here. This is a quite challenging situation in the case of latency-sensitive applications because it involves the potential loss of transmitted packets.

The assumption in this experiment involves the sending of similar requests by the mobile users to the server. The meaning is, the mobile users will request the same processing procedure for the same data over and over again. The motivation behind this assumption is, there may be a scenario in the highway (for example hazard, roadblock) that will affect the vehicles on the road and all the vehicles react to that occurrence using some traditional processing or software.

The experiment considers two different types of applications. They are, “cbr” and “zipf”. For application type “cbr” users involved in the simulator setup requests the same type of software again and again to process their demand. In the case of application type “zipf” the users are likely to request different types of service or software to meet their demands. In this experiment scenario, the number of software is limited to 100. But all of the requested services or software are not of the same popularity. Some of them are more popular and more likely to be requested by the users and some are less likely to be wanted by the users. The distribution of the popularity of the software is calculated by the Zipf-Mandelbrot law, which is a discrete probability distribution determined by the rank of involved data. According to the distribution law, vehicles can request different types of services they face any type of hazardous situation and they need to react to that occurrence. But there is always a possibility that there will be a lot of overlapping software requests because some services will be more popular than other software. We have considered the freshness of data one second in both of the applications. The meaning of content freshness is, when the processing of the data is finished, after one second it will be deleted from the data when the cache is enabled.

For the experiment, we tried to evaluate the success rate of the successfully accomplished computing tasks within the desired amount of time. Reliability and latency are both taken into account in the performance metric to gain an overall idea about how the system performs. The parameters used in the simulation are stored in Table number 1.

Parameter	**Value**
Number of runs for each combination of parameters	100
User number (light traffic), *N / kilometer*	10
User number (congested traffic), *N / kilometer*	40
User velocity (light traffic), *V meter/second*	40
User velocity (congested traffic)	10
Types of NDN applications	zipf,cbr
Types of IP applications	UDP
Size of cache, computing result numbers	0, 20
Type of cache	LRU
Result of the computation in freshness, *millisecond*	1000
Distribution of compute time, *millisecond*	U [10;100]
Time of decision for server selection procedure, *millisecond*	6
Average RTT between a cloud and a user, *millisecond*	100
Average RTT between edge servers and user, *millisecond*	[3;6]

Table 1 Parameters for assessing the simulation

4 Numerical Result

In this chapter, we analyzed the results of the simulation with our intended scenarios. We divided our whole simulation into two major parts wherein one part “Data and software both are in servers” and in the second part “Data and software both are in the network”. Then for each of the parts, we have got our simulation results and analyzed them to get insights into the effect of interest aggregation and caching mechanism of NDN.

4.1 Data and Software in the Servers

The effect of interest aggregation and caching are described below when data and software are in the edge server.

Effect of Interest Aggregation

Three main important elements are responsible for Content-Centric Network (CCN). They are Forwarding Information Base, Content Store and Pending Interest Table. The router is responsible for maintaining the FIB, which stores all the incoming interests with their prefixes so that they can be sent to their desired destination. The router also stores the contents in their local repository known as CS in an opportunistic manner. Maintenance of PIT is also a responsibility of the router to suppress unwanted and unnecessary Interests. When an Interest reaches the router it first checks if it can be satisfied by the content stored in the local cache. If it is not able to satisfy from the cache, it creates an entry in the PIT, provided that there is no similar entry in the PIT with the same name and forwards it. But when there is a similar entry with a similar name in PIT then the Interest is aggregated and not forwarded. This concept is called Interest Aggregation. The main idea of Interest aggregation is to reduce the bandwidth consumption and latency between the involved parties in the communication [39].

In the first phase of our experiment, we have considered the traditional IP-based network solution. We assumed that the end-user will send their requests to the cloud for performing some computational task over some data and they are using UDP as their communication protocol. UDP is a connectionless message-oriented protocol that consists of a minimum level of communication protocol mechanism with no guarantee of delivery. But it is a common choice for the latency-sensitive application.

With the IP-based communication system, an NDN-based implementation is considered in the experiment. In this setup, the routers and the access points of the network are based on Named Data Networking, but the implemented cloud server is using Internet Protocol.

So the router serves here as the gateway between the two different network protocols. In this kind of heterogeneous network, we disabled the network caching. This will allow us to visualize the intended effect of network aggregation. The result is reflected in Figure number

In the experiment, it is visible that, the homogeneous network model based on IP shows a 50% success rate in accomplishing the requested processing within the timespan of the first 300 milliseconds. This observation was the same in light traffic conditions as well as in congested traffic conditions. The outcome reflects that the impact of the high movement in terms of packet loss in light traffic conditions is the same also in congested traffic conditions where the interference is high, and the load is greater.

But in the case of heterogeneous NDN-based network outperforms the homogeneous network both in light and congested traffic conditions. The main responsible feature, in this case, is the Interest Aggregation. It allows to share computing resources greatly and in a better way and this effect is similar both in “cbr” and “zipf” application scenario. Not only the Interest Aggregation is responsible for this improved behavior, but also less overhead in terms of resolving IP addresses. If we dive into the result more, we can see that the improvement in congested traffic conditions surpasses the improvement in the light traffic condition. The reason is, in case of congested traffic conditions, there are more overlapping requests of the services by the mobile users.

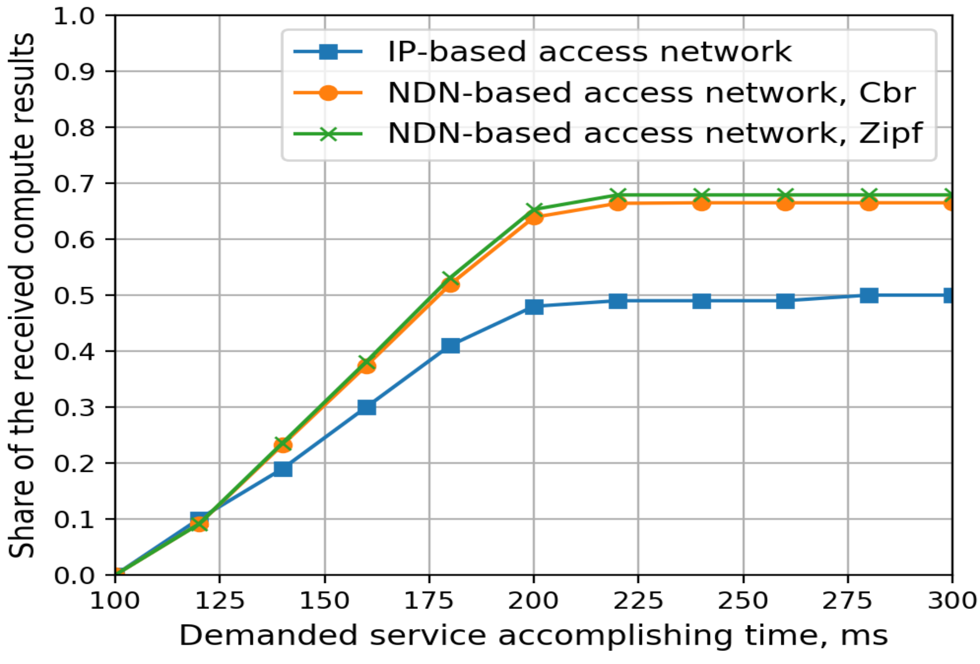

Figure 10. Effect of Interest Aggregation (Free road)

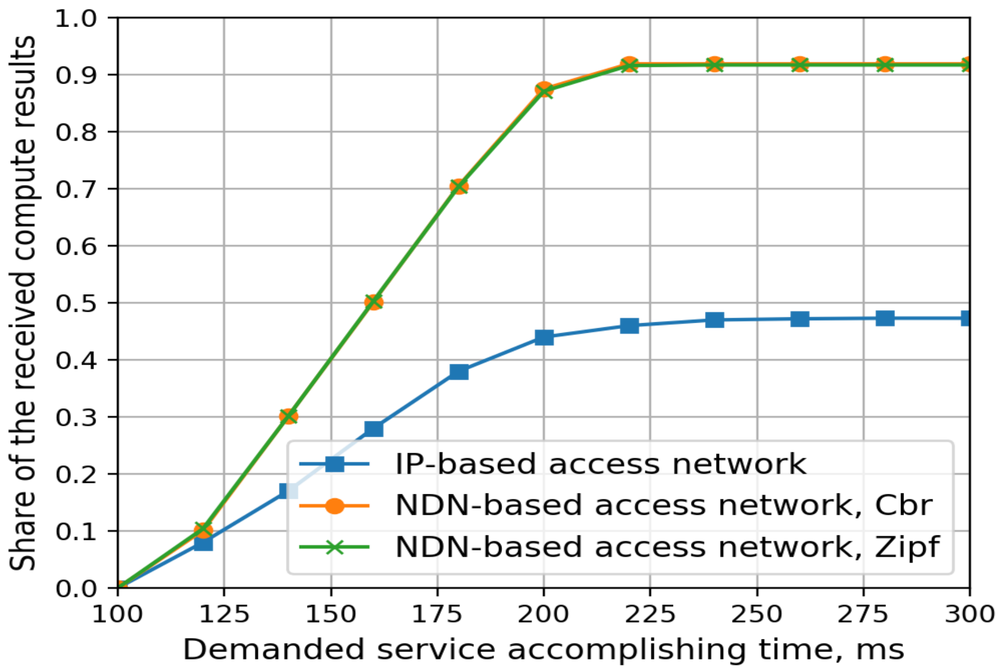

Figure 11. Effect of Interest Aggregation (Congested road)

Effect of Caching

As described in Section we know that Content Store is a primary element of Named Data Networking system. It is a temporary storage. In NDN, every data is meaningful, and it is not considered here from where the data is originated. Routers store the data in its local caching system to satisfy the future interest. This caching mechanism enables the network to process similar interests faster.

In the next set of our experiments, we came to know the effect of this caching mechanism in NDN. Caching was allowed in both the routers as well as in all the intermediate access points to satisfy future interests with the same name prefixes. The cache was configured in such a manner that there will be a maximum of 20 contents stored in the local storage and the lifetime of them will be no more than 100 milliseconds as specified in the table. We have considered our previous designed scenario in Figure and make a comparison in terms of cache enabled and cache disabled in Figure.

The result shows a remarkable improvement in the case of successfully completed service requests in the aspect of "cbr" traffic deployment. In this kind of deployment, the mobile users demand the same computing service in the time span of 1000 milliseconds. The content freshness is set to 1000 milliseconds. So, the Interests from the other users for the same service will be satisfied from the data stored in the Content Store, rather than forwarding the Interests to the Producer. Because the result for the same Interests was stored when the service was satisfied the first time.

But when using the "zipf" application, mobile users reacted to the same occurrence but used different services for processing their data. So, there were a limited number of requests for the same service as there were fewer overlapping of the requests. That made the gain of caching less. From the result, it can be derived that, in the case of latency-sensitive applications, the gain of caching is very limited, and it applies to some specific use cases.

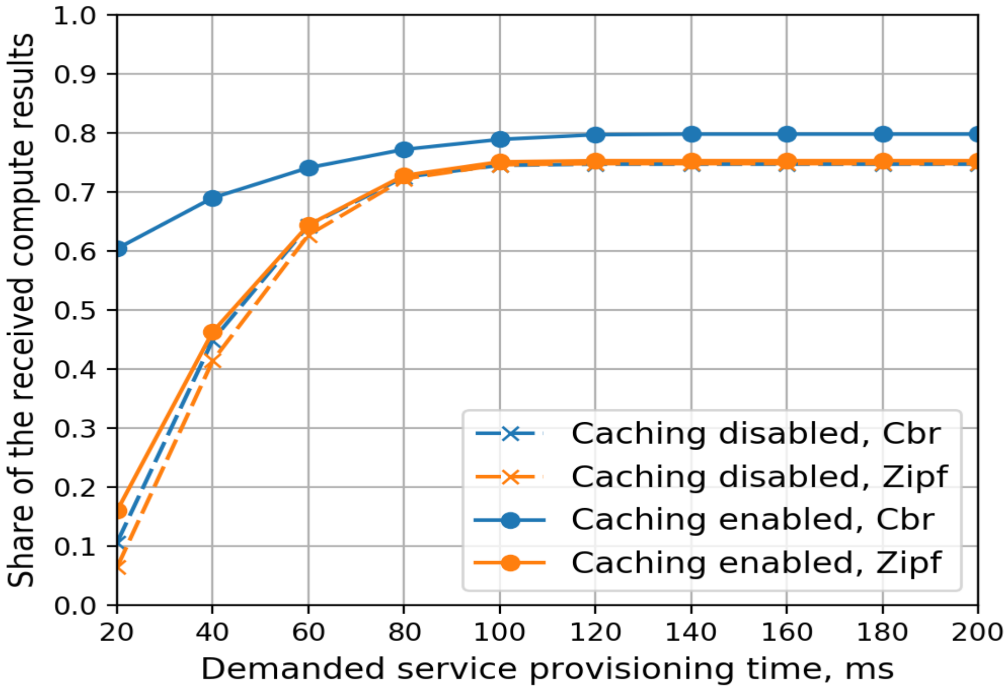

Figure 12. Effect of Caching (Free road)

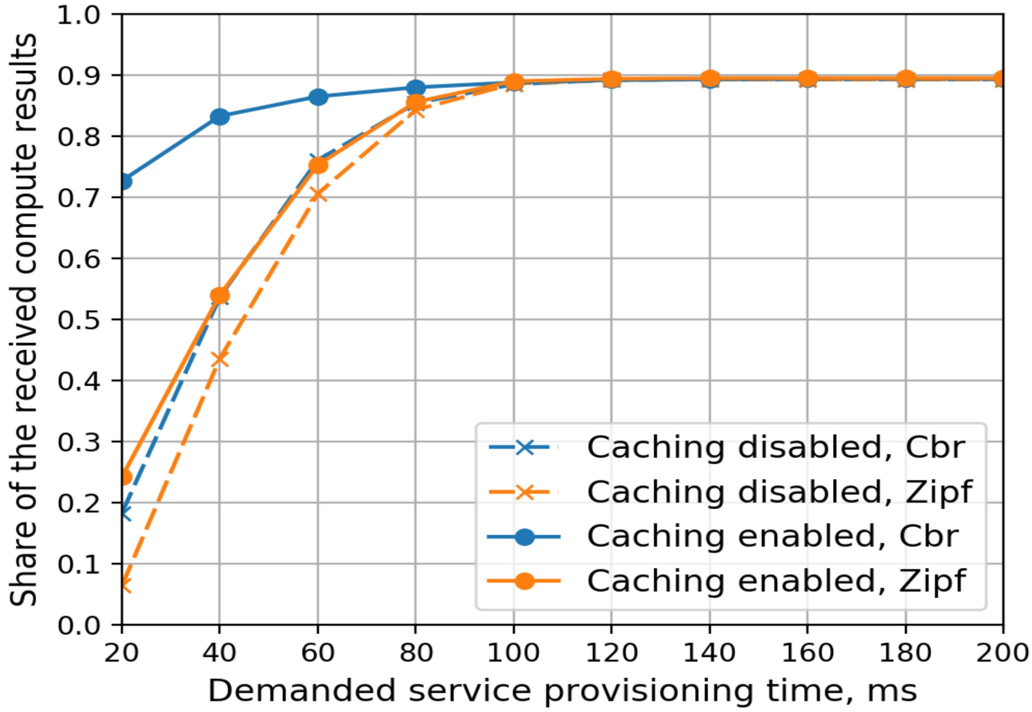

Figure 13 Effect of Caching (Congested road)

4.2 Data and Software in the Network

Our previous part of the experiment includes a scenario, where the mobile users will offload some of their computational tasks to the edge server and it will perform the computation and send back the result to the users. In this case, the users will send the Interest and the edge server will only perform the necessary calculation using the intended services with the data in the server.

Here we considered a highway scenario, where the vehicles are moving along the highway and some of them are needed with some specific information, for example, traffic condition of that road. The difference between our previous scenario is that the data is now stored inside the network. The storage can be the access points or some external trusted repositories. So, whenever a user needs information, the data will be collected from the external repository and the computation over the data will be performed in the edge server. We have considered two edge servers (EdgeV1 and EdgeV2) where the EdgeV1 gets the data inside the server and EdgeV2 gets the data from the network. The experiment is performed using both free highways and congested highways. Also, two different application types “cbr” and “zipf” are considered in this experiment. The caching was disabled here.

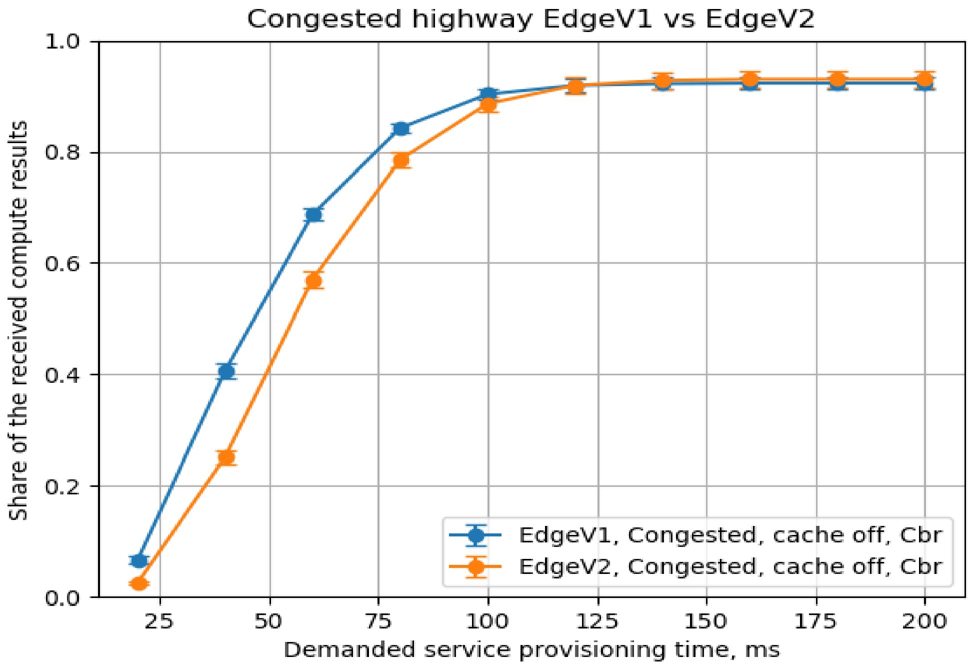

Figure 14. Data and Software in the road in congested road (cbr)

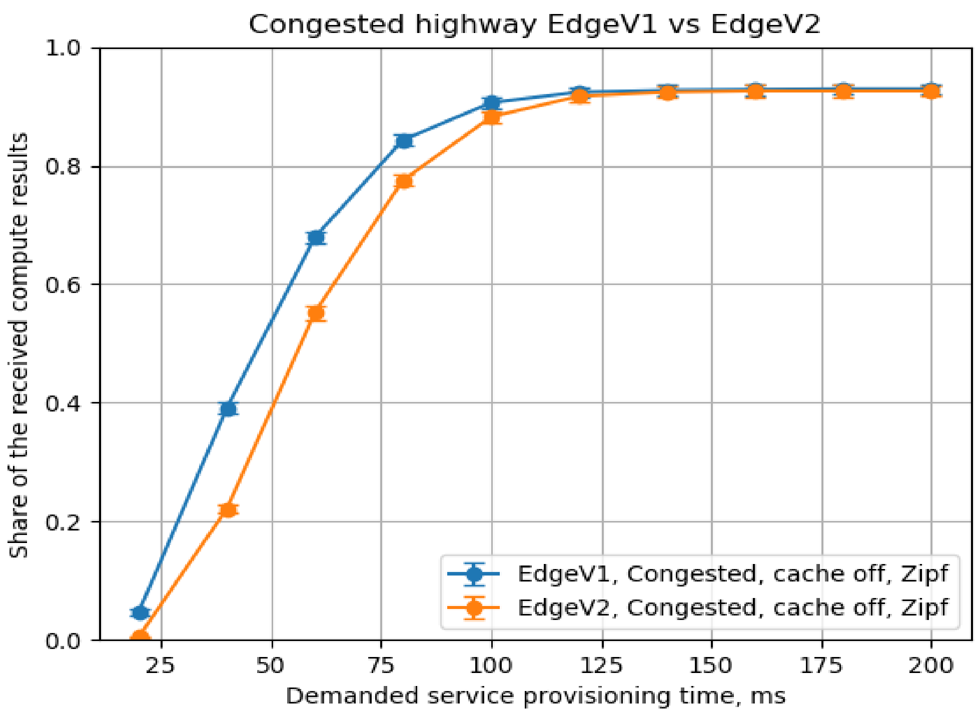

Figure 15. Data and Software in the road in free road (cbr)

Figure 14 and Fig 15 depict the performance in a congested highway scenario. It is clearly visible that, EdgeV1 outperforms EdgeV2 in terms of providing the desired computed results. The reason behind this result is, there needs an extra amount of time to look for the desired data in the network for EdgeV2 which is not needed in the case of EdgeV1. EdgeV1 gets the data from the requesting user and performs the necessary service on that data which produces a better result for EdgeV1. Both “cbr” and “zipf” application types show similar types of behavior in congested roads. So, it is visible that, the performance is application-independent.

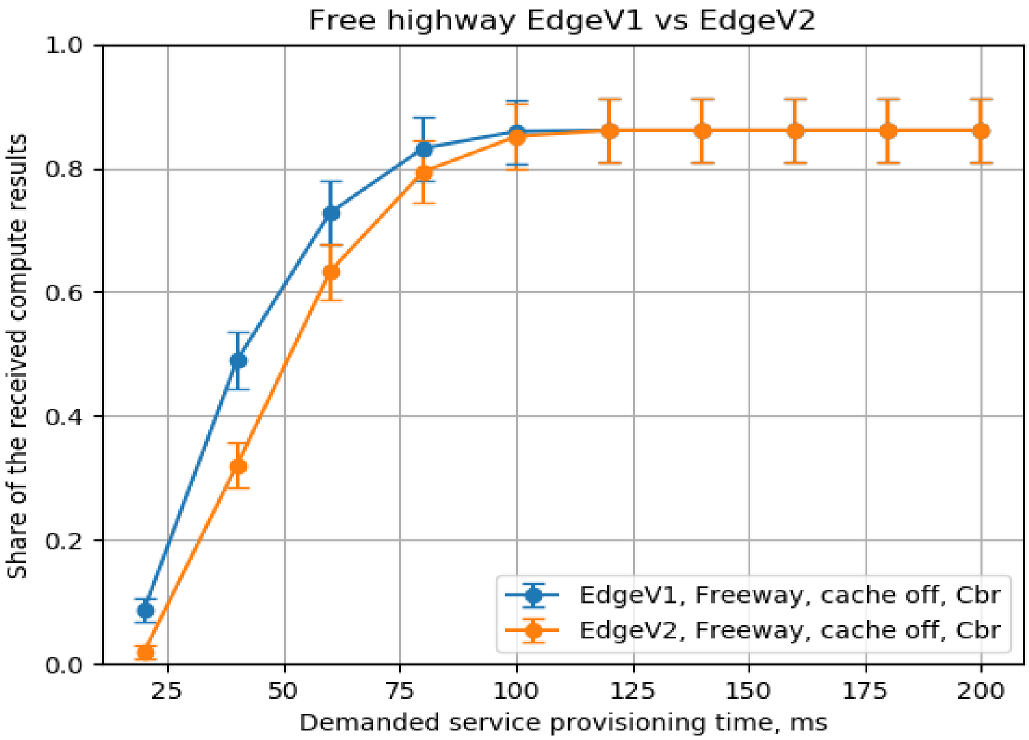

Figure 16. Data and Software in the road in free road (zipf)

A similar type of result is shown in Figure 16 and Fig 17. They show the performance of EdgeV1 and EdgeV2 on a free highway. Here also EdgeV1 demonstrates better performance than EdgeV2. The only difference between the congested highway and free highway is that we get a set of values instead of actual values in different stages. So they are provided as confidence intervals in both the figures. Here these two figures also provided application-independent results for both the edge servers.

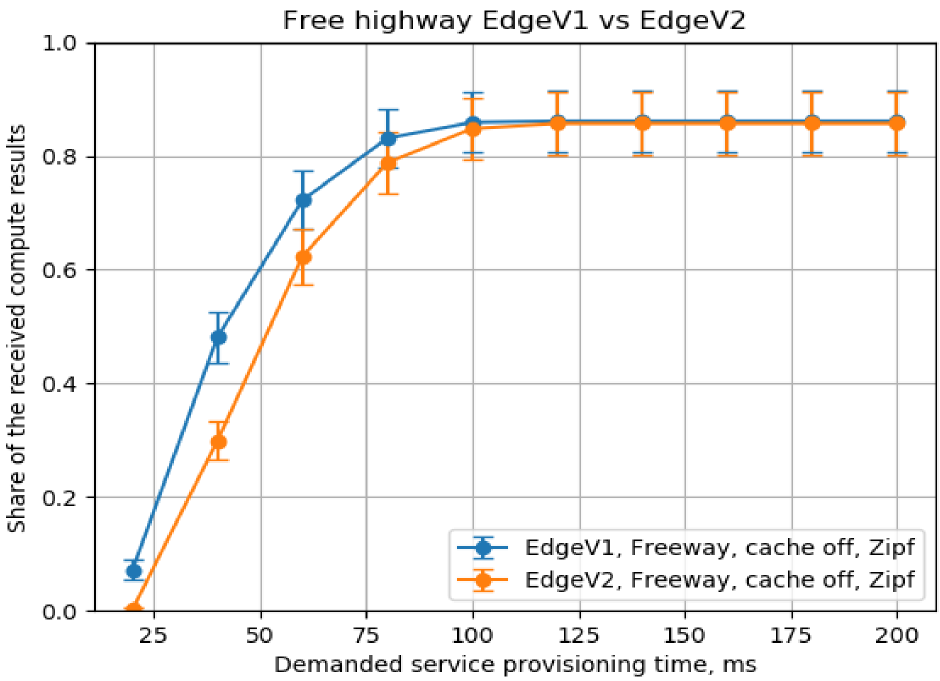

Figure 17. Data and Software in the road in free road (zipf)

5 Conclusion

The main goal of this research work was to show that, the distinct characteristics of NDN architecture can show performance improvements over the traditional TCP/IP network in a dynamic augmented computing environment. The advantages of the NDN network were demonstrated in a mobile wireless network. The considered scenarios in the simulation did not actually show the benefits of all the characteristics of NDN. It was specifically designed to show the impacts of the two most important features of NDN known as Interest Aggregation and Caching. The scenarios are also considered to show the availability of the data location and their impacts. We could also gather the answers to our research questions:

1. *What are the issues with traditional IP network?*

   The conventional IP network mechanism is solely dependent on the source and destination of the desired data packet. It does not care about what it is sending, but where it is sending. These characteristics of this network architecture are still widely accepted and serve their purpose in most cases. But in the case of the mobile network where the endpoints are moving and source and destination addresses are not fixed, it faces issues and causes loss of packets. The same communication security approach is applied in sending data through different communication channels are used to send different types of data. This approach is often not enough to fulfill the requirements.

2. *How NDN can overcome the limitations of the traditional IP network in certain cases?*

   We have simulated our intended scenarios to address some issues of IP network in mobile network architecture. From our results, we found that two main characteristics of NDN were responsible for outperforming the IP network. They are Interest Aggregation and Caching. In the case of Interest aggregation, a similar entry with a similar name was stored in the PIT and not forwarded. It helped to reduce latency and also bandwidth consumption. Caching property of NDN also helped to gain better results than the IP network. In the NDN router and intermediate access points, data were saved temporarily to satisfy future interests. That also provided faster communication results than IP.

According to the results in the simulation, we have clearly seen the advantages of Interest Aggregation and Caching over the conventional network. This result does not necessarily prove to be useful in all the application scenarios. This improvement is very much application-specific and it can be useful where the requested computing services overlap again and again.

In a typical mobile wireless network scenario, a handover mechanism is involved between the base station and mobile devices. It is not considered in our scenarios. Handover can cause loss of packets during the communication and it can affect the gain that we have demonstrated in our research. There remains a scope of research to see the gain of NDN. Also, the gain of Interest Aggregation and Caching can be evaluated in other application scenarios that will open a door to the future implementation of NDN in the mobile wireless network.

In summary, we can deduce that NDN can successfully show advantages in the dynamic edge computing environment where the resource availability is dynamic and changing with time. Mobile users can take advantage to perform their computational tasks taking the help of an edge computing system for faster communication.

REFERENCES

[1] P. Simoens, D. Griffin, E. Maini, T.K. Phan, M. Rio, L. Vermoesen, F. Vandeputte, F. Schamel and D. Burstzynowski, '"Service-Centric Networking for Distributed Heterogeneous Clouds," *COM-M*, vol. 55, no. 7, Jul, pp. 208-215.

[2] M. Sifalakis, B. Kohler, C. Scherb and C. Tschudin, '"An information centric network for computing the distribution of computations," *Proceedings of the 1st ACM Conference on information-centric networking*, pp. 137-146.

[3] D. Saxena, V. Raychoudhury, N. Suri, C. Becker and J. Cao, '"Named Data Networking: A survey," *Computer science review*, vol. 19, Feb, pp. 15-55.

[4] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang and L. Zhang, '"A Brief Introduction to Named Data Networking," *MILCOM*, pp. 1-6.

[5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, Kc Claffy Caida Uc, Ucla, S. Diego, P. Crowley, C. Papadopoulos, L. Wang and B. Zhang, '"Named Data Networking,".

[6] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs and R.L. Braynard, '"Networking Named Content,".

[7] K. Ahed, M. Benamar and R. El Ouazzani, '"New classification of Named Data Netwoking Applications," *MoWNet*, pp. 6-12.

[8] J. Haillot, F. Guidec, S. Corlay and J. Turbert, '"Disruption-Tolerant Content-Driven Information Dissemination in Partially Connected Military Tactical Radio Networks," *Proceedings of the 28th IEEE Military Communication Conference (MILCOM'2009)*, pp. 2326-2332.

[9] M. Amadeo, C. Campolo, A. Iera and A. Molinaro, '"Named data networking for IoT: An architectural perspective," *EuCNC*, pp. 1-5.

[10] Wentao Shang, Qiuhan Ding, A. Marianantoni, J. Burke and Lixia Zhang, '"Securing building management systems using named data networking," *NET-M*, vol. 28, no. 3, May, pp. 50-56.

[11] S.H. Bouk, S.H. Ahmed, D. Kim and H. Song, '"Named-Data-Networking-Based ITS for Smart Cities," *COM-M*, vol. 55, no. 1, Jan, pp. 105-111.

[12] S.H. Bouk, S.H. Ahmed and D. Kim, '"Hierarchical and hash based naming with Compact Trie name management scheme for Vehicular Content Centric Networks," *Computer communications*, vol. 71, Nov 01, pp. 73-83.

[13] G. Piro, I. Cianci, L.A. Grieco, G. Boggia and P. Camarda, '"Information centric services in Smart Cities," *The Journal of systems and software*, vol. 88, Feb, pp. 169-188.

[14] S.H. Ahmed and D. Kim, '"Named data networking-based smart home," *ICT express*, vol. 2, no. 3, Sep, pp. 130-134.

[15] M. Amadeo, C. Campolo, A. Iera and A. Molinaro, '"Information Centric Networking in IoT scenarios: The case of a smart home," *ICC*, pp. 648-653.

[16] D. Massey, C. Papadopoulos, L. Wang, B. Zhang and L. Zhang, '"Teaching Network Architecture through Case Studies,".

[17] Z. Zhu, S. Wang, X. Yang, V. Jacobson and L. Zhang, '"ACT: Audio Conference Tool Over Named Data Networking,".

[18] P. Gusev and J. Burke, '"NDN-RTC," *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, -09-30.

[19] J. Lindblöm, M. Huang, J. Burke and L. Zhang, '"FileSync/NDN: Peer-to-Peer File Sync over Named Data Networking,".

[20] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain and A. Ksentini, '"Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges," *COMST*, vol. 22, no. 1, pp. 320-351.

[21] J.M. Duarte, T. Braun and L.A. Villas, '"MobiVNDN: A distributed framework to support mobility in vehicular named-data networking," *Ad hoc networks*, vol. 82, Jan, pp. 77-90.

[22] X. Wang and Y. Li, '"Vehicular Named Data Networking Framework," *TITS*, vol. 21, no. 11, Nov, pp. 4705-4714.

[23] S.H. Ahmed, S.H. Bouk, D. Kim, D.B. Rawat and H. Song, '"Named Data Networking for Software Defined Vehicular Networks," *COM-M*, vol. 55, no. 8, Aug, pp. 60-66.

[24] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa and Lixia Zhang, '"VANET via Named Data Networking," *INFCOMW*, pp. 410-415.

[25] Meng Kuai, Xiaoyan Hong and Qiangyuan Yu, '"Density-Aware Delay-Tolerant Interest Forwarding in Vehicular Named Data Networking," *VTCFall*, pp. 1-5.

[26] Yu-Ting Yu, Yuanjie Li, Xingyu Ma, Wentao Shang, M.Y. Sanadidi and M. Gerla, '"Scalable opportunistic VANET Content Routing with encounter information," *ICNP*, pp. 1-6.

[27] Jiangzhe Wang, R. Wakikawa and Lixia Zhang, '"DMND: Collecting data from mobiles using Named Data," *VNC*, pp. 49-56.

[28] A. Attam and I. Moiseenko, '"NDNBlue: NDN over Bluetooth,".

[29] M. Meisel, V. Pappas and L. Zhang, '"Ad hoc networking via named data," *Proceedings of the fifth ACM international workshop on mobility in the evolving internet architecture*, pp. 3-8.

[30] M.M. Ucla, V. Pappas, L. Zhang and U. , '"Listen First, Broadcast Later: Topology-Agnostic Forwarding under High Dynamics,".

[31] B. Varghese, Nan Wang, S. Barbhuiya, P. Kilpatrick and D.S. Nikolopoulos, '"Challenges and Opportunities in Edge Computing," *SMARTCLOUD*, pp. 20-26.

[32] C. Meurisch, A. Seeliger, B. Schmidt, I. Schweizer, F. Kaup and M. Mühlhäuser, '"Upgrading Wireless Home Routers for Enabling Large-Scale Deployment of Cloudlets,", pp. 12-29.

[33] Dengpan Yin and T. Kosar, '"A data-aware workflow scheduling algorithm for heterogeneous distributed systems," *HPCSim*, pp. 114-120.

[34] M.T. Beck and M. Maier, '"Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms,".

[35] R. Tourani, S. Srikanteswara, S. Misra, R. Chow, L. Yang, X. Liu and Y. Zhang, '"Democratizing the Edge: A Pervasive Edge Computing Framework,", Jul 01,.

[36] C. Tschudin and M. Sifalakis, '"Named functions and cached computations," *CCNC*, pp. 851-857.

[37] M. Król, K. Habak, D. Oran, D. Kutscher and I. Psaras, '"RICE," *Proceedings of the 5th ACM Conference on information-centric networking*, pp. 1-11.

[38] M. Król and I. Psaras, '"NFaaS," *Proceedings of the 4th ACM Conference on information-centric networking*, pp. 134-144.

[39] A. Dabirmoghaddam, M. Dehghan and J.J. Garcia-Luna-Aceves, '"Characterizing Interest aggregation in content-centric networks," *IFIPNetworking*, pp. 449-457.