

**YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES**

MASTER THESIS

**RELIABILITY OF ROUTING AND CACHING IN NAMED DATA
NETWORKING**

Yakubu Yunusa Sulaiman

Thesis Advisor: Asst. Prof. Dr. Tuncay ERCAN

Department of Computer Engineering

Presentation Date:

**Bornova-İZMİR
2014**

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

Assist. Prof. Dr. Tuncay Ercan

(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

ABSTRACT

RELIABILITY OF ROUTING AND CACHING IN NAMED DATA NETWORKING

Yakubu Yunusa Sulaiman

M.Sc. in Computer Engineering

Supervisor: Asst. Prof. Dr. Tuncay ERCAN

June 2014

The named data networking (NDN) architecture was proposed to replace current host-based routing with name based routing together with in-network caching for scalable content dissemination over the network topology. NDN routing broadcast name prefixes not IP address with adaptive forwarding that supports multipath forwarding and intelligent content distribution due to built-in caching. NDN used an extended version of today's routing scheme for early implementation, but currently has its routing protocols based on name and some are proposed. The routing and caching of data in NDN is normally carried out within routers' control plane, containing three tables; Content Store, Pending Interest Table (PIT) and Forwarding Information Base and these tables need serious maintenance for both reliable performance in routing and data caching. However, the cache located in NDN routers requires some critical attention from replacement policy. Therefore, this research aimed to bring some strategies to minimize PIT size and to measure cache performance for different replacement policies.

ACKNOWLEDGMENT

It is my mandate to first appreciate and thank the almighty Allah for the strength in successful completion of this research work. Secondly and strongly appreciating the support, guidance and advice of my supervisor Assist Prof. Dr. Tuncay Ercan toward the researching objectives of this thesis all the time. My gratitude also goes to the Head of Department- Computer Engineering Department; Prof. Ahmet Koltuksuz for support, assistance and advice since from the beginning of my Masters Program in Yasar University. I would also like to acknowledge the support of other staffs in the department for their support in one way or other and the entire yasar academic environment.

I would like to thank my family for their support, prayer since the time of my birth up to date more especially my Mother, Elders and my wife for adaptive patience all the time. Thanks to friends also for advice and assistance during the course of this unforgettable moment.

TEXT OF OATH

I declare and honestly confirm that my study, titled Reliability of Routing and Caching in Named Data Networking and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions, that all sources from which I have benefited are listed in the bibliography, and that I have benefited from these sources by means of making references.

INDEX OF FIGURES

FIGURES	PAGES
Popularity use of internet.....	5
Content-centric networking in today's internet.....	13
Infrastructural transition from IP network to NDN.....	16
NDN packets.....	20
Interest request forwarding.....	25
NDN in-network caching.....	30
Name component prefix trie.....	40
Name component prefixes-integer value.....	41
Slot allocation of PIT entries.....	42
Deleting item from PIT table.....	43

INDEX OF TABLES

Tables	Pages
TCP/IP protocol stack.....	7
HTTP Methods.....	11
Pending interest table entries.....	22
Forwarding information base.....	23
PIT name prefixes-integer value.....	44
PIT hashing key.....	45

Table of Contents

CHAPTER ONE	3
1.1 INTRODUCTION:	3
1.2 Current Internet Architecture	6
1.2.1 IP Routing:	8
1.2.2 Caching and Content Distribution Support by IP Network;	9
1.3 Today’s Internet Architecture Challenges;	13
1.4 Future Internet Architecture;.....	14
1.4.1 Named Data Networking (NDN) Architecture;	15
1.4.2 Expectations on FIA-NDN;	16
1.4.3 Other Related FIA:.....	17
CHAPTER TWO	18
2.1 Named Data Networking (NDN):	18
2.2 NDN Operation:.....	19
2.2.1 NDN Components.....	21
2.2.2 Forwarding Process:.....	23
2.3 Advantages of NDN over IP Network;.....	26
2.4 Naming in NDN:.....	26
2.5 NDN Routing:	27
2.5.1 Open Shortest Path First (OSPFN);	28
2.5.2 Named-Data Link State Routing Protocol;	28
2.6 Caching in NDN;	29
2.6 Cache policy:	30
2.6.1 Cache replacement policy;	31
2.6.2 Present NDN Cache Attacks and Countermeasures:	32
CHAPTER THREE	36
3.1 NDN Routing and Caching Challenges:	36
3.1.1 Memory Performance:	36
3.1.2 Design Principle:.....	37

3.2 Hash Table:	38
3.2.1 PIT Hashing:	39
3.3 Name Prefix component Aggregation:	40
3.4.1 Numerology:	42
3.4 PIT Design Model:.....	43
3.4.1 One Way Hash Function:	45
CHAPTER FOUR	46
4.1 Experiment Evaluation:	46
4.2 Implementation:	46
4.3 Slot Allocation:	47
4.6 Analysis of the PIT hashing	50
4.7 PIT as a Database ENGINE:.....	51
CHAPTER FIVE	56
5.1 Conclusion:	56
REFERENCES	59

CHAPTER ONE

1.1 INTRODUCTION:

The global usage of internet since its evolvement in terms of connecting network devices and sharing of resources digitally based on common protocols functionality is still the same, but problematic in architectural design due to insufficient support with network components in distributing content immensely, securely for reliable performance of network that will cope with current robustness and overlapping of different users with sophisticated devices for content retrieval. Internet is a network of networks that connect and share resources among millions of users based on routing policies across the entire network paths, therefore; this task require reasonable designing model to adapt and maintain persistency especially for data provenance in between users applications. The Internet carries an extensive range of information resources and services, such as inter-linked hypertext documents, videos conferencing and streaming, online games and many more based on applications of the World Wide Web to support the navigation of multimedia links, businesses, historic, life style and educative hypertext, e.t.c. across the globe. This is carried out either via wired and wireless services and telecommunication system from one region to another using different service provider – normally known as Internet Service Provider (ISP) [2].

The internet evolutionary has shown that, its architecture was built to allow devices in a network to communicate with one another using IP addresses [1, 2]. The early stage of internet design was started since 1960s as a testing experiment under the control of Advanced Research Projects Agency (called ARPAnet formally known as DARPA) of the U.S Department of Defense with aim of connecting some computers from universities and private companies, were the work begins around 1969 with only Four-nodes network devices to implement an online experiment using 56kps circuits [2]. The result of this experiment was successful that lead to the designing of two military networks-MILNET in the U.S and MINET in Europe, later on; several users connected their private networks to

participate the advancement of this new technology “world of things”-although the addition of more networks to ARPANET brings scalability problems as a result of link congestion.

Moreover, later in 1989 National Science Foundation (NSF) an American research Organization moved ARPANET to NSFNET for distributed connectivity architecture by creating a network with centralized backbone to provide high intelligent interconnectivity between Campuses, research organizations and private companies [2].

Nevertheless, after some decades the maintaining tasks of internet interconnectivity continued by NSF and other organizations with objective of connecting a number of Autonomous System (group of routers under the same administrative control that exchange routing information based on common routing protocols) from different region usually with Network Access Point; a technology that will enable customers from different ISPs to connect to one another [2].

However, today’s internet is flexible with its original designing goal; naming communication end points for global interconnectivity [20] but users operational needs are changing dramatically with very high desire of online resources generation, distributive sharing any time everywhere, by different consumers irrespective of location nor does the content sources and the transmission medium. Because, internet users care only about the resources they want retrieve or broadcast to others without an idea of who is going to use the data or to whom it comes from, due to the current evolvement of new online technologies and services like sophisticated smart phones, tablets, facebook, twitter, whatsApp and others, as shown in figure one below. As such uploading and downloading of desirable content becomes easy and even seize the interest of many internet users nowadays, although others use internet yet now as educative and informatics forum; in which people create and share important information on popular web pages so that others can read and benefit. Therefore, with this diversity and similarities toward the need and use of internet resources, a lot of challenges also

evolved like packet routing effective performance, user reliability of internet services, authenticity of data, all this and many more need serious attention in the architecture of internet [15]. This is going to be discussed in the subsequent section of this chapter and the entire research work.

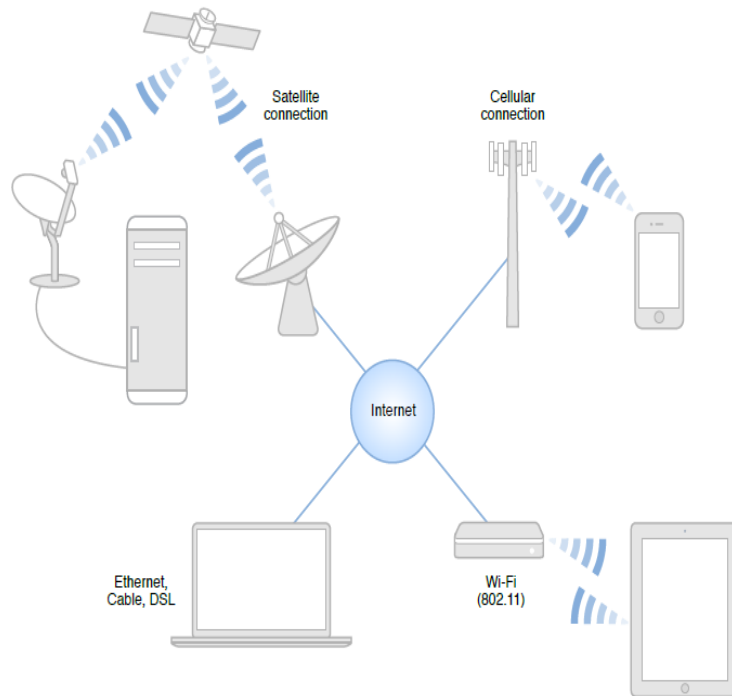


Figure 1.1: Popularity Use of Internet

Therefore, due to the aforementioned problems of today's internet, there is need of content-centric network, which will bring reliable content distribution globally with less cost, although a lot of mechanisms have been employed currently to provide content delivery and distribution among internet users, some of them are presented in subsection of this chapter, but they are implemented with many drawbacks compared to one proposed in future internet architecture; where every network node can replicate content along a communication media based on router caching policy that lead to content distribution throughout the internet. This issue is going to be discussed thoroughly within this research work, given more emphasis to routing and caching reliability in the future internet generation, for effective, scalable content distribution as a built-in feature for the architecture.

This thesis is organized as follows; chapter one relate the conceptual design goal of today's internet in terms of routing, content caching and distribution related with the proposed future internet architecture to shed light about the new design principle and direction of the architecture and focus of this research work, including the major objectives . Chapter two present the designing stage of Named Data Networking (NDN) architecture as a selective and progressive architecture of next generation internet. Chapter three will discuss the proposed mechanism. Chapter four gives the simulation result toward the reliable of routing and caching in NDN. And the thesis is concluded in chapter five with some recommendation.

1.2 Current Internet Architecture

The existing internet is based on TCP/IP architecture comprises a hierarchy of layered functionality to initiate the routing, and forwarding of request from a client to content producer, based on IP addressing that can allow the communication of connecting parties [23]. The overall working paradigm of internet is judged by a set of rules (protocols) that determines the best paths to locate a particular host for a particular content. These protocols are distributed according to the layer they reside and the function of the layer in the network activities, as it can be seen in table one below showing the five layers of TCP/IP model [23], and each of the layer corresponds to one or more layers of the seven-layers in Open Systems Interconnection (OSI) reference model proposed by the International Standards Organization (ISO). The layered architecture of this protocols was organized by network designers in order to allow the discussion of well-defined and specific functions of every protocol either in a small or a large and complex systems, that indicates the performance and function of a protocol belonging to a particular layer for a particular function for their supportive measurement either to software or the hardware of the system model. For example, physical layer and data-link layer are responsible for handling communication over a specific link typically implemented in network interface card (e.g. Ethernet or WiFi interface cards). The table below contains the TCP/IP

protocol stack and some example of protocol or technology found in each internet layer.

Layers	Example of protocols/Technology
Application	HTTP, FTP, DNS
Transport	TCP, UDP
Network	IP, ICMP
Data Link	FDDI, Frame Relay
physical	DSL, ISDN

Table 1.1: TCP/IP Protocol Stack

- Application layer: this layer is responsible for data encoding and representation to the client, it ensure the integrity of data exchange between the transmission devices that establish connection agreement between clients. For example; Domain name system protocol-DNS translate human readable name into a 32-bit network address for internet request from client. Hypertext transfer protocol-HTTP deal with web document request and transfer between client and content server.
- Transport layer; it supports communication between devices in the internet. The protocols found here are; transmission control protocol-TCP and user datagram protocol- UDP.TCP provides connection-oriented service between end points, more especially in a large network in which a bulk message is segmented for reliable transmission with accurate congestion control.
- Network layer; this layering establishes transmission of packet from one host to another; by determine the best paths through the network. It passes the packet segment and destination address sent by source host transport

layer protocol (either TCP or UDP) for service delivery to the right destination host. Major protocols here are; popular internet protocol IP protocol (IPV4.IPV6). It is responsible for packet delivery from one host to another solely based on IP address in the packet header. There is also internet control message protocol-ICMP it is mainly used by the network devices to send error messages indicating whether the requested service is not available or the content source cannot be reach due to link failure.

- Data link layer; this layer are more responsible for transmission of frames (digital data) between devices in the same local area network (LAN), that includes physical addressing and error control. The services provides by link layer depends on protocol employed along the link.
- Physical layer; physical layer move individual bits within a frame from one node to the next, that largely depends on the physical network hardware (like coaxial cable, fiber optics, twisted pair cable) and other transmission medium. This layer has multiple functions concerning bit streaming between physically connected devices, which includes signal modulation, network interface card configuration, circuit switching, and multiplexing e.t.c.

1.2.1 IP Routing:

Routing is one of the major functioning features of internet activities that enable the transfer of data from content server to consumer device via the best path- mostly done by routers. In IP network, request of data is satisfied between two end points based on the following conditions but undergo the aforementioned layered structure [24]:

- i- Destination address of content source.
- ii- Neighboring routers from which it can learn about the remote network.

- iii- All possible paths; the least cost is always chose, and is considered the best.
- iv- Client router must know how to maintain and verify routing information in the routing table for consistency.

Routers around the internet communicate with one another using; Interior Gateway Protocols-IGP (for devices within the same autonomous system-the most common routing algorithm used here are; Link State and Distance Vector routing), and Exterior Gateway Protocol-EGP (for exchanging routing information between ASs; like Border Gateway Protocol-BGP) [2, 23].

- ❖ Distance Vector Protocol; routers find the best path to a remote network by judging distance. The path with the least number of hops to the network is considered to be the best and send the routing table only to connected neighbors. Example Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP).
- ❖ Link State Protocol; routers implementing this protocol create three tables; one keeps track of every connected neighbors, one determines the topology of the entire internetwork, and one is used as the routing table. These protocols send updates containing the state of their links to all other routers on the network and know more about the entire network than distances vector. Example Open Shortest Path First-OSPF and Intermediate System-to- Intermediate System-IS-IS.

1.2.2 Caching and Content Distribution Support by IP Network;

A content delivery network or content distribution network is an extensive network of advanced data centers that include the deploying of distributive server, in order to serve end-users with high availability and performance. In IP network, the need of content delivery networking is always increasing which brings serious challenges; considering the high percentage of web pages lookup and downloading across the internet, due to consumers growing needs. For instance, many people access internet for various reasons but particularly and majority for e-commerce, sending and receiving of messages via email addresses and social

media, and online games e.t.c. However, the desires of these applications are increasing rapidly to the extent that may degrade the performance of internet services. This is what brings the idea of content distribution network, so that many servers are deployed to replicates popular content closer to clients, in order to reduce network bandwidth usage and latency of obtaining content by end users [25].

The web application developers are in doubt on how to come out with something that could make web surfing to experience an intelligent and interested working environment to ensure easy passage of messages via a distributed network of data, because there is inconsistency, unreliability and inadequacy in performance for serving content from a single server. Some developers suggest the use of local clustering to improve fault-tolerance and solve scalability problem but has its own tradeoff [26], because if the data center or the ISP providing connectivity fails, the entire clients located at such cluster will not get access to normal services, that may incur loose of resources. While Akamai in [26] suggested the implementation of more server to sites experiencing high load so that clients can be serve from nearby servers.

But nowadays, content distribution facilitate the use of web caching that includes keeping history of previously accessed links and information either in a proxy server called cache proxy or in users' browser formally known as browser cache, in order to satisfy future request of the same content [25]. Proxy server is a computer application acts as intermediary for requests from clients seeking resources from other servers. Proxies were invented to add structure and encapsulation to distributed system, which simplify serving of content, controlling and directing clients' request of files, web page connection and other resources to the right server in the internet. Therefore, a proxy cache is a shared network undertaking web transaction on behalf of a client and store the content, so that future request of the same content can be satisfy without downloading from the original server. While browser cache is part of all popular web browsers, it keeps a local copy of every recently accessed web page and when subsequent request of

these pages arises it makes use of its local copy to satisfy the clients instead of fetching the content from the content source on web.

Therefore the technology of web caching reduced transmission of redundant network traffic, improve quality of service, response times and effectiveness of transmission bandwidth, for both users depending on small and slow dial up links as well as those with relaying on faster broadband connections [25, 26]. The current internet web caching technologies are mainly produce and manage by commercial software and hardware producing companies such as; Squid, Cisco (Cisco cache engine), Microsoft (Microsoft proxy), BlueCoat e.t.c. The retrieval of services from the internet relies on fundamental protocols like HTTP and others, which directs the request of client to the right server and response back through web browser. The communication facilitation between a web client and the HTTP is performed using different mechanisms called methods, which are mentioned in table two below;

<i>Method</i>	<i>Details</i>
GET	Get the specified data
PUT	Enables the client to put a new item in a specified location on the server
DELETE	Enables the client to delete the specified item
POST	Enables the client to post information to the server
HEAD	Similar to GET, the server returns no page content except for HTTP headers
TRACE	The client gets to trace the request it made to the server
OPTIONS	Helps the client to ascertain the options for communication at the server

Table 1.2: HTTP Methods - [25]

Moreover, the current internet architecture supports content-centric networking using some common technological paradigms which resemble the above mentioned distributive networking mechanisms. The most popular technologies

are peer-to-peer (P2P) technology and content delivery network (CDN), as explained below;

1.2.2.1 Peer-to-Peer Technology;

The P2P network technology is a decentralized and distributed computer network architecture used in today's internet as a means of providing content-centric networking services, through organizing peers to equally share services in a cooperative fashion. [22] Each interconnected peer contribute some portion of their resources such as network bandwidth, processing power and disk storage, so that the throughput of retrieving a particular content will become more efficient that may lead to scalable content management and distribution [25]. In this technology every peer is expected to run a piece of software that does not need to support from the core internet applications beside conventional TCP/IP as depicted in figure two below, in order to provides heterogeneous platforms for content delivery and self-managerial task among the multiple P2P users. It is the peers' agreement that allows each peer to make a copy of content along a transmission path within a P2P network.

The overlay designing of P2P can be done structurally and unstructured, in structured p2p network, the peers normally creates a virtual address space to define relationships between peers, each peer is assigned a local address space responsible for announcing the availability of a particular object within that location [22]. While in unstructured P2P network; a node sends out a query packets by flooding the network with a smarter algorithm to search for nodes with matched requests, so that any node with available content will respond to the requester [21].

1.2.2.2 Content Delivery Network

The content delivery networking technology is mainly used for distributing, replication and redirection of client request of content, irrespective of consumer location upon a compromised subscription to the technology vendors of these services. [22] It is one of the technology providing replication and placement of content in the current internet, usually owned, deployed, maintained and

implemented commercially by an individual operator. It imply charges to both content producer or web site owner for its services, but it is transparent to web users for replicated content or in redirection of client request to appropriate content server. This technology is provided by information technology companies like Cisco, Limelight communication technology and many more, supporting content centrality based on TCP file transfer, as shown in figure two below.

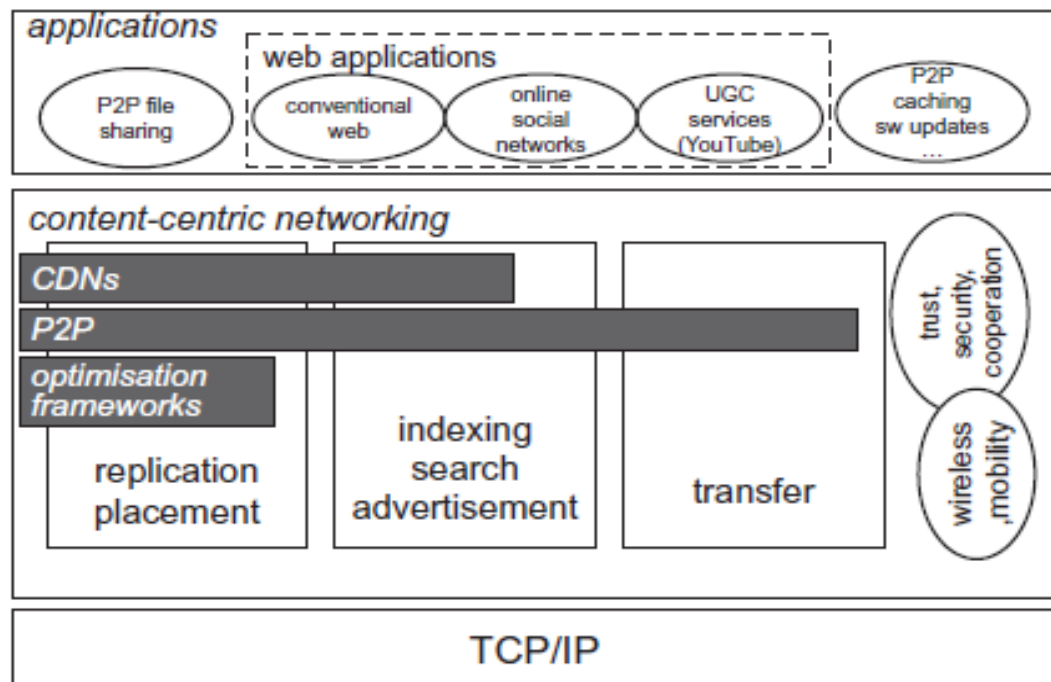


Figure 1.2: content centric networking in Today’s internet -[22]

1.3 Today’s Internet Architecture Challenges;

Based on the idea of today’s internet architecture presented above, we can judge its progress and major challenges enclosed in it, especially for today’s user’s requirement to satisfy global need in a cooperative fashion. The major challenges are mentioned below;

- a- The routing and forwarding aspect of current internet does not support multiple choices of IP prefix forwarding without looping. Because, multipath routing and forwarding is one of the solution of insecurity in any

distributive network, in which failure in one link will not prevent data retrieval from other paths. Therefore, this is one of the promising features of future internet architecture; the detail is going to be explaining in chapter two.

- b- Security is greatly important for data associative environment, but there is security vulnerability in the current internet, because the security are attached to communication link and content sources failure to any of them can render the risk of losing data. But next generation of internet architecture secure data not location or the paths.
- c- The solution to content distribution and delivery in the current internet has serious challenges like; (i) high cost of using resources across the internet due to charges from individual technology operators for content distribution mechanisms (ii) high risk and unreliability from using free services like proxies for content retrieval of redirection client request e.t.c.

1.4 Future Internet Architecture;

Future internet architecture (FIA) is a promising movement by group of engineers/scientist toward changing the designing goal of today's internet into a globally distributed and accessible network with adaptive and reliable routing and forwarding which can allow caching of data ubiquitously within a network to reduce data access latency, traffic overhead along working path(s), bandwidth utilization, and to improve the performance of the entire network. There are many ongoing proposals on next generation of internet since 2010 funded by National Science Foundation (NFS)- from different geographical location aimed at changing host-based networking of today's internet into content-centric networking paradigm: refer to [1]. Some of the proposed architectures were; MobilityFirst, XIA, Named Data Network-Named Data Networking (NDN), NEBULA e.t.c. but NDN was chose to replace the current internet architecture for the following reasons [19, 20, 21]; <http://www.named>

- It combined most of the major designing goal of other FIA with additional functionality to maintain some features of the current internet.

- NDN prioritized data not location in designing goal of internet operation.
- Trustworthiness is attached to data not channel or a particular host.
- It provides scalability in routing, forwarding and caching within a network.
- Economic incentive in the side of ISP and their clients, due to bandwidth maintenance and reduce latency.

This program has been dually supported by many international research companies like Palo Alto Research Center (PARC), National Science Foundation (NSF)-Originator and many Universities around the world, more especially from United State of America, Europe and Asia such as Standford University-California, Yale University. University of California Los Angeles, and University of Memphis, e.t.c. [27].

1.4.1 Named Data Networking (NDN) Architecture;

The NDN architecture focuses on replacing today's IP addresses by naming data directly in order to build network application, for effective communication environment without considering data location or where it follows between a requester and the producer. The security of the architecture is built on the data not its location [1, 20]. Its features overcome many challenges in today's internet; like content distribution, truth verification by data receiver- it allows every node to verify the data before making copy, forwarding adaptability, and network address space minimization by implementing hierarchical naming structures that give unbounded namespace.

NDN architecture is designed to run over any network and vice versa, because *“NDN is a new architecture but whose design principles were derived from the success of today's internet, reflecting the understanding of the strengths and limitations of the current internet architecture [20]”*. Therefore, this shows that there is infrastructural transition between today's IP network application and NDN that may involve little modification and extensions, as shown in figure three below. For example; early implementation of NDN was done using extended version of current routing protocol; Open Shortest path First-OSPF that shift to

OSPFN, [9] a very popular link state routing algorithm. Another example is HTTP a well known to be an application layer protocol used in today's architecture, particularly for request driven model between a client and content server, it guide and direct a request for retrieval of a particular content or web pages with help of Universal Resource Locator (URL) [1]. Therefore, this can be replaced with named data instead of URL to facilitate the processing of resources from the internet irrespective of location.

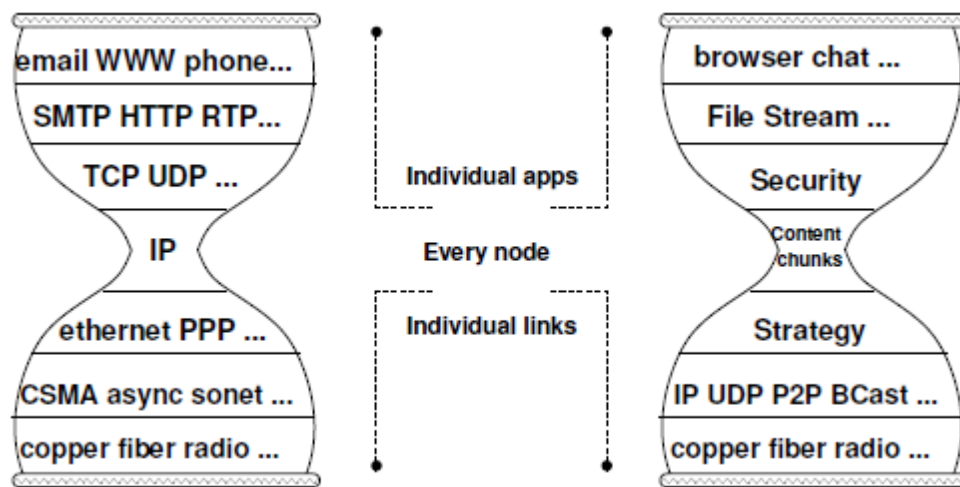


Figure 1.3: Infrastructural Transition from IP network to NDN-[1]

1.4.2 Expectations on FIA-NDN;

The new architecture is expected to show an intelligent improvement, more especially in the following point of interest.

- ❖ Persistency; the new architecture have started with durability measure in respect to data retrieval, because failure along an attack path will not deny the processing of a particular content, due to multipath forwarding of name prefixes.
- ❖ Availability; NDN distributive caching will allow content availability in the network without implementation of any proxy(s).

- ❖ Truth; the security issue in NDN is always to be testified by the node receiving network resources because every NDN verified any data upon receiving to ensure the authenticity of the data using the verification sent with data packet.

1.4.3 Other Related FIA:

The FIA project are many each presenting different approaches and structure ranging from PC base computing to mobile computing, content distribution and delivery and security, but NDN and the ones mentioned above consider the best [27]. For example;

- MobilityFirst: This architecture aim to address the issue of interconnecting fixed endpoint for resource exchange in the current internet, by introducing and encouraging cellular convergence among mobile devices- focusing the spread of services to various location for reliable content multicast delivery, it also includes the use of strong security mechanisms, trust requirements among services due to dynamic association.
- NEBULA: Is another FIA program giving emphasis to cloud computing centric network architecture, aim to interconnect various data centers to provide content availability among the cloud users environment through the use of data replication via wired and wireless links connecting nearest data centers.
- Expressive Internet Architecture (XIA): This architecture focus on security by using self-certifiers for all principals to ensure good building blocks between the communication entities.

CHAPTER TWO

Introduction: This chapter explained the design stage of Named Data Networking (NDN) architecture, where the fundamental issues in relation to routing and caching will be presented including their challenges and some basic solutions measured to by researchers in order to make it more scalable and robust. To avoid confusion the words; user, client, consumer are mean the same.

2.1 Named Data Networking (NDN):

Named data networking (NDN) is a newly proposed future Internet architecture that considers data as the first class entity in the entire network communication system. The NDN routers announces name prefixes throughout the network for requesting contents irrespective of producers' location or addresses [1, 2, 5], unlike in the current Internet, where both the requesting party(s) and producers have to know their IP addresses to initiate resource transmission. NDN uses name prefixes to identify and retrieve data not necessarily from the content producer because of its in-network caching, which allows the intermediate nodes along the routing path, to copy any data passing through it based on router caching policy.

NDN architecture is an instantiation of content-centric networking (CCN) also known as Information-centric networking (ICN) [2, 8, 13], which facilitates content distribution and delivery with better communication model for scalability and security of data, between individual users connected to Internet that require less latency and bandwidth capacity. Because, the current internet is experiencing global challenge of resources management and allocation, due to increase in extensive use of internet services for daily activities, as a result of desired content availability and the evaluation of other online services and social network. Therefore, users are always increasing to surf the internet for different purposes whether in their houses, offices, schools even in vehicular devices such as train buses, which at the same time increased internet mobility using sophisticated mobile devices of different model.

Obviously, the commonness and diversity of the use of internet for the need of resources with different views from users, the internet infrastructures are facing serious challenges; because attackers on other hand are always trying to deviate the users legitimacy and the entire internet architecture functionality, through the introduction of malicious content that will degrade the network performance [15, 16]. As a result of this, NDN evolved to maintain the hourglass shape of today's internet by including named data instead of location at its thin waist, that brings efficient and secured data retrieval due to digital signing of all chunk of data from origin to avoid further middleware configuration between network layer and application layer [1].

2.2 NDN Operation:

NDN is user-driven communication based on two types of packets; interest packet and data packet. Interest packet means a user request for a particular data in the network, while data packet means the corresponding data in return from a content server or any node with matching requested data. The consumer sends out an interest packet throughout the network which carries the name of content identifying a block of data in order to get the desired data packet, without knowledge about the content producer, because NDN client believe that the request can be satisfied from any node within the network [21]. Any router having the matched data will respond to it by sending the data packet along the same way the interest passed, because an NDN interest request always leave a trail trace during the forwarding process, so that a symmetric routing is made between the two packets from the requester to the content producer or any node satisfying the interest. Every interest packet is always satisfied with corresponding name prefix in the content name in the data packet.

The NDN interest and data packet are different from IP packet not only considering the replacement of addresses with named data, but there are some special fields attached to both interest and data packet which brings uniqueness and security issue to both consumers and producers. For example, the figure below shows the content of interest and data packet with all the necessary part.

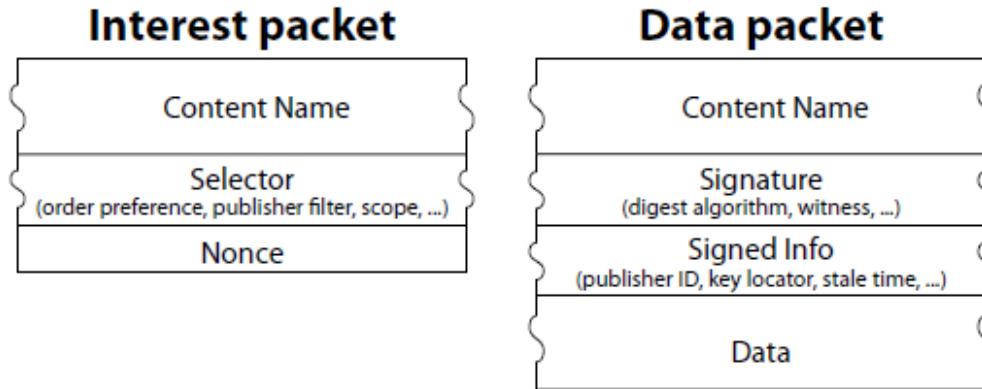


Figure 2.1: NDN Packets [Jacobson et al, 1]

The fields indicated in the above diagram are explained below; starting with first one Content Name found in both packets, data can only satisfy an interest if the content name in the interest packet is a prefix in the content name of data packet [Jacobson et al, 1]. The nonce in interest packet is unique random value attached to interest that keeps track of matching data packet normally generated by data consumer. While during network congestion; the NDN routers use the nonce value to detect interest duplicate by remembering the name and nonce of each received interest to determine whether the newly arrived interest is indeed a new one or an old one. It loops back any identified interest duplicate, to allow others to get into the pending interest table (PIT) for processing. The selector field also offer an important function in data retrieval, where it prefers an interest to obtain data packet from least cost path during the routing of interest packet.

Moreover the attached fields in the data packet are created and announces by the original data producer including the necessary information to verify authenticity and integrity of the requested content [12];

- i- Signature field; this signature binds the content together with data name.
- ii- Key locator field; the key to verify content signature, it contains the verification (public) key, certificate containing verification key and NDN name referencing verification key.

- iii- Exclude field (optional); name components description that should not appear in the data packet in response to the interest.
- iv- Answeroriginkind; determine data packet should whether be from CS or by the producer.
- v- Scope; this field limits where the interest may propagate that are leveled with numbers; 0, 1, 2. Scope 0 and 1 limits propagation to the originating host, and scope 2 limits propagation to the next host.

2.2.1 NDN Components

The routing, forwarding and caching of data in NDN is performed within every router's control plane containing three tables; Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB).

- Content Store (CS); is the route's buffer memory, where cached data is stored for a period of time, and serve future request of consumers with the available data to reduce user's latency and save network bandwidth [3, 4].
- Pending Interest Table (PIT):

This is a table containing name prefixes of unsatisfied arrival interest and corresponding incoming interfaces. PIT entry records the interest name, the incoming interface (s) of the interest and the outgoing interfaces which the interest can be forwarded for retrieving data packet, based on Forwarding Information Base (FIB) longest prefix match, as shown in table 2.1 below. The entries are removed from the PIT after the lifetime assigned to it expires without data packet returns, so that it is left to the consumer to retransmit again, in order to reduce PIT explosion. PIT avoid forwarding of request with the same content name, the router only appends interface of the new request having the same name with the one already forwarded. PIT is responsible for data multicasting delivery based on its entries, because multiple entries with the same request can be serving at the same time. It initiates and coordinates

the routing of interest packet without specifying a source or destination address.

Pending Interest Table	
Interest Packet Name	Incoming Interfaces
NDN:/google.com/Videos	Interface02, Interface04
NDN:/yahoo.com/mail	Interface01
NDN:/yasar.com/News	Interface05, Interface09

Table 2.1: Pending Interest Table Entries

- Forwarding Information Base (FIB):

FIB is a table of name prefixes and corresponding outgoing interfaces, in order to route interest to the matching data packet. It contains multiple interfaces to forward different consumers' request as shown in table 2.2. It decides where and what are the longest prefix matches for interest packet to follow. NDN FIB differs from IP FIB in two ways; (1) NDN FIB contains multiple forwarding interfaces for next-hop count while that of IP contain single next-hop. (2) IP FIB contains next-hop information only while NDN FIB contains information both from routing and forwarding plane to provide better forwarding decision based on the updated network information to avoid following failed link.

NDN Forwarding Information base Table	
Contents Name/Prefixes	Outgoing Interfaces
NDN:/google.com/Videos	Interface25, Interface14
NDN:/yahoo.com/mail	Interface30
NDN:/yasar.com/News	Interface15, Interface45

Table 2.2: Forwarding Information Base

2.2.2 Forwarding Process:

The forwarding of interest for processing of desired data undergoes some series of steps within the NDN router's control plane with adaptive behavior between the two packets. NDN forwarding is different from that of IP network; because the retrieval of data in NDN is performed along the best performing due to multiple forwarding interfaces in routers' FIB and this allows quick detection of packet from any attacked communication path. Another feature that makes NDN forwarding more adaptive is the introduction of NACK interest from upstream to downstream after the expiration of assigned round trip time to an interest without data packet return [19]. The NACK interest is sent to explain the purpose of not satisfying the interest, which contains the name and the nonce value of an interest, specifying either as a result of congestion of duplicated request or no data that match the interest. Therefore, every interest packet arriving at NDN router takes path through the following steps, the details are shown in figure 2.2 (flow chart for Interest Packet Request)

- i- Upon receiving an interest packet in NDN router, the router first checks its CS if the desired data is available, it will respond to it
- ii- If not found, the router will forward the interest to its PIT to check whether there is another interest with the same name awaiting for data packet, if there is, it only appends the receiving interface to PIT without data name, since it already exists and does not allow duplicate of the same name.

- iii- Otherwise, creates a PIT entry for this Interest and add the name and the incoming interface.
- iv- Forwards Interest to the next-hop interface by looking up FIB.
- v- When Data packet returns, the PIT forwards the Data packet over all the requesting interfaces in the corresponding PIT entry and deletes this entry.
- vi- The CS caches the Data packet based on policies.

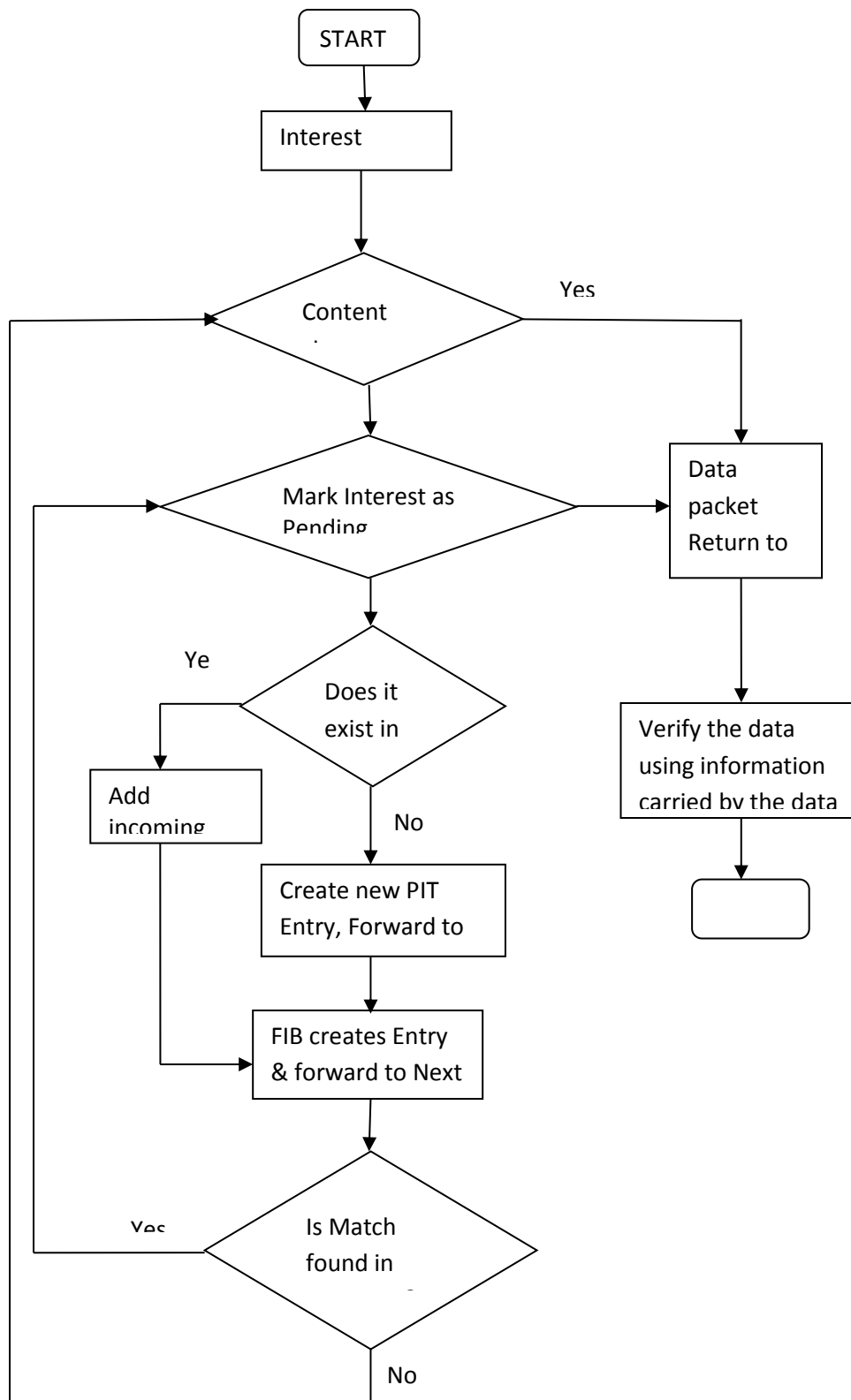


Figure 2.2: interest Request Forwarding

2.3 Advantages of NDN over IP Network;

The architectural shift from IP network to NDN brings a common interest between clients and the designing goal, because users care about the content they want surf not the source of the content or the transmission media. Therefore, the following are some of the NDN incentives;

- 1) It supports multipath routing and forwarding without looping or congestion problem due to interest/data packets flow-balance. This brings the idea of adaptive forwarding by allowing each router to retrieve data via the best path and measure data delivering performance through considering request round trip time, cache hit, latency and throughput [17]. and this lead to link failure detection, congestion control so that the retransmission of interest packet follow another direction, because of NDN multiple path forwarding interfaces in routers' FIB. Therefore, this adaptability of NDN forwarding reduced routing plane task and dependency, making it to concentrate on network routing update dissemination.
- 2) In-network caching; NDN allows routers to cache every data packet passing through it on traversing, and this lead to reduction in bandwidth consumption and data access latency [6].
- 3) Multicast data delivery; different PIT entries can request the same data synchronously or asynchronously, therefore; in return of data packet the PIT can send to various recorded interfaces.
- 4) NDN architecture secure network content not the channel; by encrypting data digitally together with its name from the origin, before announcing or giving access upon request. Unlike in IP network that secure the channel or the hosts, in which failure to a host or the transmission media will brings loss of packet[1].

2.4 Naming in NDN:

NDN names are hierarchically structured which composes of multiple components each with variable-length that can allow the formation of any number

of identifier within a single domain representing relationships between chunks of data [5, 17, 18]. The name is opaque to the NDN routers- it only knows the boundaries making the name delimited with slash “/” character or dot, although it is part of the name but are not included in the packet. The NDN naming scheme follows some basic feature of current URL addressing in terms of uniqueness and tree hierarchy but without source or producer addresses and protocol port number. For example, the CNN News with URL address `www/cnnNews/EgyptCrisis/2013` can be represented in terms of NDN name as `/NDN/cnnNews/EgyptCrisis/2013`, this show that each part of the components indicating the direction of path toward data retrieval from child tree to the domain name. A consumer can make request of a particular content using the full name of the content or its prefix; e.g `/NDN/cnnNews/` is a prefix to `/NDN/cnnNews/ EgyptCrisis/2013` and the match data can be returned to the requester.

2.5 NDN Routing:

Routing protocols in NDN are responsible for disseminating network topology, computing routes and handling short term network changes. Routing strategies in NDN initiates forwarding process were aimed to provide multipath routing and multiple path selection to support content dissemination without looping. NDN multipath routing means a host can obtain data from multiple content providers through multiple paths [5, 6, 7]. Unlike in IP network in which multipath routing can only be possible either by looping or the two communicating hosts should have more than one path and will incur bandwidth consumption and traffic congestion.

After thorough discussion by engineers and scientists, they come up with two designing stages: initial design and long term design. In the initial designing of NDN the extended version of intra-domain Open Shortest Path First (OSPF) protocol- a link state routing protocol that uses name prefix to process interest request and inter-domain Border Gateway Protocol (BGP)-responsible for routing coordination reachability between Autonomous Systems in the internet were implemented for their internet functionality [16][17]. These protocols extension were implemented to meet NDN prototyping in;

- Multipath forwarding without looping (true feature of NDN).
- Paths selection to maximize data delivery performance.
- Minimizing computation, latency and overhead.

2.5.1 Open Shortest Path First (OSPFN);

OSPFN is an extension of OSPF link state routing protocol for NDN that supports name prefix routing and distribution with multipath routing configuration in order to retrieve data from multiple interfaces within the network, so that users can select working links when the best link fail to bring data back [9]. It uses opaque link state advertisement to announce name prefixes for ensuring backwardness compatibility by the PIT entries, but maintains shortest path calculation to provide only single next-hop for each destination. OSPF has a link state database where the link state information of the entire network is stored, which can be updated upon receive a network changes. The entire designing of OSPFN had really support the major NDN features but lacks automatic multipath selection and naming system need additional task, because of bounded component naming just like IP address.

2.5.2 Named-Data Link State Routing Protocol;

NLSR uses name to identify and retrieve data from a list of forwarding rank options for each name prefix, due to multiple routes calculation to each name prefix, which facilitates NDN adaptive forwarding. It propagates link state advertisement (LSA) throughout the network and allows every router to build its network topology based on its adjacency LSA and all associative routing information [7]. Every router has a link state database (LSDB) where latest versions of LSA are store in order to keep the network stage at all nodes. Because, routers exchange their hashes of the LSDB periodically to detect inconsistent and recover from them. The NLSR design an associative hierarchy of name, where routers ID are named according their domain, keys are associated with their corresponding owners and any routing updates

originated by NLSR should have the process name as its prefix to easily identify the message originator.

Other routing protocols proposed are;

- A two-layer intra-domain routing scheme for NDN with two layer task partition for routing name prefixes and link state advertisement broadcast in order to update link state database routing information. Topology maintaining layer for network topology maintenance and calculation of shortest-path tree. Secondly, Prefix Announcement for content advertisement either in actively published or passively serve. There is FIB explosion when content are published actively and give also network traffic congestion if content were served passively. That is why, in the end they conclude a compromise; so that popular content to be published actively and unpopular content to be serve passively [16].
- Controller-based routing protocol; the controller is responsible for named data location and routing. It learns the topology in the bootstrap phase and compute routes to all the routers.

2.6 Caching in NDN;

Caching in network aspect is a mechanism for providing temporary storage to reduce network bandwidth, server load and response time [3]. All NDN routers are allowed to cache data copy along the forwarding path on traversing for future use upon requesting the same named data-this reduce the overall network load and content access delay (figure 3 below illustrate NDN in-network caching). Unlike in today's Internet caches are located in specific servers and replicas can be placed in any of these caches [3].[25] Despite the NDN built-in caching feature, there is serious challenges in cache privacy and the entire cache management which will some time render the vulnerability from unknown adversary(s) as a result of frequent request from consumers for Internet resources, because of the presence of multimedia files, on-line games, movies and social networks activities involvement, and this is in-line with various and distinctive attacks from other side.

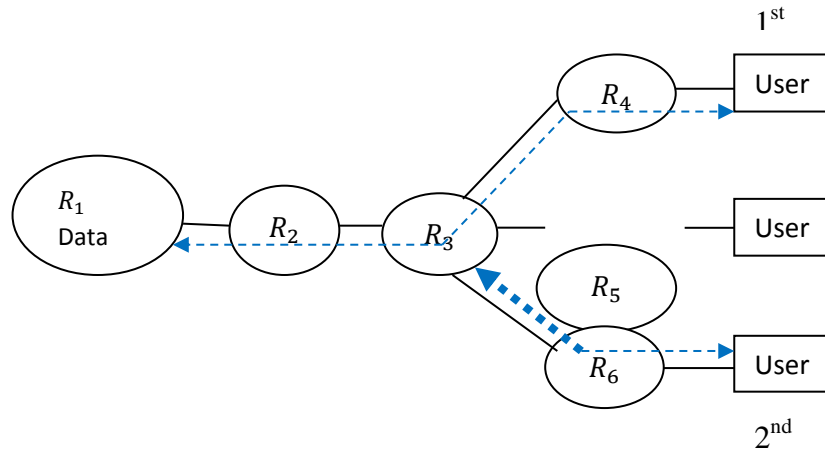


Figure 2.3: NDN in-network caching

The figure 2.3 shows that the 1st user send request to R_1 -content producer for a particular data object, and the matched data packet return to requester following the same way the interest was sent. R_2 and R_3 cached the copy of the data packet on passing through them. Later 2nd user make request for the same data requested previously by 1st user. 2nd user request was responded by R_3 instantly from its cached copy, that reduce latency of getting the desired data packet, because it does not need to reach the original content source R_1 as a result of NDN built-in caching

2.6 Cache policy:

This mechanism decides the caching of content based on data packet behavior from historic analyses of data request within the network. It can either be the popularity or unpopularity of the content in between the communicating parties involved. The default policy is to cache everything [11]. Prior the review of NDN caching presented in the previous section, this is an important aspect of cache privacy; because some content are less confidential than the others, for example people give emphasis to videos than historic data in the internet; this can allow hackers to flood malicious videos that affect the system even during caching.

For this reason, many ongoing researches had tried to come up with reliable caching scheme for NDN scalability, referencing paper [5]; the idea of “cache less for more in information centric networking” by presenting centrality-based caching algorithm using betweenness centrality to improve caching gain as shown in the equation (1). This scheme cache content along the content delivery path between requester and the producer. Betweenness centrality; measures the number of times a node lies on the content delivery paths between all pairs of nodes in a network topology. Their performance evaluation shows that the scheme increase cache hit by reducing cache replacement rate that lead to overhead- “interested idea”.

$$C_B(V) = \sum_{i \neq v \neq j \in V} \frac{\sigma_{ij}(V)}{\sigma_{ij}} \dots\dots\dots(1)$$

Where σ_{ij} is the number of content delivery paths and $\sigma_{ij}(V)$ is the number of content path passing through node V.

Secondly, popularity-driven coordinated caching in NDN provides effective caching by allowing routers within an ISP to coordinate their caching decision, so that NDN router can only cache the popular content [3]. The popularity of content is determined by the router’s updated historic statistic; measured using Topdown caching and AsympOpt caching algorithm to compare their incentives in the world of inter ISP traffic minimization. These algorithms involved using two methods-Aggregation stage and decision making point. Information aggregation involved sorting out the content object historic data from bottom nodes to upper ones [3]. While the decision is made based on the updated request record from the aggregation table. Each node makes caching decision for each content object according to its popularity measured by the aggregated request records statistics of its sub-tree.

2.6.1 Cache replacement policy;

This policy decide the eviction of object content from routers’ caches for the new incoming one when there is no availability of space to accommodate the another

object. This is also important in caches management, and is normally done based on object arrival time. The default mechanism is carried out randomly, but other suitable approaches are now available such as; first-in-first out (FIFO), least recently used (LRU), least frequently used (LFU). e.t.c.

For example, the use of Least Recently Used (LRU) replacement policy to measure a relationship between time for requesting content and delivery time, cache size and bandwidth using derived mathematical expression [4]. LRU is popular cache replacement policies that always replace recently used content to give room for new one based on round trip time. This method minimizes bandwidth but incur overhead during repeated replacement of content with time.

2.6.2 Present NDN Cache Attacks and Countermeasures:

NDN cache attacks take different form, directly or indirectly from the cache behavior, because; NDN routers connected to a single cache(s) can be able to share some local information like popular contents, number and time of access to a particular content so that the adversary decide how to mount an attack. Based on this cache characteristics, cache attacks can be classified as follows:

1) Interest flooding attacks (IFA): This affects the entire network, in which the adversary (attacker) announces large number of interests that cannot be satisfied by routers caches [12]. the attacker generate a closely related name space with aim of populating PIT in the victim routers so that the legitimate users will not get access to network resources and may even lead to the seize of producers functions. This is a serious attack that will lead to network overhead because of too much computation, if to tackle the problem with signature verification.

The author in [12] mentioned the use of push-back mechanism to prevent the action of the adversary so that if router suspect on-going attack for a particular name-space in its PIT and find out new interest for the same name-space, it will report to routers connected to that interface in order to limit more interest forwarding for the same name-space under attack, this will push back an attack all the way to its source(s).

2) Cache pollution attacks: this is a direct attack to the routers where the adversary aimed to violate the content locality in the cache server so that cache hit from legitimate users/consumer will be missing [8]. This is done either by requesting unpopular content to weaken content locality in a cache (locality-disruption attack), or by filling up the cache with unpopular content due to repeated request of those content object (false-locality attack). It resemble interest flooding attack in terms of filling up router's PIT with frequent request but different from the source of the attack. Because in IFA the adversary does not know anything about network resources, consumers, content producer and history of content object. While in cache pollution attack the adversary might be in the same network with the legitimate user and share some common cache and silently accesses the cache hit performance of a particular object and the requesting party to target an attack based on object hypothesis.

The generic countermeasure called cacheshield function that does not require further coordination from other routers in the same domain [8]. Cacheshield can be seen as add-on to the CS and operate with any replacement policy, containing two components; shielding function and record of content object names. They make use of probabilistic function to compute the shielding function as shown below;

$$\Psi(t) = \frac{1}{1+e^{(p-t)/q}} \quad , t = 1, 2, \dots$$

Where Ψ is probability of shielding function to fetch new content object into CS for future use, t is time interval for request while p and q are function parameters.

The primary function of shielding function is to discourage unpopular content from being cache in CS, by considering content name of an object and number of requesting time, by the use of probabilistic function to computes the shielding function. This technique can decide to cache or not cache, in order to confirm the popularity of an object; that is why there is high cache hit using shield function for less popular content. Their experiment shows that even under attack the cacheshield remain unchanged while for the normal caching get deteriorate.

Other reputable paper [11] suggest the use of reaction protocol lightweight mechanism to detect cache pollution attack with less resources so that the adversary will have negligible percentage of bandwidth and resource access. The lightweight mechanism works based on the following pseudo code:

Inputs: S is the sample set

: T = threshold value

: analyzed_co = number of analyzed content

: co-count = array of content objects in S

: p(i) = probability value associated with co

: $n_r(i)$ = number of occurrence of co

: snap-size = size of window measurement

1: For S ← co ...IDs

2: if co ∈ S then

3: increment co-count [co]

4: if ((analyzed_co + 1) mod snap-size) = 0 then

5: Compute T

6: if T = p(i) , $p(i) = \frac{n_r(i)}{\sum_{i \in S} n_r(j)}$ then

7: attack detected

8: else

9: Go To line 2,

10: end if

11 end if

12 end if

13: end for

3) Cache poisoning: the purpose of this attack is the use of the routers by the adversary to forward and cache fake data packet so that the legitimate consumers will be prevented from receiving right content for their request. In this attack an adversary will hold number of data packet so that upon receiving an interest through the network it response with fake content, although, this attack is less effective due to signature verification by all NDN routes in order to confirm the content security. [12] Therefore, this attack can easily be detected with proper signature verification.

Moreover, the authors of paper [12] proposed another countermeasure called self-certifying and human readable naming (SCN); which allows parties to verify the association between a name and corresponding data object without relying on auxiliary information like public key certificate. This make SCN effective countermeasure against content/cache poisoning attacks [15].

CHAPTER THREE

Introduction: The purpose of this chapter is to present a designing mechanism for enabling reliable routing and caching in Named Data Networking. The overall routing and caching activities are taking place within NDN network control plane which encompasses the content store responsible for data packet caching, pending interest table that keeps track of unsatisfied forwarded interest in forwarding information base with respect to longest prefix match. Therefore considering the functionality assigned to these tables within each router, there is obvious repeated and continues lookup, insertion and deletion of entries between interest and data packets, because PIT insert incoming interest and delete on arrival of matching data, FIB creates another table containing name prefix and outgoing interface while CS evict data to give space for new one using different replacement policies. Based on this operational tasks in the NDN routing tables; they both require serious attention in terms of their designing, operation especially for fast access time due to NDN name complexity. Because, the major challenging factors with this new architecture includes the memory space for name component due to large number of variable length in the interest and data packet, which require more space than the current IP address components.

3.1 NDN Routing and Caching Challenges:

The NDN architecture requires very fast memory in the networking components and the designing principle associated with routing plane that involves; pending interest table (PIT), forwarding information base (FIB) and content store (CS).

3.1.1 Memory Performance:

The size of both interest data packet in NDN is quite huge is very distributive among multiple users. Therefore, there is need of high performance memory to support the new architecture working environment. [27] From origin, random access memory (RAM) is highly used for storing operating system, application program and data in used, which can be quickly access by the computer's processor. RAM is faster than other computer storage like hard disk, floppy disk and CD-ROM in terms of data read and writes. The memory of computing

devices is not part of this research work, but introduced to address the reader about the space requirement in NDN networking and to think of possible solution. The major high speed memories are;

- i. Static random access memory (SRAM): Is a faster and more reliable memory than DRAM, it has access time of about 60 down to 10 nanosecond, which does not need to be refresh over time. It is normally used for network cache memory because of its expensive [28].
- ii. Ternary content-addressable memory (TCAM): Is a specialized type of high speed memory that searches its entire contents in a single clock cycle. The term “ternary” refers to the ability of memory to store and query data using different inputs; 0, 1, and x-where x is the wildcard value that enables TCAM to perform fast searches than RAM but expensive to maintain, normally used in networking equipment such as routers and switches [29].

3.1.2 Design Principle:

This includes the topological arrangement of the network devices and the structure of the data to be processed and store, which can really influence the performance of hardware. This is part of my research work in relation to routing table and caching portion. Therefore, one key solution for successful NDN name routing and caching of data; is the use of dynamic structuring of data for each content name components, which can be implemented using different data structure to measure the performance of content name lookup, insertion and deletion of both interest and data packets within the data plane. The major data structure used in the current computer network storage for the above mentioned parameters are; hash table implementation that uses hash function to map a key to the value of particular item in a data store, bloom filter with rapid memory efficiency in probabilistic determination of an item in a set of arrays, and prefix trie which is an ordered tree data structure used to store associative data array from parent tree to the child node each with a specific key, where the keys can be any value and many more.

The idea of using good data structure within the NDN routing tables will greatly reduce memory consumption, increase effectiveness during routing process and data caching. Because, NDN content distribution brings an extensive improvement over the current internet users' requirement for popular content across the globe, due to its multipath delivery and in-network caching with less bandwidth exhaustion and data access latency.

3.2 Hash Table:

Hash table or hash map is a data structure used for storing a set of items efficiently, implemented using an associative array in relation with two parameters; key and corresponding value, an item can be search, insert and delete for a given key and its single value or multiple values. The hash table operation is perform with a hash function; the hash function takes a given key and map to a possible value. The length of the table is called a bucket or an array of integers [30].

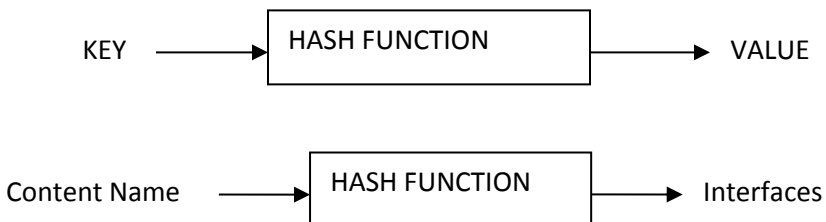
For example to store a set of n items, assuming each item is an element of some finite set U called the universe; U denote the size of the universe, which is just the number of items in U . a hash table is an array $T[1\dots m]$, where m is smaller than U , the hash function is a function $h: U \rightarrow \{0, 1 \dots m-1\}$ that hashes each possible item in U to a slot in the hash table. E.g. we say x item hashes to the $T[h(x)]$. One of the hashing is collision which occurs when two items x and y hashes to the same hash value: $h(x)=h(y)$, but this problem can be resolve using different methods; the most common one are; by chaining and open address.

- a. Chaining: This method is used to resolve a collision between two items with the same key in a table slot, in which they are put in a linked list so that each entry $T[i]$ is not just a single item but rather (a pointer to) a to the next items that hash to $T[i]$. Each node in the list composed of a data and reference pointing to the next node [30].

- b. Open addressing: this method resolve the collision by looking elsewhere in the table, so that different hash functions in the sequence always map to different locations in the hash table [30].

3.2.1 PIT Hashing:

The PIT table is experiencing persistence lookup and updating (insertion and deletion) for different data size; some with similar content request either from the same or different interfaces. Therefore, this work consider the content name of the data request in the router’s PIT as the hashing key while the incoming interfaces to be the hashing value; so that the insertion, searching and deletion of PIT entries can be perform based on client’s key to locate the position of their matched data or when the life time of interest expires before the returning of equivalent data packet, the router can simply use its mapping key to remove the associated interface for that interest. This operation can also be performing when the forwarded interest brings back a data packet. For example the following formula is always applied to both operations in relation to PIT.



Based on the above explanation and the PIT table below; containing five entries from different consumers with their corresponding interfaces and encoded value of each name components to simplify and minimize the size of bit. Looking at the table some of the interests were requested from the same interface, therefore we can allocate a single linked list to store a number of interfaces requesting the same content name; which can greatly save the PIT size and give good looking arrangement of the table, instead of shifting the table slots. The PIT size minimization can be achieve with good hash function method, that can distribute various interest key into the array slots of the table containing their incoming

interface, as such this work employed the use of modulo division to compute the hashing key of a given content for the location of an item in the array slot during the searching, insertion and deletion.

The PIT design in NDN is facing different direction from different contributors in order to achieve better performance in keeping records of unsatisfied interest before the return of data. But always the structure of PIT entries is also matters, like in [31] the PIT design focus on storing fingerprint instead of string using hash-base technique

3.3 Name Prefix component Aggregation:

The idea of this mapping mechanism is based on Prefix Tree which is an ordered tree data structure that is used to store a dynamic set or associative array where the keys are usually strings. The purpose of this method involve the resolution of NDN name prefix components with common prefixes to be aggregated, so that retrieval of interest packet can simplify where each component is prefix match to the parent tree as in the table 3.1 below. The purpose of this aggregation in the PIT table is to reduce the memory consumption and to avoid interest duplicate. Because the length of NDN name component is unbounded, larger than even the bounded 32.bit variable length of IP addresses, therefore storing them directly will increase the cost of storing content name that will increase the computational cost and network overhead. Comparing the named data [/com/parc.documentation/ndn-article/paper-presentation](http://com/parc.documentation/ndn-article/paper-presentation) may require more space and processing time than 198.100.10.0 IP address.

The implementation of the proposed PIT designing where carried out in order to show interest name prefix condensation into a reasonable data structure like hash table, but the name component need to be resolve into an integer value in order to speed up the overall operation of packet exchange within the router environment, the diagram illustrate a name prefix in order to derive a content name relationship among the network clients.

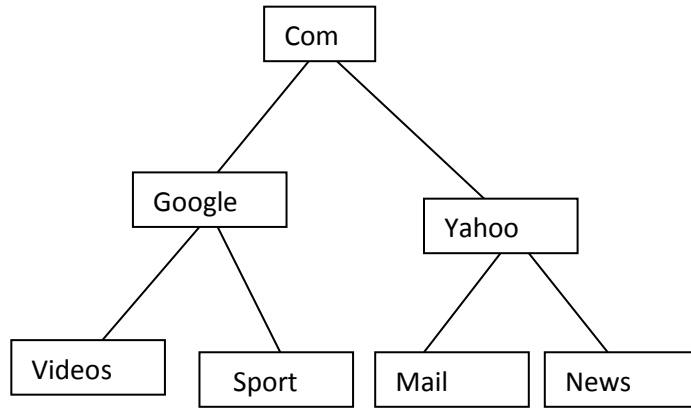


Figure 3.1: Name Component prefixes Tree

Name Prefixes	Interfaces
/google.com/	2
/google.com/videos/	5
/google.com/sport/	2
/yahoo.com/	3
/yahoo.com/mail	1
/yahoo.com/news	4

Table 3.1: Name prefixes Component Table

The above table can be deformed to a simple table after being aggregated based on the name components similarities from the name prefix tree illustrated in figure 3.1 above. The PIT uses this aggregation technique to suppress unexpected attacks from non-legitimate users [14], because one of the challenges facing NDN architecture involved the introduction of fake interest from unknown router. Comparing the two tables, the 1st, 2nd, and 3rd entries are aggregated to their parent node and the same to the last three entries. This shows that multiple NDN names can share a common components name prefix, e.g. /google.com/, /google.com/videos/, /google.com/sport/ share a common prefix /google.com/ and /yahoo.com/, /yahoo.com/mail/, and /yahoo.com/news/ also share a common prefix /yahoo.com/.

Name Prefixes	Interfaces
/google.com/	2
/yahoo.com/	3

Table 3.2: Aggregated Name prefixes Component

3.4.1 Numerology:

This technique is base on the study of symbolism of numbers through which strings of characters can be converted into a single integer value. The procedure works as follows [32];

1. Adding up the digits in name component (if more than one digit)
2. Adding up the digits of the second components up to the last component (if more than one digit)
3. Add up the answers from (1), and (2) above.

With all the above calculations, we keep adding until we end up with a single digit, or an 11 or 22 (which are special cases in numerology, known as "Master Numbers").

1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	

Using figure 3.1 the following integer value can be derive;

For /google.com/: $7+6+6+7+3+5+3+6+4=47=4+7=11$

For /google.com/videos/: $11+4+9+4+5+6+1=39=3+9=12=1+2=3$

For /google.com/sport/: $11+1+7+9+2=31=3+1=4$

.....

For/yahoo.com/mail/: $7+1+8+6+6+3+6+4+4+1+9+3=58=5+8=13=1+3=4$

3.4 PIT Design Model:

The overall NDN routing roughly depends on the performance of PIT, while the PIT operation is largely based on the number of incoming and outgoing interest/data packet, which can be affected by the packet size. Although the aggregation of duplicate requests and expiration time given to each interest in the table reduced high interest queue and avoidance of legitimate users from being satisfied as a result of over population of pended request.

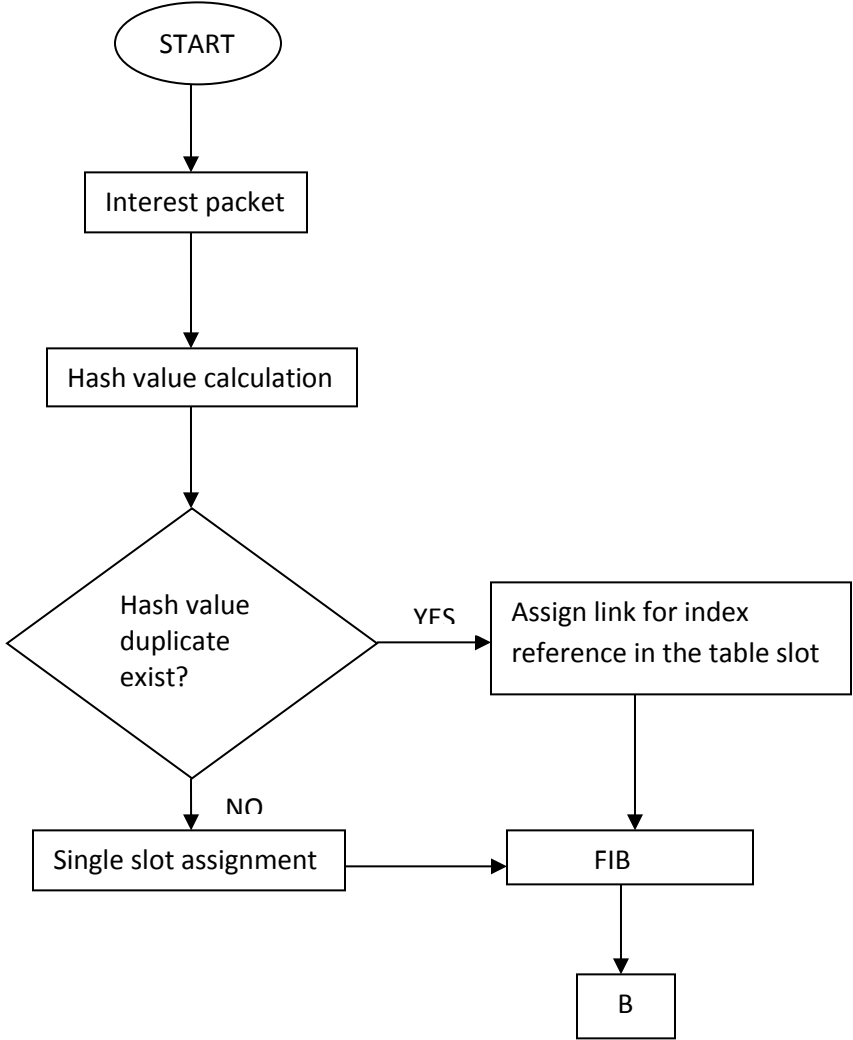


Figure 3.3: Slot Allocation of PIT Entry

Therefore, PIT hashing is implemented in relation to the type of request received by the router, it first compute the hashing key of every incoming interest/data packet and then make a decision on the calculated value before deciding the proper direction of the searching key for insertion, lookup and deletion; that is either the key is located along a chain as a result of having same key with other item or it is in separate slot. The detail of the PIT hashing procedure is explained in the algorithm below representing the depicted flowchart in figure 3.3 & 3.4.

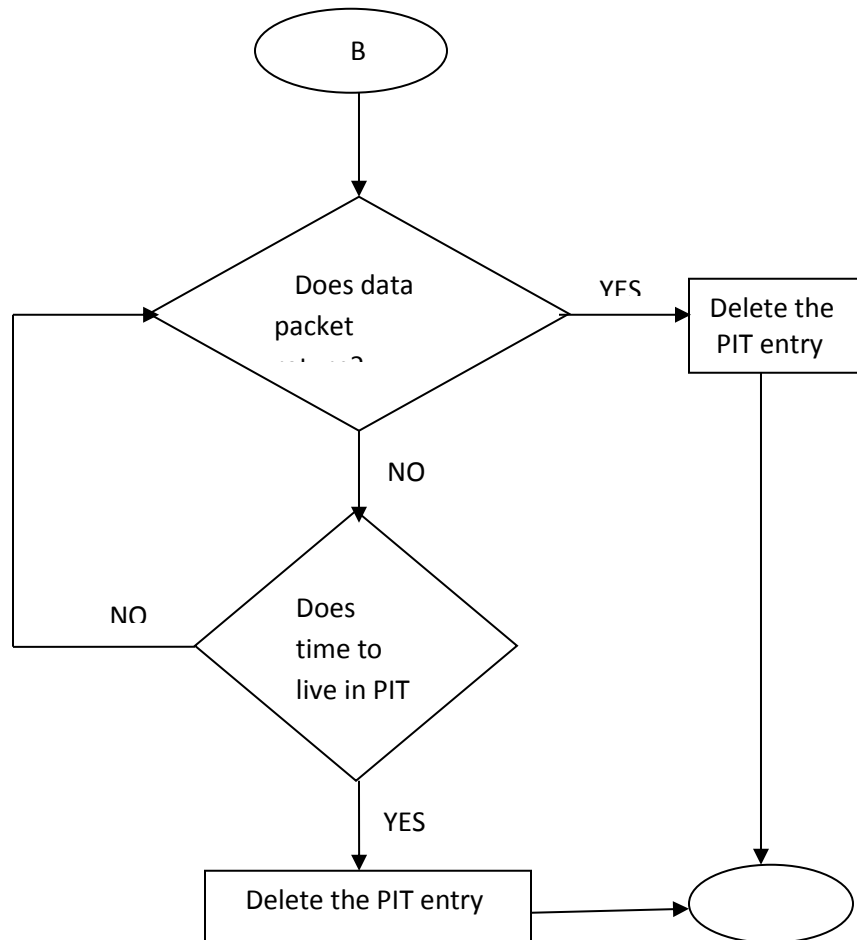


Figure 3.4: deleting Item from the PIT Table

3.4.1 One Way Hash Function:

Is a cryptographic algorithm that turns an arbitrary-length input into a fixed length binary value and this transformation is one way that is given a hash value. It is statistically infeasible to come up with a document that would hash to value [33]. The widely used hash algorithms are MD5 which produces a 128-bit, SHA produces 160-bit and SHA-256 produces 256-bit hash, e.t.c.

The purpose of employing this algorithm is to allow the conversion of string of message into a routable binary/integer value, in which Message Digest -5 (MD5) is implemented. MD5 takes input message of arbitrary length and produces output of 128 bit length, it does not normally produce two messages with the same message digest, which compressed message in a secure manner usually in cryptosystem, but can be applied to hierarchical relationship of storing data dynamically as in the case of PIT table.

CHAPTER FOUR

4.1 Experiment Evaluation:

Based on the proposed system, the size of the PIT table does not need to be large provided the keys are of integer type which can allow chaining of keys with the same value in the table array size. Suppose that the interest packets arriving at a particular router in a network are assigned a memory location storing the set of all pending request in a universal set $U : \{0, 1, \dots, m-1\}$ representing their incoming interfaces. The keys are assumed to have their array values ranging from $T\{0, \dots, m-1\}$ and the period of the operation during searching, insertion and deletion follows a dynamic running time depending on the execution step involved in the table component arrangement.

4.2 Implementation:

Using the prefix trie of name component illustrated in the previous chapter (figure 3.1 & 3.2) various name can easily be derived with simplest unit bit of integer form that will reduced the memory consumption of the name prefixes. The table below was formed from the tree component key calculation and slot allocation in the array index.

Name prefix Component-Integer Resolution	
Name Prefixes	Integer Value
/google.com/	11
/google.com/videos/	3
/google.com/sport/	4
/yahoo.com/	11
/yahoo.com/mail	4

Table 4.1: PIT Numerology Prefix Tree-Integer Value

4.3 Slot Allocation:

Considering the table above and the proposed PIT design; hundreds or thousands of item can accommodate an array of small size for a given computable integer size. For example if there exist an array of size five {0.1.2.3.4} to assign a memory location of PIT entries numbered with above mention converted integer value from numerology calculation and their respective interfaces

sh Key Computation Using %		
Items	Modulo Division	Index Number
11	%	0
3	%	1
4	%	2
11	%	3
4	%	4

4.4 Simulation Setup:

The proposed mechanism was carried out using java programming tool to illustrate the internal operation of NDN router's PIT. I first assumed the nature of PIT table storing converted strings component to an arbitrary integer value to test the effectiveness of how to design PIT to retrieve content more easily. Secondly, each interest with their corresponding interfaces in form of string type is tested.

4.5 Simulation synopses:

- **String Mapping:** The first simulation result storing a set of interest packet in form of string is tested and the experimental snapshot is shown below from java eclipse development environment, while the code used is presented in appendix A. The code executed is working both for insertion and deletion of items from the table.

➤

Interest name/incoming interface

/google.com/videos/: face02

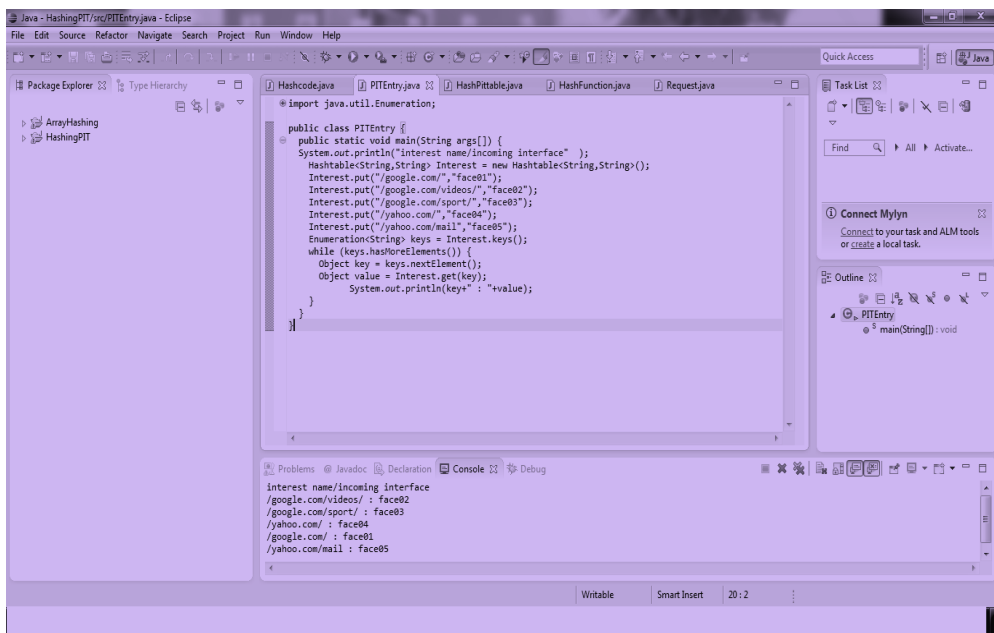
/google.com/sport/: face03

/yahoo.com/: face04

/google.com/: face01

/yahoo.com/mail: face05

String Mapping



```
import java.util.Enumeration;

public class PITEntry {
    public static void main(String args[]) {
        System.out.println("Interest name/incoming interface" );
        Hashtable<String,String> Interest = new Hashtable<String,String>();
        Interest.put("google.com/","face01");
        Interest.put("google.com/videos/","face02");
        Interest.put("google.com/sport/","face03");
        Interest.put("yahoo.com/","face04");
        Interest.put("yahoo.com/mail","face05");
        Enumeration<String> keys = Interest.keys();
        while (keys.hasMoreElements()) {
            Object key = keys.nextElement();
            Object value = Interest.get(key);
            System.out.println(key* : "value);
        }
    }
}
```

Interest name/incoming interface
/google.com/videos/ : face02
/google.com/sport/ : face03
/yahoo.com/ : face04
/google.com/ : face01
/yahoo.com/mail : face05

- **Integer Hashing:** The second PIT hashing is based on modulo divisions which allow the storing of interest with the same key in a single slot with chaining, the snapshot of the experiment is shown below while the code executed is in the appendix B.

Modulus Index= 0 for value 5

Modulus Index= 1 for value 11

Modulus Index= 3 for value 13

Modulus Index= 4 for value 14

Modulus Index= 0 for value 15

Number of Collisions = 2

Modulus Index= 2 for value 17

Number of Collisions = 5

Modulus Index= 2 for value 22

Number of Collisions = 9

Modulus Index= 2 for value 77

Number of Collisions = 14

Modulus Index= 4 for value 24

Number of Collisions = 18

Modulus Index= 0 for value 30

Number of Collisions = 27

Modulus Index= 0 for value 45

Number of Collisions = 37

Modulus Index= 3 for value 63

Number of Collisions = 45

Modulus Index= 1 for value 66

Number of Collisions = 56

Modulus Index= 0 for value 100

Number of Collisions = 69

Modulus Index= 0 for value 150

Number of Collisions = 83

Modulus Index= 0 for value 185

Number of Collisions = 98

Modulus Index= 0 for value 90

Number of Collisions = 114

Modulus Index= 0 for value 50

Number of Collisions = 131

Modulus Index= 4 for value 49

Number of Collisions = 145

Modulus Index= 0 for value 170

Number of Collisions = 164

Sum of Collisions = 164

0 1 2 3 4 5 6 7 8 9

5 11 15 13 14 17 22 77 24 30

10 11 12 13 14 15 16 17 18 19

45 63 66 100 150 185 90 50 49 170

20 21 22 23 24 25 26 27 28 29

4.6 Analysis of the PIT hashing:

The performance of hash table is measure using; good hash function, that is why I consider the implementation of two-by-two cases (that is direct simple hashing for raw string component of PIT entries and indirect modulo division for integer value array implementation).

The string hashing design a hash table of interest packet without having similar request showing flat structure in which the insertion/deletion and lookup of data could be done with constant running time[$O(1)$].

While the integer value hashing involved putting item with similar hashing key, and their running time is always [$O(n)$], because several key will be mapped to the same value.

- Worst case: The keys of hashing PIT entry must some time hold common index in the table, because many consumers may request the same data; as such we assign a single slot to reduce the memory size, as implemented using array of integer value resolution.
- Average case: The average complexity of the entire PIT hashing is always assumed to be the constant load factor (ratio of slot searching per array size) plus the number of iteration to locate an item in the table.
- Best case: The best of PIT hashing is always constant when the prepared operation is successful else it follows slot verification for a given hash key.

4.7 PIT as a Database ENGINE:

A database is a collection of information that is organized, so that it can easily be accessed, managed and update. It keeps records which are subject to change. The nature of PIT operation can be considered and implemented just like normal database; therefore I created a simple database table in Netbeans integrated environment for java, using the following fields to hold NDN content name;

Database name: Pending_Interest_Table

Table name: Conten_Name

ID
Interest_Packets
Incoming_Interfaces

The ID keyword in the table served as the primary key of the database that can be used to identify table records. It contains a unique value for each row and the value most not be null. The table created for this database has the following characteristics; as shown in snapshot below

For ID field

Key: checked

Index: checked

Null: Unchecked

Unique: checked

Column Name: ID

Data Type: Integer

For Interest_Packet field

Key: Unchecked

Index: Unchecked

Null: Unchecked

Unique: Unchecked

Column Name: Interest_Packet

Data Type: VARCHAR

Size: 50

For Incoming_Interfaces field;

Key: Unchecked

Index: Unchecked

Null: Unchecked

Unique: Unchecked

Column Name: Incoming_Interfaces

Data Type: VARCHAR

Size: 20

Empty Database table for Content_Name records

Table name: Content_Name

Key	Index	Null	Unique	Column name	Data type	Size
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ID	INTEGER	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interest_Packet	VARCHAR	50
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Incoming_Interfaces	VARCHAR	20

Buttons: Add column, Edit, Remove, Move Up, Move Down, OK, Cancel, Help

- The table below shows the records being inserted

select * from YAKUBU.CONT... 88

Page Size: 20 | Total Rows: 7 Page: 1 of 1 | Matching Rows:

#	ID	INTEREST_PACKET	INCOMING_INTERFACES
1		1 /ndn/netlab.cs.memphis.edu/	face03, face05
2		2 /google.com/videos/	face015
3		3 /ccnx.org/ndnForwardingDaemon	face010
4		4 /yahoo.com/mail	face011
5		5 /www.yasar.com/	face020
6		6 google.com/videos/	face017
7		7 /google.com/sport/	face022

2:1 | 1

CHAPTER FIVE

5.1 Conclusion:

The core architectural planned of NDN design as the proposed future internet architecture is achieved via the routing mechanism of the entire communication system which differs in so many different ways with the current IP network. The key features includes the replacement of IP addressing with content name in search of data, securing the data itself from the original source with no reference to communication media- so that client doesn't need to care about the data source or the channel through which the request can be sent or the retrieval of the data, multipath forwarding and delivery of data without looping, as a result of built-in caching along the transmission path depending on the routers' cache policy.

The new architecture was initiated to flamboyantly overcome the major challenges associated with the current internet operation due to the emergent of new and sophisticated applications in the computing environment which largely depends on internet. Today' internet is facing serious challenges with the current flash smart phones and social media together with enhancement for online businesses taking place everywhere via the internet. However, this shows an obvious need of content distribution and delivery within the internet, so that popular and unpopular content can be retrieve with less latency and minimize bandwidth consumption. Therefore, NDN has come with absolute solution of the above listed problems in IP network.

The NDN routing and caching are centralized within the control plane (data plane) containing three tables: the PIT, FIB, and CS. The PIT is a unique table which has no equivalent in today's internet, where routing effort is concentrated, because it keeps all the pending interest for some period of time and delivers the return matching data in return or when the time to live of an interest expire it drop it, while the FIB can aggregating request with the same name prefixes and CS serve an instance request if there is matching upon request.

However, considering the centrality features of NDN architectural objectives, it requires high speed memory (this is not of part this work) and efficient data structured for managing the routing tables more especially the PIT. Because it experience dynamic lookup, insertion and deletion of content prefixes for accommodating and adapting persistent searching for every interest that has no equivalent data in the CS, PIT initiate and finalize the routing processing of every incoming interest. Therefore, its speed is highly considered for efficiency and reliability in routing and caching of data.

This thesis had really introduced a sound solution toward the optimistic design of PIT which can support better performance of routing activities, by employing the hashing techniques in the entire PIT operation in order to maintain compatibility with memory size in storing and retrieving of content. Therefore, to conclude my work the effectiveness of routing and caching policy in NDN not only give emphasis to routing protocol and caching policy but also their implementing container.

The NDN architecture is a very interesting technology but requires coordinative management to provide its objectives goal in the area of networking due to high memory usage and complex computation, not only the issue of additional cost to maintain the architecture but the structuring of the database management of packets exchange between clients and producers. The major solutions to this includes the use of supportive routing protocol like NLSR which is already in place with all required component providing the routing of name prefixes and Link State Advertisement throughout the network, good caching mechanism and good management of the routing tables using any efficient data structure including the one proposed in this work which is supported by many researchers like in paper [20 & 21] with different perspective in performance measurements and others; like the use of bloom filters which also provide better management system [12].

Therefore, in this thesis various design techniques have been implemented base on hash table as a data structure to speed up the routing of packet within the

routing plane, so that the legitimate NDN network users will not have confuse with the size router memory, although it requires fast memory operation together with structure of how data is stored and retrieve within the data plane

REFERENCES

- [1] **Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, Rebecca L. Braynard** “Networking Named Data”; Palo Alto Research Center. New York-USA: 2009.
- [2] **Sam Halabi, Danny McPherson**. “Internet Routing Architecture” Second Edition: 2001.
- [3] **Jun Li, Hao Wu, Bin Liu, Jianyun Lu, Yiw-Tsinghua** University Beijing, Xiu Wang-State University of New York, Yanyong Zhang, Lijun Dong-Rutgers University North Brunswick; NJ. Popularity-Driven Coordinated Caching in Named Data Networking-Presented in the eight ACM/IEEE Symposium on Architecture for Networking and Communication System-New York: October 29, 2012.
- [4] **Yusung Kim Ikjun Yeom**. “Performance analysis of in-network caching for content-centric networking”: 2013.
- [5] **Wei Koong Chai, Dilliang He, Ioannis Psaras, George Pavlou**. “Cache “less for more” in information-centric networks”. Computer communication journal; volume 36 issues 7: 2013.
- [6] **Wang Guo-ging, Huang Tao, Liu Jiang, Chen Jian-ya, Liu Yun-jie**. “Modeling in-network caching and bandwidth sharing performance in information-centric networking”. Journal of china Universities of posts and telecommunications: 2013.
- [7] **AKM Mahmudul Hoque, Syed Obaid Amin, Adam Alyayan, Lang Wang**. “Named data link state routing protocol”University of Memphis, Beichuan Zhang- University of Arizona,Lixia Zhang- University of California, Los Angeles.
- [8] **Huichen Dai, Janyuan Lu, Yi Wang, Bin Liu**. “A two layer intra-domain routing scheme for named data networking”. Globecom-Next generation networking and internet Symposium; Anachein CA: 2012.

- [9] **Lang Wang, AKM Mahmudul Hoque, Cheng Yi, Adam Alyyan, Beichuan Zhang.** “Open shortest path first routing protocol for named data networking”. Technical report NDN-0003, July 25, 2012.
- [10] **Mengjun Xie, Indra Widjaja, Haining Wang.** “Enhancing cache robustness for content-centric networking”. Proceedings IEEEINFOCOM; 2012.
- [11] **Tabias Lauinger, Nikolaos Laoustaris, Pablo, Rodrigue, Thorsten Strufe, Ernst Biersack, Engin Kirda.** “Privacy risk in named data networking: what is the cost of performance?” ACM SIGCOMM computer communication; volume 42 issue 5; 2012.
- [12] **Haowei Yuan and Patrick Crowley-washington university st. Louis Missouri.** “Scalable pending interest table design from principles to practice”. IEEE INFOCOM 2014-IEEE conference on computer communications.
- [13] **Mauro Conti, Paolo Gasti, Marco Teoli.** “A lightweight mechanism for detection of cache pollution attacks in named data networking”. Computer network journal; volume 57 issue 16; 2013.
- [14] **Paolo Gasti, Gene Tsudik, Ersin Uzun, Lixia Zhang.** “DoS & DDoS in named data networking”. arxiv:1208.0952V2[CS.NI] 7 Aug. 2012.
- [15] **Albert Compagno, Mauro Conti, Paolo Gasti, Gene Tsudik.** “Poseidon: mitigating interest flooding DDoS attacks in named data networking”. arxiv:1303.4823V3[CS NI] 1 Aug 2013.
- [16] **Mohan Li.** “recent advances in named data caching and routing”. Technical report Dec. 20, 2013.

- [17] **Michele Tortelli, Luigi Alfredo Grieco, Gennaro Boggia.** “Performance assessment of routing strategies in named data networking”. GTTI 2013 session on telecommunication network.
- [18] **Cheng Yi, Jerald Abraham, Alexander Afunasyev, Lang Wand, Beichuan Zhang, Lixia Zhang.** “On role of routing in named data networking”. Technical report NDN-0016-2013.
- [19] **Cheng Yi, Alexander Afanasyev, Ilya Moissenko, Lan Wang, Beichuan Zhang, Lixia Zhang.** “A case for stateful forwarding plane”. Computer communications journal –volume 36, issue 7: 1 April 2013.
- [20] **Yi Wang, Yuan Zu, Ting Zhang, Kunyang Peng, Qunfeng Dong, Bin Liu, Wei Meng, Huichen Dai, Xin Tian, Zhonghu Xu, Hao Wu, Di Yang.** University of Science and Technology of China. “Wire Speed Name Lookup”. A GPU-based Approach-2013.
- [21] **Huichen Dai, Bin Liu, Yan hen, Yi Wang.** “On pending interest table in named data networking”. ACM/IEEE symposium on architecture for networking and communication systems: New York- 2012.
- [22] **Andrea Passarella.** “A survey on content-centric technologies for the current internet”. CDN and P2P solutions computer communication journal volume 35 issue1-2011.
- [23] **James F. Kurose.** University of Massachusetts Amherst, Keith W. Ross-polytechnic institute of NYU. “A topdown approach on computer networking”.fifth edition: 2010.
- [24] **Aaron Bulchunas.** “CCNA study guide”-V2.62 -2012.
- [25] **S.V Nagaraj** Chennai India. “Web caching and its applications”. Springer’s eBook store: 2004.

- [26] **John Dille, Bruce Maggs, Jay Parikh, Harald Prokop, Bill Weihl, Akamai** technology. “Globally distributed content delivery”. IEEE internet computing: Sept/Oct 2002.
- [27] **jianli Pan, Subharthi Paul, and Raj Jain**-Washington University. “A survey of the research on future internet architecture”. IEEE communication magazine July, 2014.
- [28] http://en.wikipedia.org/wiki/Static_random-access_memory.
- [29] <http://searchnetworking.techtarget.com/definition/TCAM-ternary-content-addressable-memory>.
- [30] MIT_ Lecture 10 Hashing and Amortization, Supplemental reading in CLRS: Chapter 11; Chapter 17 intro; Section 17.1.
- [31] **Haowei Yuan and Patrick Crowley**-Computer Science and Engineering Washington University. ”Scalable Pending Interest Table Design:From Principles to Practice”. IEEE INFOCOM 2014 - IEEE Conference on Computer Communications.
- [32] <http://www.astrology.com/numerology>.
- [33] http://en.wikipedia.org/wiki/Cryptographic_hash_function

APPENDIX A

A string of interest packet being added to PIT table using hashing;

```
import java.util.Enumeration;

import java.util.Hashtable;

public class PITEntry {

    public static void main(String args[]) {

        System.out.println("interest name/incoming interface" );

        Hashtable<String,String> Interest = new Hashtable<String,String>();

        Interest.put("/google.com/", "face01");

        Interest.put("/google.com/videos/", "face02");

        Interest.put("/google.com/sport/", "face03");

        Interest.put("/yahoo.com/", "face04");

        Interest.put("/yahoo.com/mail", "face05");

        Enumeration<String> keys = Interest.keys();

        while (keys.hasMoreElements()) {

            Object key = keys.nextElement();

            Object value = Interest.get(key);

            System.out.println(key+" : "+value);

        }

    }

}
```

```
}
```

APPENDIX B:

The following codes populate PIT table with Interest packet as Array with linked list to chained those entries with the same hashing key

```
import java.util.Arrays;
```

```
public class HashFunction {
```

```
    String[] theArray;
```

```
    int arraySize;
```

```
    int itemsInArray = 0;
```

```
    public static void main(String[] args) {
```

```
        HashFunction theFunc = new HashFunction(20); // creating table size
```

```
        String[] elementsToAdd2 = { "5", "11", "13", "14", "15", "17", "22", "77",  
            "24", "30", "45", "63", "66", "100", "150", "185", "90", "50", "49",  
            "170"};
```

```
        theFunc.hashFunction2(elementsToAdd2, theFunc.theArray);
```

```
        theFunc.displayTheStack();
```

```
    }
```

```
    // Simple Hash Function that puts values in the same
```

```
    // index that matches their value
```

```

public void hashFunction1(String[] stringsForArray, String[] theArray) {

    for (int n = 0; n < stringsForArray.length; n++) {

        String newElementVal = stringsForArray[n];

        theArray[Integer.parseInt(newElementVal)] = newElementVal;

    }

}

```

```

public void hashFunction2(String[] stringsForArray, String[] theArray) {

    int sumOfCollisions = 0;
    float averageOfCollisions = 0;
    int numberOfCollisions = 0;

    for (int n = 0; n < stringsForArray.length; n++) {

        String newElementVal = stringsForArray[n];

        // Create an index to store the value in by taking
        // the modulus

        int arrayIndex = Integer.parseInt(newElementVal) % 5;

        System.out.println("Modulus Index= " + arrayIndex + " for value "
            + newElementVal);
    }
}

```



```

// Cycle through the array until we find an empty space

while (theArray[arrayIndex] != "-1") {
    ++arrayIndex;
    numberOfCollisions++;

    //System.out.println("Collision Try " + arrayIndex + " Instead");
    //System.out.println("Number of Collisions = " +
numberOfCollisions);
    // If we get to the end of the array go back to index 0

    arrayIndex %= arraySize;
}

if (numberOfCollisions > 0)
{
    System.out.println("Number of Collisions = " +
numberOfCollisions);
}

theArray[arrayIndex] = newElementVal;

}

sumOfCollisions += numberOfCollisions;

averageOfCollisions = sumOfCollisions / 20;

```

```

        System.out.println("Sum of Collisions = " + sumOfCollisions);

        System.out.println("Average of Collisions = " +
averageOfCollisions);

    }

    // Returns the value stored in the Hash Table

    public String findKey(String key) {

        // Find the keys original hash key
        int arrayIndexHash = Integer.parseInt(key) % 5;

        while (theArray[arrayIndexHash] != "-1") {

            if (theArray[arrayIndexHash] == key) {

                // Found the key so return it
                System.out.println(key + " was found in index "
+ arrayIndexHash);

                return theArray[arrayIndexHash];

            }

            // Look in the next index

            ++arrayIndexHash;

            // If we get to the end of the array go back to index 0

```

```
        arrayIndexHash %= arraySize;

    }

    // Couldn't locate the key

    return null;

}
```

```
HashFunction(int size) {

    arraySize = size;

    theArray = new String[size];

    Arrays.fill(theArray, "-1");

}
```

```
public void displayTheStack() {

    int increment = 0;

    for (int m = 0; m < 3; m++) {

        increment += 10;

        for (int n = 0; n < 170; n++)

            System.out.print("-");

    }

}
```

```
System.out.println();
```

```
for (int n = increment - 10; n < increment; n++) {
```

```
    System.out.format("| %3s " + " ", n);
```

```
}
```

```
System.out.println("|");
```

```
for (int n = 0; n < 71; n++)
```

```
    System.out.print("-");
```

```
System.out.println();
```

```
for (int n = increment - 10; n < increment; n++) {
```

```
    System.out.print(String.format("| %3s " + " ", theArray[n]));
```

```
}
```

```
System.out.println("|");
```

```
for (int n = 0; n < 71; n++)
```

```
    System.out.print("-");
```

```
System.out.println();    }  }
```