INTERNATIONAL RESEARCH JOURNAL ON ADVANCED SCIENCE HUB
e-ISSN : 2582 - 4376
Open Access
RSP SCIENCE HUB
(The Hub of Research Ideas)
Available online at www.rspsciencehub.com

Special Issue of First International Conference on Innovations in Engineering Sciences (ICIES 2020)

Analysis of NDN Messenger Using Chrono Sync Library

Prof.Vijaya Eligar[1], Dr. Nalini Iyer[2], Jeet Shah[3], K H Manjunath[4]

[1]Assistant Professor, School of Electronics and Communication Engineering, KLE Technological University, Hubli, India

[2]Professor and HOS, School of Electronics and Communication Engineering, KLE Technological University, Hubli, India

[3]Student, School of Electronics and Communication Engineering, KLE Technological University, Hubli, India

[4]Student, School of Electronics and Communication Engineering, KLE Technological University, Hubli, India

Abstract

Named Data Networking (NDN) is an Internet Architecture which is designed to network computing devices by naming data instead of the conventional data containers used in Internet Protocols today.The main purpose is to evolve the Internet's host-based packet delivery model. The Named Data Networking service uses the name as the object to retrieve the data in the most secure, reliable and efficient way. Chrono chat is based on NDN (Named Data Networking) which is the future, depicting the ways messages will be sent and received. Chrono chat, being a multi-user chat room, runs over an NDN server. In this paper we propose the functioning of the Chrono Sync Library to achieve synchronization which aids in the transmission of information in real time implementation of the application functioning over the Named Data Network. This paper proposes an implementation of chrono sync library to provide a better alternative using C++, Java, JavaScript and python languages for Linux and Windows based operating systems.

Keywords: NDN, Chrono-sync, Messenger, Chat Application, Networking

1. Introduction

There is significant evidence of increasing drawbacks of CDN (Content Distribution Networks) due to the failure rate in data transmission and slower transmission speeds thereby creating a need for a better method for data delivery which being more secure and faster thereby resulting in a creation of a better model being NDN (Named Data Networking) NDN is changing the way IT networks are controlled and managed. The NDN perspective relies on using the name of the object to retrieve information in the most secure way possible. Unlike the traditional method NDN (Named Data Networking) does not require middle boxes like TLS as TLS does not have the functionality to encrypt at the data level but encrypts at the channel level NDN takes the security of data transmission to another level by encrypting at the data level securing the data from the User rather than securing the channel like the conventional method. In NDN all the work is done in a single software stack enabling it to run everywhere making it leaps and bounds better than the traditional methods.

The NDN (Named Data Networking) is one of the new developing Internet structures that can

leverage the strengths and weaknesses of the current local communication and communication structure to naturally adapt to emerging forms of communication. The project learns about the technical challenges that must be addressed in order to validate the NDN as the future of the Internet. The Major objective of this project being creation of a better alternative to enable secure and faster transmission of data which led to the creation of Chrono Chat for various OS systems and not restraining only to android or Windows or Linux. The motivation to create Chrono Chat was to create an instant message app for various operating systems over NDN. Chrono Chat makes it possible to create a cha troom which is created by chrono chat clients within which multiple users can join and communicate effectively. Chrono Chat makes use of the Chrono Sync library to create a multi user serverless chat room. NDN (Named Data Networking) is one of the newest developing internet architectures that can capitalize on the strengths and weaknesses of the current, point-to-point communication architecture in order to naturally accommodate emerging patterns of communication. The project studies the technical challenges that must be addressed to validate NDN as a future Internet architecture. The following papers contribute to the literature survey upon which our development and contribution to the following subject have been built. Ashlesh Gawande, Jeremy Clark, Damian Coomes, Lan Wang† et al. (2018) [1,2] proposed a paper on Decentralized and Secure Multimedia Sharing Application over Named Data Networking it concluded that In this work, they explored the design and implementation of a fully decentralized application over NDN. Their experience demonstrates that it is feasible to develop such an application, but it requires new approaches to designing the application namespace, establishing trust, discovering potential friends, and performing pub-sub

1.1 Literature Survey

Sulfath PM et al. (2006) [2] proposed a paper on Named Data Networking (NDN): Fundamental Concepts Benefits it concluded that the Internet has been a huge success but today's Internet hourglass architecture is no longer a good match to its primary use. NDN is the new networking paradigm proposing the evolution from today's host centric network architecture to data centric network architecture. It generalizes the Internet architecture by replacing the focus on where endpoint addresses of hosts with what identifier of the content that users and applications care about. NDN is aimed to show

- Communication is more secure.
- Infrastructure is more efficiently utilized.
- Applications are simpler.
- New things are possible.

It operates on top of the Internet protocols and it doesn't need a complete replacement of the current Internet architecture.

In [3] Yunita Noor Rohmah, Dodi Wisaksono Sudiharto, Anton Herutomo et al. have proposed a technique for Performance Comparison of Forwarding Mechanism between IPv4 and Named Data Networking (NDN). The paper concludes that the NDN routing through its usage of TCP protocol simulation choses a path for sending the packets in line with the smallest RTT. The results indicate that the RTT estimation has an important role in determining the NDN network's selected route. This is different from IP routing mechanism where the selected route is chosen by performing based on the routing table. This can be attributed to the NDN router maintaining its paths depending on the information which are brought by packets across it. The IP network has a better performance than the NDN network in a normal condition which is no hijacked node. The NDN network performs several drop packets which is different from the IP network. It can be attributed to the drop packets happening on several nodes. Whenever there is a hijacked node, the IP network fails to provide a solution. This is so because the IP-based network cannot get SYN packets to the destination node. Therefore, the destination cannot understand if there is a node which executes a request to it. Then this condition generates no data packets that can be performed in the simulation as a request by the sender. The NDN network still can be performed even if there is the hijacked node. There are still drop packets, however, they are not data packets. They are only the Interest

packets which are dropped by facing the existence of the hijacked node. All data packets on the NDN network simulation have received by the destination

An Overview of Security Support in Named Data Networking, IEEE Communications Magazine, November [4] [2018] presents an overview of the security mechanisms in the Named Data Networking (NDN) architecture that have been developed over the past several years. NDN changes the network communication model from the delivery of packets between hosts identified by IP addresses to the retrieval of named and secured data packets. Consequently, NDN also fundamentally changes the approaches to network security Opportunities and Challenges for Named Data Networking to Increase the Agility of Military Coalitions Proceedings of Workshop on Distributed Analytics Infrastructure and Algorithms for Multiorganization Federations (DAIS), [5] [2017] The fundamental aim of this paper is to position the opportunities and challenges for adopting Named Data Networking (NDN)

Schematizing Trust in Named Data Networking, ACM Information Centric Networking Conference [6-8] [2015] talks about Securing communication in network applications involves many complex tasks that can be daunting even for security experts. The Named Data Networking (NDN) architecture builds data authentication into the network layer by requiring all applications to sign and authenticate every data packet. The purpose of the network is to allow reliable communication between all the nodes that are connected to the network. IP is a method to achieve this and is currently the mostly widely adopted method to achieve communication between the nodes connected to the network.

1.2 Limitations of Traditional Networks

IP uses point to point packet delivery, the network here is a collection of links between the nodes. To enable data transmission from one node to another the IP chains multiple links and nodes for packets to move between one point to another. Packets travel through the chain created with all the nodes in between being operational.

IP being the most used protocol has its drawbacks the major of which being

1.Link Fails.
2.Node Movement
3.Change in connectivity

To overcome these drawbacks NDN (Named Data Networking was used in the project to achieve reliable message transfer.

2. Methodology

2.1 Functional Block Diagram

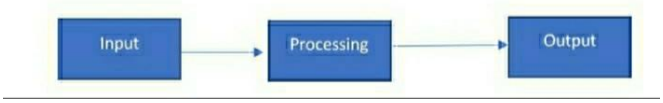

Fig 1. Functional Block

2.2 NDN Architecture and Functionality

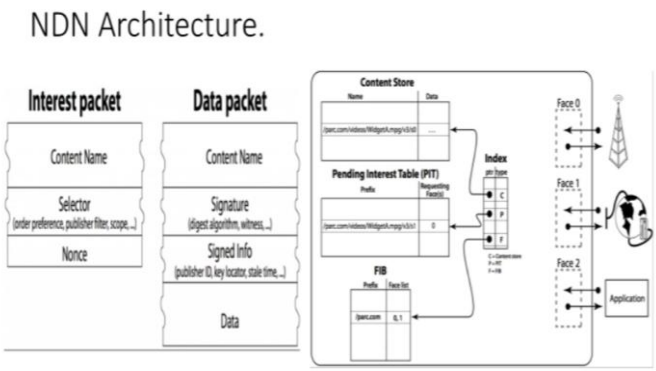

Fig 2. NDN Architecture

2.3 Block Diagram

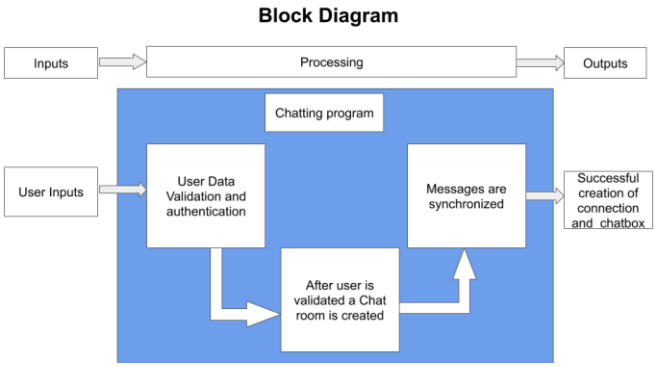

Fig 3. Functional Block Diagram

2.4 Input

The first step after the user launches the application Input from the user is the first step. Input from the User consists of a screen name, a

chatroom name, and an NDN prefix for chatroom data.

2.5 Processing

●Chrono Chat consists of Activities and a Service as its primary components.

●The central component is the component that is launched when the user first opens Chrono Chat and which generally coordinates tasks is Main Activity.

●When first launched, Main Activity will discover that no login information has been set, and launch Login Activity to request it from the user.

●When the user signs in, Login Activity will return the information to Main Activity, which will then display the chatroom screen.

●Network tasks are handled by the Chrono Chat Service component.

●Communication between Main Activity and Chrono Chat Service consists of Intents, which are the usual means of inter process (and "inter-component") communication.
●When Main Activity signals Chrono Chat-Service to send a message, the service starts up, establishes a connection to the chatroom, and publishes the data via Chrono Sync.

●It remains running as a foreground service to prevent the OS from killing the Chrono Chat process whenever the user is not interacting with it.

●The service maintains its connection to the chatroom and relays any received messages to Main Activity via Intents, while also keeping track of the chatroom roster and periodically sending "heartbeat" messages to the chatroom.

2.6 Output

●After login, the user enters the chatroom.

●After Chrono Chat connects to the chatroom, other connected clients will appear to "join" the chat.

●The user can send messages using the field at the bottom of the chatroom window.

●The toolbar at the top may be used to leave the chatroom, view the roster, or quit the app.

●Chrono Chat displays an ongoing notification while connected to a chatroom This helps prevent the OS from terminating the app while it is running in the background.

●The app will also alert the user via a notification when messages are received while the app is running in the background.

2.6Specifications and System Architecture

The specifications and the technical details of the proposed systems are discussed in this part.

The project performed is a software-based project and the specifications required include a Linux Based Operating System, NS3 Tool, Python, JavaScript and C++ languages and modules and also required is the chrono sync library.

2.7 The Inner Operations of How the application functionsis explained as follows

●Chrono Chat consists of Activities and a Service as its primary components.

●The central component is the component that is launched when the user first opens Chrono Chat and which generally coordinates tasks is Main Activity.

●When first launched, Main Activity will discover that no login information has been set, and launch Login Activity to request it from the user.

●When the user signs in, Login Activity will return the information to Main Activity, which will then display the chatroom screen.

●Network tasks are handled by the Chrono Chat

Service component.

●Communication between Main Activity and Chrono Chat Service consists of Intents, which are the usual means of inter process (and "inter-component") communication.

●When Main Activity signals Chrono Chat-Service to send a message, the service starts up, establishes a connection to the chatroom, and publishes the data via Chrono Sync.

●It remains running as a foreground service to prevent the OS from killing the Chrono Chat process whenever the user is not interacting with it.

●The service maintains its connection to the chatroom and relays any received messages to Main Activity via Intents, while also keeping track of the chatroom roster and periodically sending "heartbeat" messages to the chatroom.

```
class Chat(object):
    def __init__(self, screenName, chatRoom, hubPrefix, face, keyChain,
      certificateName):
        self._screenName = screenName
        self._chatRoom = chatRoom
        self._face = face
        self._keyChain = keyChain
        self._certificateName = certificateName

        self._messageCache = [] # of CachedMessage
        self._roster = [] # of str
        self._maxMessageCacheLength = 100
        self._isRecoverySyncState = True
        self._syncLifetime = 5000.0 # milliseconds

        # This should only be called once, so get the random string here.
        self._chatPrefix = Name(hubPrefix).append(self._chatRoom).append(
          self._getRandomString())
        session = int(round(self.getNowMilliseconds() / 1000.0))
        self._userName = self._screenName + str(session)

        self._sync = ChronoSync2013(
           self._sendInterest, self._initial, self._chatPrefix,
           Name("/ndn/broadcast/ChronoChat-0.3").append(self._chatRoom), session,
           face, keyChain, certificateName, self._syncLifetime,
           onRegisterFailed)

        face.registerPrefix(self._chatPrefix, self._onInterest, onRegisterFailed)
```

Fig 4. A virtual chat room can also be created using python and code written to do so

```
jeetshah194@ubuntu:~/Desktop/ndnSIM/PyNDN2-master/examples$ nfd start
NFD version 0.7.0-16-gcb27d916 starting
```

Fig 5. The following command is used to start NFD (NDN Forwarding Daemon)

3. Results

The Observation and the functioning of the application are made on various tools which include NS3 Tool, Python, C++, Java and JavaScript and for various platforms like Windows, Android and Linux.

The simulation of the environment occurs in the NS3 tool and the chatroom is created using Python and its basic functionality is tested using the same, further development of the application is done in Java and the Graphical User Interface for the following is created using JavaScript. Fig. 7 shows the raw implementation of Chrono Chat using Python and Fig. 8 and Fig. 9 shows the finished application with its respective GUI which acts as the end product for final users.

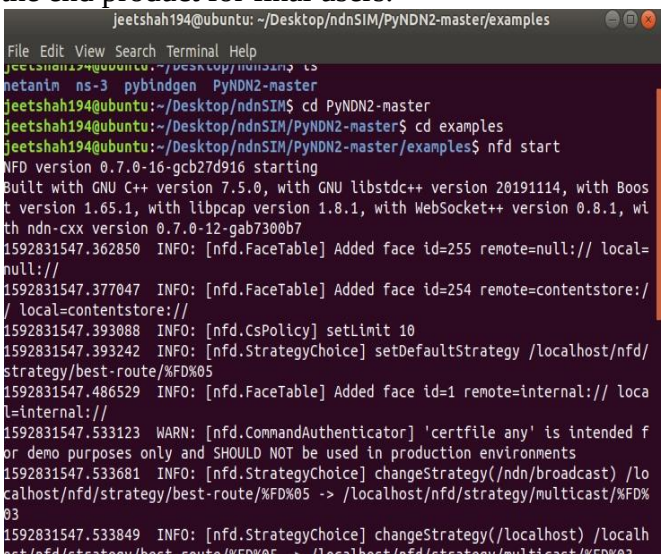

Fig 6. The following is the emulation of starting NFD to run as a background service

The user is able to join the chat room and send messages successfully. And the messages are synchronized in real-time to reduce the time delay for the communication to occur.

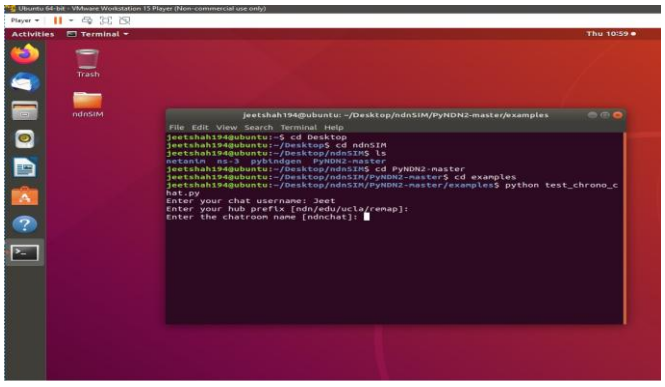

Fig 7. Implementation of Chrono Chat using Python

Conclusion

Secure Message Transmission and receiving is one of the most important aspects in today's life but

the current methods for message delivery not only put the user at risk but also have slower data delivery speeds when compared to Chrono Chat which uses NDN.

The Proposed Design for the Chat room achieves excellent performance with a modest number of users within the chatroom providing good service and legitimate data delivery between the users using the chatroom.

The Chatroom makes it possible to deliver data at faster speeds and with high security. Which is the main purpose we want to achieve so that the user doesn't get annoyed by the delays is sending messages and giving them the highest data security as much as possible.

NDN, an emerging technology implemented in Chrono Chat limits the user to access only a single chat room every use and does not provide the functionality to switch between different chat rooms.

Conclusion we drew from this project is that NDN based networks are more secure and faster compared to the conventional network and thus are the future internet architecture to be used to send and receive information and media.

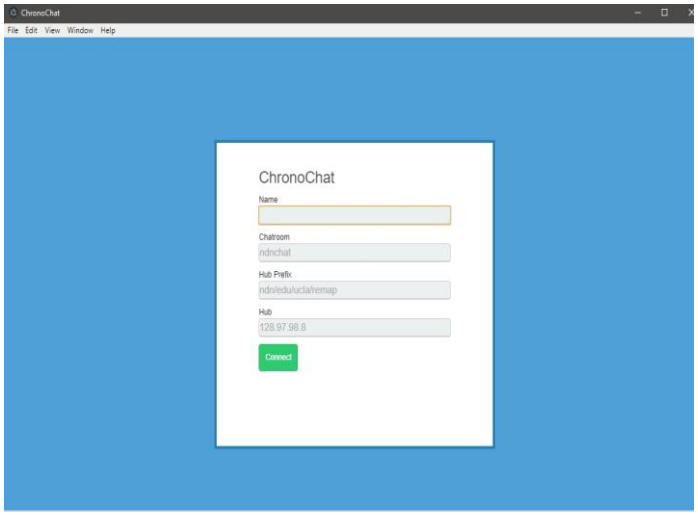

Fig 8. The GUI of the login page of Chrono Chat

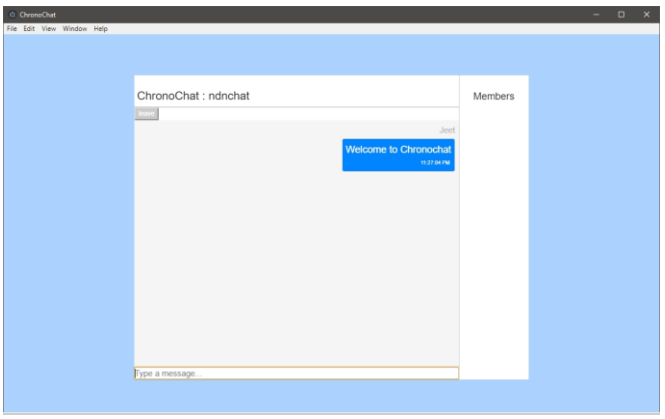

Fig 9. Chatroom creation and exchange of messages

Future Scope

Although NDN promises faster Data Delivery time and better security than the traditional methods it does not offer the ability to switch between chat rooms nor does it offer the user the ability to communicate with a person existing in a different chat room until the user quits the current chatroom and joins an another chatroom which consists of the node that the user is trying to communicate with.

A Further Attempt to diversify the abilities of the chatroom can be made enabling the user to seamlessly send data to users existing on different chat rooms.

We can also enable the delivery of Media Transfer which can include jpeg Images and video files or documents within the chatroom which is currently only restricted to messages. With the current trend of having customization within the application we can also add customization features which may include changing the theme of the application and various other aspects of the application.

The application can also be modified for peer-to-peer file sharing app or file sharing from one node to many nodes i.e., broadcasting using WIFI. Hence reduce the reliance on internet connectivity to share files and reduce the usage of internet bandwidth.

To increase security of the application we may also use end-to-end encryption techniques while transferring data. By encrypting the data the we can make the data readable for only the sender and the receiver and avoid and unwanted data misuse

Acknowledgement
It gives us immense pleasure to take this opportunity to thank the department of Electronics and Communication Engineering and our college for enlightening us with the Minor project course. We take this opportunity to thank Dr. Ashok Shettar, vice-chancellor, KLE Technological University, for providing or giving an opportunity to carry-out the undertaken Minor project. Our special thanks to our guide, Prof. Vijaya Eligar, of the Electronics and Communication Engineering department, for guiding us throughout our project and making it a grand success. We are also thankful to the technical staff and academic staff of the Electronics and Communication Engineering department for supporting us during the making of our project. We also thank our institution, KLE Technological University, for giving us this opportunity to test and improve our skills. We would like to thank all our seniors, family members and friends who have been a great support for us. Finally, we would like to thank everyone who has directly or indirectly contributed towards the completion of our project.

References

Journals

[1] M. Zhang, V. Lehman and L. Wang, ”Scalable name-based data synchronization for named data networking,” IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, 2017, pp. 1-9,doi: 10.1109/INFOCOM.2017.8057193.

[2]] Z. Zhu and A. Afanasyev, ”Let’s ChronoSync: Decentralized dataset state synchronization in Named Data Networking,” 2013 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, 2013, pp. 1-10, doi: 10.1109/ICNP.2013.6733578.

[3] Z. Zhang et al., ”An Overview of Security Support in Named Data Networking,” in IEEE Communications Magazine, vol. 56, no. 11, pp. 62-68, November 2018, doi: 10.1109/MCOM.2018.1701147.

[4] Y. N. Rohmah, D. W. Sudiharto and A. Herutomo, ”The performance comparison of forwarding mechanism between IPv4 and Named Data Networking (NDN). Case study: A node compromised by the prefix hijack,” 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, 2017, pp. 302-306, doi: 10.1109/ICSITech.2017.8257129

[5] Gawande, Ashlesh Clark, Jeremy Coomes, Damian Wang, Lan. (2019). Decentralized and Secure Multimedia Sharing Application over Named Data Networking. 19-29. 10.1145/3357150.3357402.

[6] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang and L. Zhang, ”A Brief Introduction to Named Data Networking,” MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1-6, doi: 10.1109/MILCOM.2018.8599682.

[7] M. A. Hail, ”IoT-NDN: An IoT Architecture via Named Data Networking (NDN),” 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), BALI, Indonesia, 2019, pp. 74-80, doi: 10.1109/ICIAICT.2019.8784859.

[8] M. M. S. Soniya and K. Kumar, ”A survey on named data networking,” 2015 2nd International Conference on Electronics and Communication Systems (ICECS).