NDN-Hippo: a streaming media system over Named Data Networking

Kai Lei*, Yunbo Xun*, Longyu Yu*, Tao Chen*, Zhi Tan†
Institute of Big Data Technologies
*Shenzhen Key Lab for Cloud Computing Technology & Applications (SPCCTA),
School of Electronics and Computer Engineering, Peking University
†GuangDong Poya Information & Technology Co.,LTD, GUANGZHOU 511400, P.R.CHINA
*leik@pkusz.edu.cn, †tanz@poya.com.cn

***Abstract*—In order to demonstrate the NDN network's suitability in streaming media applications, we design and implement a streaming media system called NDN-Hippo in NDN. It is based on a similar streaming media application in IP network, named Hippo. NDN-Hippo, taking advantages of a few features of NDN network, is implemented on a simulation platform called ndnSIM, which is a module in NS-3 and frequently used as a test platform for NDN applications. This paper introduces the design of modules in our system, including the detailed description of three servers, PreServer, SourceServer, DirServer and a client. For experiments, we implement NDN-Hippo and compare the performance with another media system BitTorrentVideo, which is also implemented in NS-3. The result demonstrates the capability of NDN to support streaming media.**

***Index Terms*—Named Data Networking, Streaming Media, NDNSim**

I. Introduction

With the high speed development of the Internet technology and computer applications, the current Internet, which is designed based on IP addresses of reachable end hosts, has undergone dramatic changes and faces many challenges and obstacles in supporting content-oriented applicaitons. So many unprecedented applications present new demands for the network architecture. As one of the future Internet architecture proposals, Named Data Networking (NDN) [1]–[4] was developed to solve problems. NDN uses names of content, rather than IP addresses of end hosts , to route Data packets. It transfers the focus of network routing from searching for device with specific IP-Address to retrieving specific content, which can make the network search the content spontaneously without the participant of the application layer.

In NDN, data name replaces the role of IP address and becomes the unique naming identification. Data name in NDN can initiate connection between different nodes, making the request and receiving of data more naturally. NDN network has only two types of packet: Interest and Data. Data communications in NDN are driven by the consumers who express the request Interests of corresponding Data. The core forwarding component of NDN consists of three data structures: Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). To request the data of interest, a consumer first sends out an Interest packet, carrying a name that uniquely identifies the desired data. If receiving the Interest, a NDN router stores the incoming interface of the Interest in their local PIT tables where each entry contains the name of the Interest and a set of incoming interfaces, then looks up the corresponding name in its FIB table, and finally forwords the Interest. Once the Interest arrived at a node with the requested data, a Data packet is issued back. The Data packet has the same name of the requestig Interest and the corresponding content, and was signed by the Data producer. With the signature, the consumer could verify the authenticity of the Data packet. When the Data packet arrived at a NDN router, the router finds the PIT entry and forwards the Data packet to all the interfaces listed in that entry. Then the router deltes this PIT entry of the Interest and caches the Data in the Content Store is cached for serving the later Interests with the same name. The Content Store runs a replacement policy for effective in-network caching.

Many researchers have proposed the idea of combining the NDN and streaming media. They also give a prediction about the performance. But few of them implement this idea and measure the performance.

Hippo [5] is a live streaming media system developed by the Internet Research and Engineering Center at Peking University. It is a Hierarchical P2P streaming media system, and has good scalability and robustness. Three functions, which are live-streaming, video-on-demand, and downloading, are provided and large-scale environment is supported by Hippo. The system structure of Hippo is a hierarchically hybrid one, consisting of four components: Source Manager (SM), Region Manager (RM), Static Peer (Speer), and Dynamic Peer (Dpeer). The top layer of the system is source manager, which manages resources of the whole network, answers the administator's function requests and users' look up requests. The intermediate layer is region manager, which is responsible for maintaining the relationship of resources and nodes in specific region. Speer and Dpeer are the two kinds of nodes of the bottom layer. Speers are the super nodes configured by administator for stationally serving for Dpeers or other super nodes. Dpeers provide other data services of user level.

This paper designs an NDN-based video on-demand streaming media system, named NDN-Hippo. NDN-Hippo is similar to the Hippo and leverage the inherent properties of NDN. But it differs from the existing P2P streaming media system in following aspects: 1) NDN-Hippo transform the traditional Tracker to distributed servers pool and deploy RMs (distributed Tracker) in different region. 2) NDN-Hippo is more capable and efficient to discover and it uses available resources in local network. 3) We design and implement an adaptive scheduling strategy which flexibly combines Greedy and RF algorithm in NDN-Hippo.

To demonstrate NDN-Hippo and evaluate the performance, We implement NDN-Hippo in ndnSIM, and compare it with an existing streaming media system BitTorrentVideo[1] [6] whose structure is similar to Hippo.

The main contributions of this paper are as follows:

- We design NDN-Hippo which is an NDN-Based video on-demand streaming media system.
- We find a more suitable way to combine NDN network and streaming media.
- We implement NDN-Hippo in ndnSIM, compare it with the existing Hippo system, and analyze the benefits that are brought by NDN network.

The remainder of this paper is organized as follows. Section II introduces the system desgin of NDN-Hippo. Section III and Section IV describes the design of the Servers and the Client respectively. Section V evaluates the performance of NDN-Hippo. Section VI discusses related work. In the end, section VII remarks conclusions.

II. The System Design of NDN-Hippo

In this section, we first present the overview of NDN-Hippo architecture according to the various modules of the system. Subsequently, we describe the basic functions of the system design principles and implementation methods respectively.

A. Design Philosophy

NDN-Hippo keeps the original advantages of the Hippo, such as P2P function and distributed deployment. These advantages can ensure the data storing and forwarding functions to be shared to each peer. Moreover, these advantages eliminate the dependence on a situational resource, and help the load balance between different nodes.

NDN-Hippo also tries to take the advantages of NDN. According to the features of NDN, every node becomes a NDN router via NFD and stores video data blocks in Content Store.This can help NDN-Hippo keep data copies all over the system, so that users can always retrieve a block of data from adjacent users. On the other hand, NDN provides server-less service of data source retrieving, which is similar to the pure distributed P2P network. This function makes users retrieve global data periodically without using the central server.

NDN-Hippo does not include the RM server of Hippo, which is deployed as a tracker server to provide data location service. We use the routing mechanism underling the NDN network to complete the locating of data source.

[1]BitTorrentVideo is a streaming media system simulator based on NS-3 and is developed by Hamburg University in Germany

B. Whole Systems Design

NDN-Hippo is a streaming media system based over NDN. It removes the tracker server of the Hippo. The whole system architecture diagram is shown in Fig. 1.

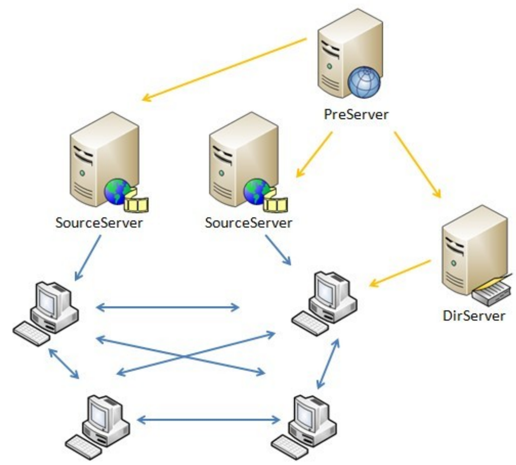

Fig. 1. The system architecture

As shown in Fig. 1, NDN-Hippo system is composed of the following four modules:

- The DirServer, the directory server module, is the directory which stores the on-demand video data in the system. It is responsible for the management of the video directory information about content all over the network. DirServer can offer users the list of videos. In addition, it also provides the interactive access interface for the administrators to manage the content, including the addition, deletion and update operations.
- The PreServer, the pre-process server module, is in charge of pre-process of the video files' publishing and maintenance. It has three functions: 1) process the multimedia data according to the relevant types of video files and the NDN network; 2) manage the online and offline of video; 3) manage the data source server within the global scope.
- The SourceServer, the data source server module, is mainly responsible for storing a series of processed video data and it can respond to the users' data requests according to NDN network.
- The Client, the client module, is installed on the users' nodes. Its states are changed following the operation of the users. In addition, the local content storage in that node will cache part of data.

C. The Rules Of Data Naming

The NDN network uses the hierarchical naming mechanism as one of its design scheme, which is similar to the URL naming mechanism under current IP network. In this paper, we develop a naming scheme based on NDN. The system naming scheme consists of three parts: the fixed prefix, the target prefix and the function part.

As shown in Figure. 2, we define the top-level domain of the data name in NDN-Hippo as */hippo.pku.edu.cn/ndn_vod*.

$$\textbf{/hippo.pku.edu.cn/ndn\_vod}\left\{\begin{array}{l}\textbf{/}\cdots\textbf{/}\\ \textbf{/server\_name/}\end{array}\right\}\textbf{<data or para>}$$

Fig. 2. The system naming scheme.

This is the basic prefix and should be kept as long as possible in general.

The Interest packet may be forwarded to every node in the network. In order to distinguish the Interests packet responders, we divide the Interest packet into two different types and mark them with different target prefix. The first kind of Interest has no specific receiver. And the other kind of Interest has opposite functions.

Each Interest packet has different components for different functions. But they have the same format: "/(function name)/(parameter list)". The parameter list also uses "/" as a separator to facilitate the processing of application. In the client program, users remove the prefixes part first, and then extract the function name field to select the corresponding processing threads.

D. Data preprocessing

The regular streaming media server in IP network usually stores a lot of multimedia files. When a user needs a certain audio or video file, the server will generally separate the file into many pieces at first, and then encapsulate these pieces into a certain format of data packets and send to the client.

In NDN-Hippo system, all the media data will be stored in the Content Store. System administrators will first publish a new film in the PreServer. The file will be streamed to a set of NDN data blocks there. And then the PreServer distributes these blocks around the network according to the interest mechanism and notifies the SourceServer to save these data blocks. Some users will cache the copy of the blocks too. Unlike the system in IP network, NDN-Hippo does not keep the original file. Only the data segments are stored.

III. The Design of the servers

In this Section, we will introduce the design of DirServer and SourceServer in detail. The PreServer has already introduced in Section II.

A. The Desigen of DirServer

1) Function Analysis: The biggest change compared with IP media application is the removal of the tracker server, which greatly helps to reduce the information communication in the network. Due to the characteristics of NDN network, users can get data not only from the source server, but also from other users. The main function of the DirServer is to obtain the request from the PreServer about the online, offline and update of the data. These video resources will be stored by the DirServer and form a large list of programs. The DirServer is in charge of responding to the program list request, but the quantity of this kind of responds is relatively lower than that given by the SourceServer.

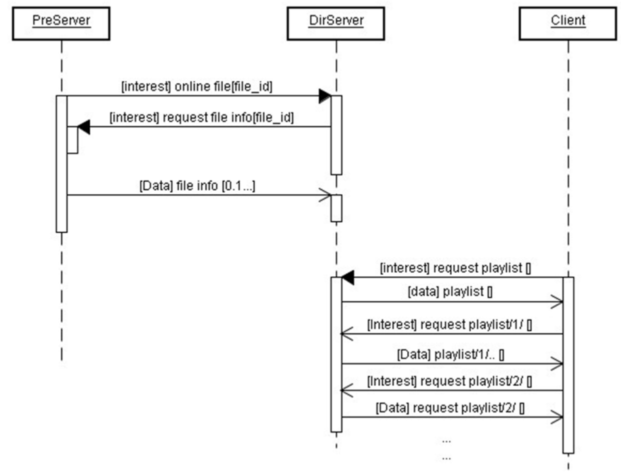

Fig. 3. The Interaction of DirServer

2) Interaction process: When a system administrator publishes a new video file in the PreServer, he will first extract the basic information of this video. This information will be stored in the DirServer. Their interaction steps are shown in Fig. 3 and described as follows:

- The PreServer firstly sends the Interest packet to DirServer, applying for the program modifications (e.g. online, offline, update), and notifying the DirServer that related programs need to be changed at the same time. The request format is (We use "∼" to replace "/hippo.pku.edu.cn/ndn_vod"): *∼/dirsvr/stream_id/consist_id/online* .
- The DirServer discards this Interest after receiving, and then checks whether the related field meets the requirement. After that, the DirServer returns an interest to user, and asks for the changed data formally. The request format is: *∼/presvr/stream_id/consist_id/disp*.
- When PreServer receives the request for program information, it returns appropriate program information. The packet of content may be divided and sent in multiple times.

B. The Design of SourceServer

1) Function Analysis: SourceServer is the server that provides the video data stably. It is long-term deployed in the system , and is responsible for data storage and transmission. It is similar to the Static-Peer in Hippo. Because of the heterogeneity of network, all the nodes which keep some specific data may be offline, or their upload bandwidth may be limited. As a result, these nodes cannot provide data transmission services all the time. Therefore in the streaming media system, in order to ensure the quality of the download speed and playback, it is necessary to deploy a SourceServer.

2) Interaction process: When the administrator publishes a new video file on the PreServer, the PreServer will store the content information in the DirServer. After that, the PreServer will send three formats of data to the SourceServer in order.

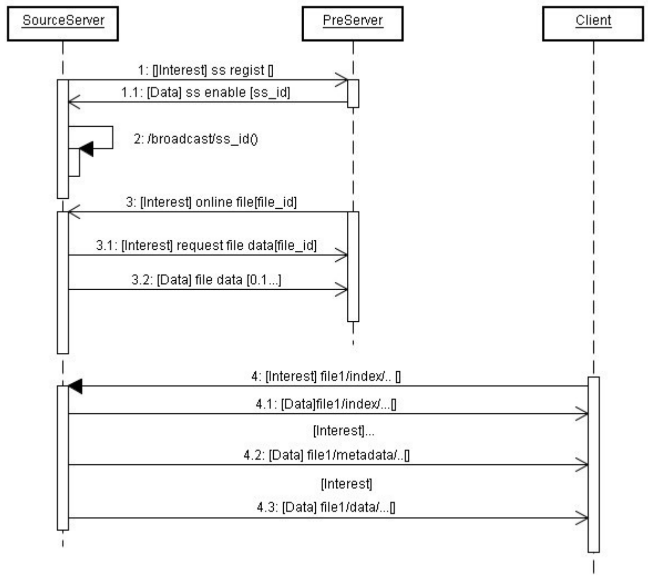

Fig. 4. The Interaction of SourceServer

The interaction between the PreServer and SourceServer is similar to the interaction between PreServer and DirServer, shown in Fig. 4.

Before a SourceServer joins in the system, we need first to register in the PreServer, so that the PreServer can keep a record of the prefix. Then the new registered SourceServer will send an Interest to the PreServer to request some information with prefix: *∼/presvr/ss_regist/*. The returned packet contains the physical information, such as the memory of the node, the performance of CPU and the size of the disk, etc. If the node's storage capacity meets the need, the PreServer will return a data ACK, which means PreServer consents the application. This prefix is *∼/ss_regist/*, with a registration number like */sourceserver_0301/*. This registration number is just like a fixed URL of this node, which can be used to exchange data.

IV. The Design of the Client

In NDN-Hippo, the client first needs to send a directory request to the DirServer to get the directory information Then the client can ask for data according to the obtained information. We design a request strategy of data blocks in the client's scheduling module with consideration of the PULL mechanism of NDN. There are three main functions:

- Provide consecutive data stream for the player.
- Request data from other clients of SourceServer.
- Provide data to other clients.

A. The Data Blocking

Considering the relationships among data block, playback position and the request time, we arrange the states of the data block into four types, and numbered them from zero to three, '0' (not download/not request), '1' (downloaded), '2' (requested) and '3' (lost). The requested block means that the client has sent the Interest but not received data yet. The application will re-transmit the Interest after a certain time. The lost block refers to the data which should be played but have not been downloaded yet. The occurring probability of this kind of block is used as a performance index to evaluate the playback continuity of the system. Among these 4 types, the block states '0' and '2' have higher-priority.

B. The Windows of data

We divide the data window to three parts according to the request priority: reserve window, emergency window and prefetching window in order. They are shown in Fig. 5:

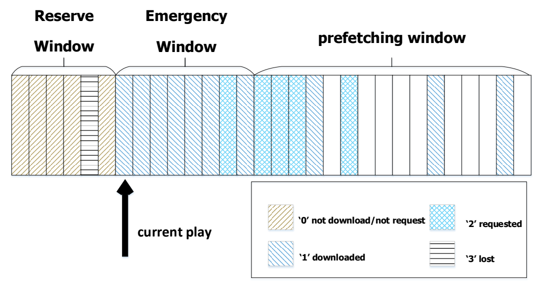

Fig. 5. The Architecture of Windows

The reserve window stores the data already been played, these data are preserved for the user's fallback need. The emergency window stores the data which will be played in a short period of time. This part can ensure the stability of the playback. The client generally needs time to cache data before starting playing. During the playback, if some pieces are lack in the interval, the player will skip these pieces. Thus this results in instability. So the emergency window is important and has the highest priority. Prefetching window is mainly served for the whole system. It can keep the data block uniform distributed in the system. Its priority is the lowest.

C. The Data Scheduling

The scheduling algorithm in NDN is simpler than that in IP. In the original Hippo system, the scheduler is divided into two parts: neighbor node selection and block request scheduling, which is complicated and hard to transplant.

NDN concentrates on content. And its design philosophy is position-independent. So the user only needs to use the name of content to retrieve data, without caring about the location of data. So in the design of the new system, we remove the neighbor node selection module. The NDN router can choose a close face to forward Interest automatically by just sending an Interest. We also do not need to establish connection between nodes as we did in IP, so we do not need to arrange the partnership between nodes.

Based on the local data windows, NDN-Hippo only needs to consider the data related to the local user. There are mainly two factors that should be noted: download speed of the data and bit rate of the on-demand video playback. We use these two parameters to dynamically adjust the scheduling requests.

In order to maintain the smoothness of the video playback, the emergency window is set to 300 pieces, which can support

10 seconds' playing. The number of data blocks requested each time in current node is set to *maxReqCount*, the download speed is set to *curDownSpeed* and the video bit rate is set to *bitrate*.

At first, the user dynamically adjusts the current download speed.Then, we use emergency window and prefetching window to handle data block scheduling, based on the download speed. If the current download speed is slower than the video bit rate, the system only requests the data block in emergency window; if download speed is higher than the bitrate, the system will calculate the required number of data blocks in the emergency window according to the following formullar.

$$urgentQeqCount = log(\frac{a * bitrate}{curDownSpeed}) * maxReqCount$$

In this paper, NDN-Hippo requests the data in the emergency window in order, which is similar to the method in Hippo system; and NDN-Hippo requests the data in prefetching window randomly.

V. Evaluation

In this section, we pesent the evaluation of our proposed NDN-Hippo and compare the performance metirics with BitTorrentVideo's. In our simulation scenario, we use the topology generator BRITE [7] to generate a random network with 5000 nodes at most. We set the update bandwith of nodes in the range of [100kbps,10Mbps] and the link delay in the range of [10ms,100ms]. The number of users in our experiment simulation range from 1 to 4000.

A. The Data Load of The Source Server

The data load refers to the size of the video data that the source server sends to users in a certain time. The data load rate is the percentage of the upload size of the source server in the total network. In the traditional streaming media system in IP, the data load and load ratio are usually used to evaluate the dependence between the system and source server, as well as the contribution that the data transmission system provides in P2P network.This paper simulates different scales. And the results are shown in Fig. 6.

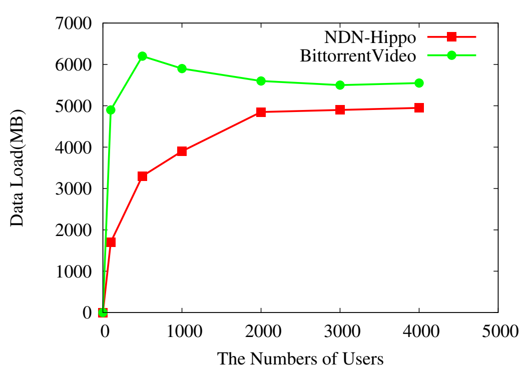

Fig. 6. The Data Load of The Source Server

For NDN-Hippo, when the number of users is between 1 and 1000, the data load grows at a low speed. When the number exceeds 1000, the increment rate becomes stable. The reason of this phenomenon is that when the scale is small, most requests are forwarded to the source server. In this phase, the data load will increase as the number of users grows. But this kind of increment is not linear. The speed of the increase will slow down gradually, and become stable when the number of users reaches a certain value, because of the in-network caching of NDN. In this way, NDN transfers the load from source server to the buffer layer.

We can see in Fig. 7 that, with the increase of the number of users, the load rate of the source server reduces. When the number of users reaches 4000, only 6.77% of content comes from the source server, indicating that the source server only servers 27 nodes in such situation.

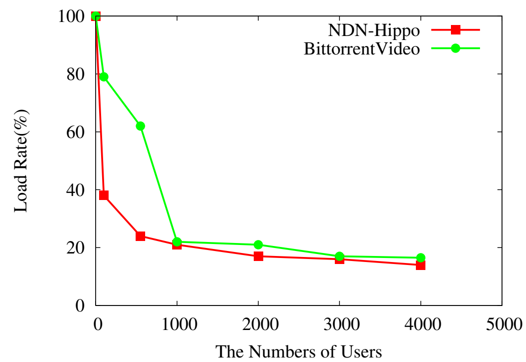

Fig. 7. The Load Rate of the source server

The contrast between NDN-Hippo and BitTorrentVideo shows that the data load rate in NDN-Hippo is always lower than that in BitTorrentVideo. The reasons are: 1) when the scale is small, the peer-list comes from tracker server in BitTorrentVideo cannot satisfy all users' requests. But there may be some nodes which already have the requested data that are not selected as server. 2) rarest first algorithm is used in BitTorrentVideo. The advantage of this algorithm is that users can request the harder packet first, but the system needs to spend more bandwidth in dealing with load balance. 3) when the number of users increases to a certain degree, some parts of the message can't exchange with the tracker in time. As a result, there will be a delay in providing the information.

B. The average delay

The delay time refers to the time difference between requesting data and receiving the response. It is used to measure the distance between the user and the destination. This paper adopts the method in [8] to calculate the delay time. Fig. 8 shows the average delay with different numbers of nodes. As shown in this figure, when the node scale is small, the node mainly requests data from source sever. The average time reflects the transmission delay between node and the server. With the participation of more nodes, the later nodes do not need to visit source server. Instead, they ask for the cached data of nearest nodes. The tendency of the average delay time in NDN-Hippo and BitTorrentVideo are almost the same, but the delay time of NDN-Hippo is shorter. Since NDN network shrinks the distance between users and content.

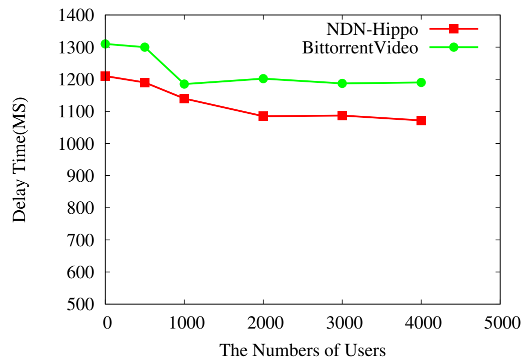

Fig. 8. The Average Delay

C. The Playback continuity

Playback continuity is defined as the proportion of the continuously viewing time to total viewing time (including waiting time) [9]. It reflects the performance of system in a period of time. This paper adopts the strategy in [9]: if the playing is stuck, the player just waits; and if the player does not receive data blocks after a certain time, it just skips certain frames and continues to play.

Fig. 9 shows the users' average playback continuity in different network environments. From an overall perspective, the users' playback continuity is between 93.2% and 95%, which is desirable. When the number of nodes is between 50 and 2000, the continuity increases obviously. The testing result of playback continuity is the same under NDN-Hippo and BittorrentVideo. When the user scale is 0, playback continuity is not defined. So it represents the playback continuity of a single node here.

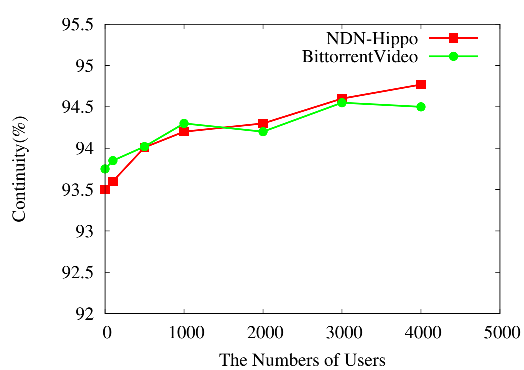

Fig. 9. The User's Playback Continuity

VI. Related Work

UCLA proposed Named Data Networking(NDN), one of Future Internet Architectures in 2009. NDN [4], [10] has achieved great progress in most aspects, such as application development, routing and forwarding strategy, and security. In order to make a better development platform for NDN, UCLA provides NDNx [3] and NFD [11], [12] as the core of the platform which is an open-source software prototype. In addition to attracting more concentration of developers on NDN, the open-source platform encourages developers to improve the functions of NDN. For better research and evaluation in NDN, UCLA provides NDN-based NS-3 simulator, called ndnSIM [13], [14] to design and evaluate applications or modules of NDN protocol in the simulation environment.

Many applications have been designed and implemented in the NDN testbed. These experiments and evaluations present the advantages of NDN in different scenarios. VoCCN [15], which is the first applicaiton over NDN, demonstrates the feasibility and flexibility of NDN. Audio Conference Tool (ACT) [16], which is a distributed audio conference tool over NDN, shows the advantages of NDN in real time conference area. The NDNVideo [17] is a live and prerecorded video streaming system over NDN. The ChronoChat [18] is a server-less multi-user chat system over NDN. The ChronoSync [19] protocol of NDN implemented in this application illustrates the synchronization of distributed content in NDN.

This paper compares functions of Video Streaming and Real-time Conferencing, combines them, and implements in NDN-Hippo.

VII. Conclusions

This paper tries to research the performance of streaming media applications in NDN. So we design and implement an application named NDN-Hippo in a simulator, ndnSIM. This application is similar to Hippo, which is a media system running in IP network for a couple of years. It mainly consists of two kinds of servers, the DirServer and the SourceServer. DirServer provides the program list and the SourceServer provides a stable source. Contrast to Hippo, the architecture of NDN-Hippo is simpler. It removes the tracker server and simplifies the data scheduling strategy. Tracker is important for a P2P application in IP, helps users to find nearest data. But since NDN has the property of in-network caching, NDN-Hippo transfers some works in application layer to transport layer. Thus the data distribution logic is much simpler.

Our work demonstrates the feasibility of the implementation. And we compare the performance of our system with BitTorrentVideo, another simulator based on NS-3. According to the results, NDN-Hippo has a better performance in Source-Server load rate and the network throughput. NDN is suitable for streaming media and large-scale data applications.

Our future work focuses on the deployment of NDN-Hippo, which a streaming media system, in real network and implementation of a SNC control panel [20] in NDN-Hippo.

VIII. Acknowledgement

This project has been financially supported by National Development and Reform Commission Fund of China ([2013]1309), Shenzhen Gov Projects (No: JSGG20140516162852628, JCYJ20130331144541058 and JCYJ20130331144416448).

REFERENCES

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[2] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[3] Project ndnx website. [Online]. Available: https://github.com/named-data/ndnx

[4] Named data networking (ndn) project annual progress summary. [Online]. Available: http://named-data.net/project/annual-progress-summaries/

[5] C. P. L. Y. Kai LEI, Lihua Li, "Hippo: an adaptive, scalable, hierachical p2p live streaming system," *Advanced Materials Research Vols. 989-994 (2014) pp 4556-4561, July, 2014.*, 2014.

[6] The bittorrent client. [Online]. Available: http://www.bittorrent.com/

[7] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*. IEEE, 2001, pp. 346–353.

[8] L. M.Zhang, Q.Zhang and S.Yang, "Understanding the power of pull-based streaming protocol: Can we do better?" in *IEEE Journal on Selected Areas in Communications vol.25,no.9*. ACM, 2007, pp. 1678–1694.

[9] D. M. C. Tom Z. J. Fu and Z. Lei, "Desining qoe experiments to evaluate peer-to-peer streaming applications," *In SPIE Visual Communications and Image Processing*, 2010.

[10] J. B. V. J. k. c. P. C. C. P. L. W. Lixia Zhang, Alexander Afanasyev and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review (CCR), July 2014*, 2014.

[11] Project named data networking forwarding daeeemon(nfd) website.

[12] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, and I. Moiseenko, "Nfd developers guide," NDN, Technical Report NDN-0021, February 2015.

[13] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, October 2012.

[14] Ns-3 based named data networking (ndn) simulator. [Online]. Available: http://ndnsim.net/

[15] V. Jacobson, D. Smetters, N. Briggs, M. Plass, P. Stewart, J. Thornton, and R. Braynard, "Voccn: voice-over content-centric networks," in *Proceedings of the 2009 workshop on Re-architecting the internet*. ACM, 2009, pp. 1–6.

[16] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "Act: audio conference tool over named data networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. ACM, 2011, pp. 68–73.

[17] D. Kulinski and J. Burke, "NDN Video: Live and prerecorded streaming over ndn," NDN, Technical Report NDN-0007, September 2012.

[18] A. A. V. J. Zhenkai Zhu, Chaoyi Bian and L. Zhang, "Chronos: Serverless multi-user chat over ndn," NDN, Technical Report NDN-0008, October 2012.

[19] Z. Zhu and A. Afanasyev, "Let's chronosync: Decentralized dataset state synchronization in named data networking," *Proceeding of the 21st IEEE International Conference on Network Protocols (ICNP 2013), Goettingen, Germany, October 2013.*, 2013.

[20] Y. L. Kai, Lei and W. Jun, "Scalable control panel for media streaming in ndn," *Proceedings of the 1st international conference on Information-centric networking. ACM, 2014.*, 2014.