Centaur: A Evolutionary Design of Hybrid NDN/IP Transport Architecture for Streaming Application

Xiang Zhang*, Lirui Gong*, Yunbo Xun*, Xuewei Piao*, Kai Lei†
†Shenzhen Key Lab for Cloud Computing Technology & Applications (SPCCTA),
School of Electronics and Computer Engineering (SECE)
Peking University, Shenzhen 518055, P.R. China
Email: leik@pkusz.edu.cn

Abstract*—Named Data Networking (NDN), a clean-slate data oriented Internet architecture targeting on replacing IP, brings many potential benefits for content distribution. Real deployment of NDN is crucial to verify this new architecture and promote academic research, but work in this field is at an early stage. Due to the fundamental design paradigm difference between NDN and IP, Deploying NDN as IP overlay causes high overhead and inefficient transmission, typically in streaming applications. Aiming at achieving efficient NDN streaming distribution, this paper proposes a transitional architecture of NDN/IP hybrid network dubbed *Centaur*, which embodies both NDN's smartness, scalability and IP's transmission efficiency and deployment feasibility. In *Centaur*, the upper NDN module acts as the smart head while the lower IP module functions as the powerful feet. The head is intelligent in content retrieval and self-control, while the IP feet are able to transport large amount of media data faster than that if NDN directly overlaying on IP. To evaluate the performance of our proposal, we implement a real streaming prototype in ndnSIM and compare it with both NDN-Hippo and P2P under various experiment scenarios. The result shows that *Centaur* can achieve better load balance with lower overhead, which is close to the performance that ideal NDN can achieve. All of these validate that our proposal is a promising choice for the incremental and compatible deployment of NDN.

I. Introduction

With the increase of content-centered service demands and products, present Internet infrastructure is becoming more and more complex, which increases the cost of the implementation, maintenance and management. There is an ongoing tendency towards weakening location and reducing dependency on centralized servers. Information-Centric Networking (ICN) [1] is a new communication architecture which is compatible with the decentralized nature of the future Internet. ICN shifts the "thin" waist of the Internet from the host-centric paradigm based on perpetual connectivity to content-centric paradigm. Among these ICN proposals, Named Data Networking (NDN) [2] has gained the most attention of the research community, which is designed to facilitate Content Centric Network (CCN) [3].

Presently, the focus of NDN project has moved from architecture design and protocol development to the actual deployment. The deployment research is very important for the proposals of the future Internet. For one thing, prototype network is used to demonstrate the NDN's advantages mentioned above. For another, many studies need to do experiments in large-scale network scenarios, which contributes to optimize the new Internet architecture. However, it's very difficult to deploy ideal NDN prototype, owning to lack of specialized hardware to support its protocol. Inspired by the previous solution of IP deployment, NDN is deployed on Internet infrastructure as IP-overlay, which transfers data relying on IP tunnels. But as the experiment V-A shows, overlay network causes high performance overhead and inefficient transmission. Thus, how to make utilization of the IP infrastructure to deploy practical and efficient NDN is a quite significant research topic. Although, there are already some softwares and tools for NDN deployment, work in this area is still in their infancy.

IP was designed to create a communication network, and it's skilled at end-to-end data deliver. NDN is able to use almost all of the Internet's well-tested engineering properties to solve a much broader range of problems, including not only content distribution but also security and control problems. Given the respective advantages of IP and NDN, we consider a hybrid network to be a promising NDN deployment scheme for streaming application. In addition, hybrid network is more suitable for traditional IP-based applications than pure NDN network.

Accordingly, this paper proposes *Centaur*, a transitional architecture of NDN/IP hybrid network for streaming application. As the core of *Centaur*, the Hybrid Transport Layer consists of two communication modules: NDN transport module and IP transport module. The former is a logical controller, and the latter is a skillful performer. NDN's intrinsic features, such as caching, multi-path and intelligent forwarding, can be leveraged by NDN transport module to design smart data transfer mechanism which is able to provide efficient resource discovery and retrieval. Besides, IP translation module takes charge of transferring clumsy data more steadily. This cooperation promises that our proposal can achieve higher efficiency in content distribution. The Application Layer of *Centaur* introduces two aspects: one is that the application uses Hybrid Transport Layer to achieve data delivery and the other is that control cluster employs NDN to implement exchange of control messages in a straightforward way. Based on *Centaur's* architecture, this paper designs and implements a streaming media application prototype, and compares it with ideal NDN application and P2P system in ndnSIM. Experiments results validate that our proposal is a promising candidate for the

978-1-5090-1496-5/16/$31.00 © 2016 IEEE

incremental and compatible deployment of NDN.

The main contributions of this paper are summarized as follows:

- We shed light on the problems of NDN transmission as IP overlay, and propose a transitional architecture dubbed *Centaur* to solve these problems.
- The concise architecture takes full advantages of NDN and IP to achieve smart content discovery and stable data delivery. Moreover, *Centaur* is more adaptive to distribution of resources, and its data transfer mechanism can be realized easily and efficiently.
- We implement a streaming application prototype, and compare its performance with NDN application and P2P. The result demonstrates that *Centaur's* ability of content distribution is close to that of ideal NDN, and *Centaur* can achieve load balance.
- *Centaur* provides a new perspective on the research of practical NDN deployment, and our proposal makes a contribution to the development of NDN project.

The remainder of this paper is organized as follows. We first introduce the background of NDN and our motivation in the Section the II. Section III presents the architecture of *Centaur* and its data transfer mechanism. Section IV describes the control cluster and application prototype. Simulations and analysis are presented in Section V. Section VI discusses the related work. At last, we conclude the paper and discuss future work in Section VII.

II. BACKGROUND AND MOTIVATION

A. Named Data Networking (NDN)

NDN is a pull-based, content oriented communication model, which is driven by the receiving. NDN adopts a human-readable and hierarchical naming scheme instead of host locations to identify resource. There are two types of packets: `Interest` packet and `Data` packet. Consumer issues Interest to upstream routers or producers to retrieve name-identified Data. The NDN packet forwarding engine includes three key components: Pending Interest Table (PIT), Forwarding Information Base (FIB), and Content Store (CS). PIT is used to record the Interest's incoming face in order to send back the corresponding Data. The next-hop forwarding face(s) is determined by looking up name prefixes in FIB which is populated by name-based routing protocol. CS is employed to cache Data packet to meet the same request in the future.

When a NDN router receives Interest packet from downstream routers or consumers, it firstly checks the Interest's name in the CS. If the corresponding Data is found in the CS, the router will return this Data through the Interest's incoming face, otherwise it will turn to search for the packet name in PIT. If a match is obtained in PIT, the router will insert Interest's incoming face into this matching entry. If not, PIT will create a new entry to identify the Interest, at the same time the router will forward the Interest to outgoing faces based on FIB and forwarding strategy. When NDN router receives a Data packet, if Data's name can be found in PIT, the NDN router will send the Data to all faces that are recorded in this PIT entry and cache Data in CS. Otherwise, the Data packet will be discarded.

B. Motivation And Objective

As a future Internet architecture, NDN is difficult to subvert the current TCP/IP network architecture in a short term. As lacking of specialized hardware to support NDN protocol, NDN need to make deployment with the infrastructure of IP. NDN work as IP-overlay and work together with IP for some time. Therefore, A feasible deployment scheme not only need to consider the architecture design, but also consider the current network factors.

NDN Forwarding Daemon (NFD) is a network forwarder that implements and evolves together with the NDN protocol. At present stage, NFD is installed on IP infrastructure as software router that forwards NDN packets. And NDN is deployed as IP-overlay, which relies on IP channels to transport packets. This deployment scheme causes some problems. Firstly, when NFD use IP network to transfer data, the logical network topology will not be consistent with the actual data transfer path. This leads to an increase in the number of hops. Secondly, as the experiment V-A shows, overlay network brings high overhead, generates invalid load, such as redundant packet header. At last, complete software forwarding is ineffective in terms of transmission, which has a negative impact on the performance of NDN network. In addition, ideal NDN model is incompatible with the traditional IP-based services. Consequently, the motivation of this paper is to reduce the overhead caused by the NDN deployment, and to improve the data delivery performance of overall network. Our proposal is able to achieve smart system control via NDN, and decrease the overhead of data transmission with IP.

III. CENTAUR ARCHITECTURE

In this paper, we concentrate on the issue of reducing NDN deployment cost and inefficiency transmission of streaming application to design a transitional architecture of NDN/IP hybrid network, named *Centaur*. It complies with the requirements listed below-efficiency, scalability, concise architecture, lower transmission overhead, intelligent self-control, supporting adaptive resource selection, and exploiting a pull-based content distribution using multicast and cache. Our proposal leverages NDN to achieve efficient resource discovery and scalable system control. And IP is used to transport data with low overhead. In *Centaur*, hybrid nodes can switch between two kinds of identity: producers and consumers. Producers only focus on the production of their core tasks, rather than maintaining the consumer's status. And consumers retrieve resource info and get media data from multiple producers stably, without polling the availability of resource frequently. Compared with the traditional IP's centralized services, the control cluster module has better scalability, and can automatically select the user nodes with good performance to join the control cluster according to the scale of participants. In

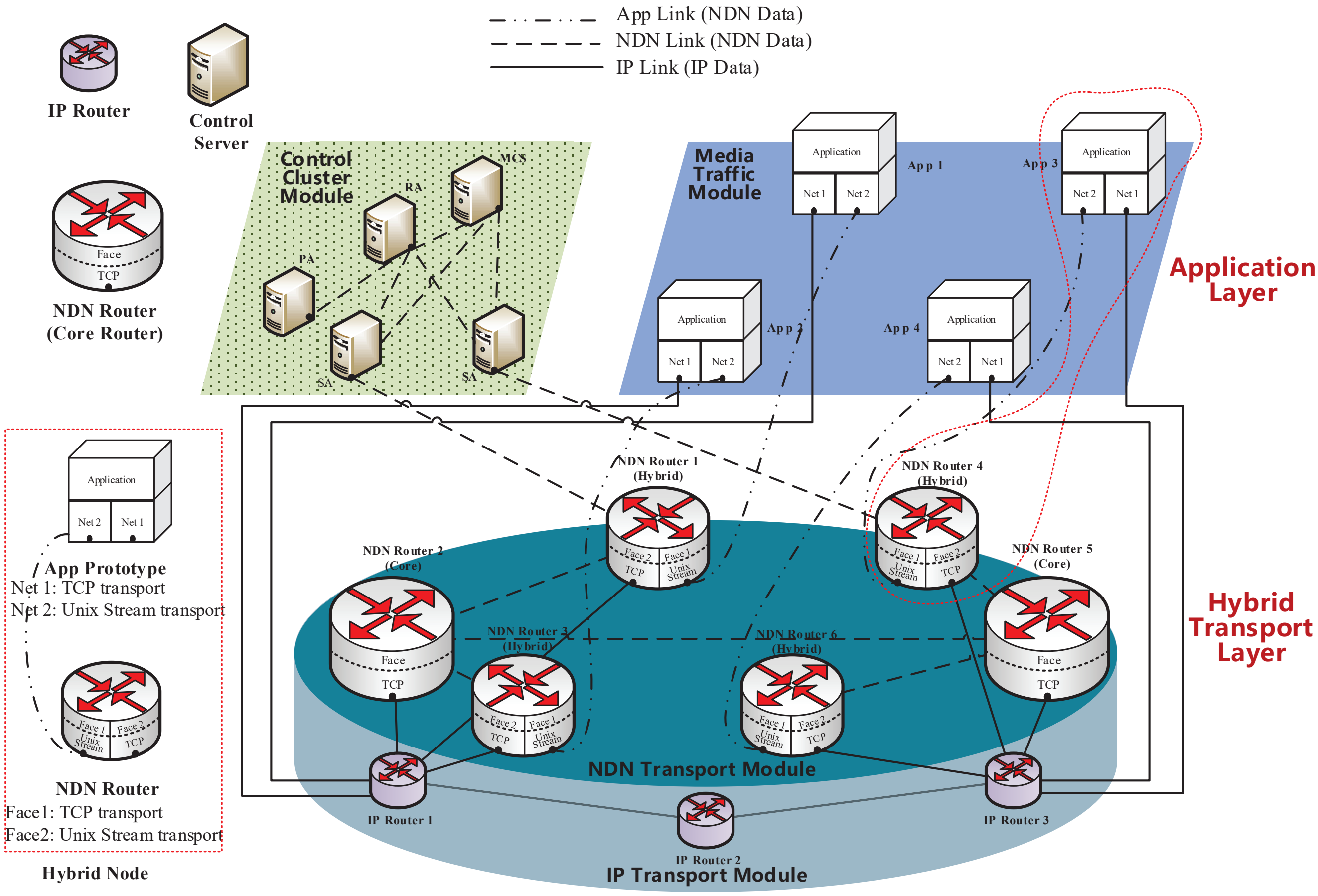

Fig. 1: The Architecture of Centaur.

addition, *Centaur* naturally deals with substantial churn in participant state, allowing a large number of users to frequently join and get out.

A. *Centaur Overview*

Fig. 1 illustrates the *Centaur's* architecture, which mainly includes Hybrid Transport Layer and Application Layer. As the core of *Centaur*, Hybrid Transport Layer takes charge of efficient content distribution and resource discovery. We devise a data transfer mechanism for Hybrid Transport Layer to cope with the distinction of network topology between NDN and IP. The Application Layer composes of control cluster and media traffic modules. We utilize the advantages of NDN to design a secure and scalable control cluster, which cooperates with Hybrid Transport Layer to exchange control message. Besides, the media traffic module demonstrates how to implement applications on the top of Hybrid Transport Layer.

Centaur has four kinds of nodes: NDN Core Router, IP Router, Control Server and Hybrid Node which can use both IP protocol and NDN protocol to communicate. Hybrid Nodes contain application prototype and NDN software router, and chain-dotted line shows the Unix Stream channel between these two components. In addition, we use the dotted line to represent NDN link and the solid line to represent IP link. Then, we introduce the *Centaur*'s Hybrid Transport Layer in detail. And Section IV will describe the Application Layer.

B. *Hybrid Transport Layer*

Hybrid Transport Layer consists of two units, NDN transport module and IP transport module, which Transport data cooperatively with low overhead and smart control. Currently, NFD is installed on IP infrastructure to implement NDN protocol, and NDN is deployed as IP overlay. As shown in Fig. 1, NDN Routers employ NFD to send NDN packets to IP Routers through an interface, and NDN communication relies on IP tunnel. This leads to high deployment overhead and inefficient transmission. Thus, Hybrid Transport Layer is designed to transport light-weight info and heavy-weight media data separately.

The smart *Centaur's* brain is NDN transport module, which transports light-weight info including resource info and control message. NDN transport module is employed to realize efficient resource discovery and smart system control. The NDN's advantages in content distribution, robust security and flexible control, simplify the implementation of these functions, and improve the system scalability. IP transport module works as the skillful *Centaur's* body, and it only needs to cope with end-to-end data delivery, without considering resource selection and system control. IP transport module makes utilization of IP network to transfer heavy-weight data, which is called resource data. In this way, media data delivery don't generate invalid flow, and the transmission is efficient. We will introduce the data transmission mechanism below.

C. Data Transfer Mechanism

This section illustrates the hybrid data transfer mechanism, which combines NDN's smartness with IP's powerful delivery to handle content distribution. This cooperation promises efficient resource discovery and low overhead. The Algorithm 1 briefly describes the data transfer mechanism, which constructs two transmission channels. As is shown in Fig. 2, App1 is the Consumer/Client and App2 work as the Producer/Server. And the process that App1 obtains resource from App2, is divided into three steps.

- Through NDN transport module, Consumer/Client sends Interest packet to locate the Producer/Server in which the resources are stored. NDN's multi-path and in-network cache are leveraged to search resource efficiently.
- When a Producer/Server receives an Interest packet, it checks resources for this request. If there is any resource matched, the Producer/Server returns back the Data packet that contains resources info and the corresponding Producer/Server IP-address description.
- The Consumer/Client analyzes the returned data packet, and then gets resource data from the Producer/Server via IP transport module.

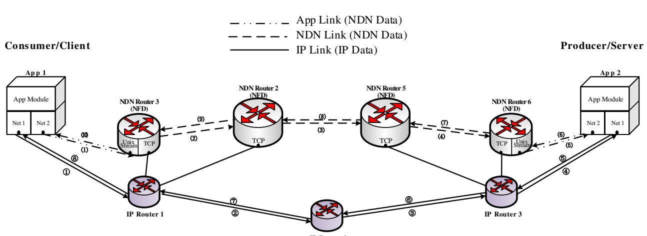

Fig. 2: The data transfer mechanism of Centaur

Compared to traditional P2P, *Centaur* achieves resource discovery and content distribution by NDN network layer instead of DHT, which is a patch on IP application layer. NDN's instinct simplifies resource discovery and improves system performances. As resource data delivery does not require flexible and complex control, end-to-end IP channel can be used to ensure transport efficiently, and reduce overhead caused by deploying NDN as IP overlay. After obtaining the resource, Consumer/Client turns into Producer/Server, which can provide the resource for other nearby consumers.

D. Cache Expiration

Similar to sharing application, *Centaur*'s application also faces the situation of cache expiration, and demands efficient synchronization of dataset among multiple parties. Consider this scenario: NodeA desires resource S and sends Interest to search S. NodeB which can provide resource S, returns a Data bearing the resource info of S and NodeB's IP-address. Data packet is cached in NDN routers along the path. If the dataset is not synchronized, the future request of resource S will gather to NodeB, causing congestion.

Algorithm 1 Hybrid Data Transfer

function : Transfer_Module ($Name$, $Path$)
  $Info \leftarrow$ Consumer($Prefix$, InfoName($Name$))
  $InfoData \leftarrow Info$.opt_Retry_Data(0, 1)
  $Info$.Finish( )
  $addr \leftarrow$ Address($InfoData$)
  $data\_name \leftarrow$ DataName($Name$)
  $seg\_index \leftarrow$ Seg($InfoData$)
  $Data \leftarrow$ Client($addr, data\_name, Path, seg\_index$)
  $Data$.run_TCP( )
  $Data$.opt_TCP(0, seg_index)
  $Data$.Finish( )
end function

At present, NDN community has started to focus on synchronization strategy of application, such as ChronoSync [4] and ISync [5], but still at an early stage. On the one hand, it's difficult to employ the ChronoSync in ndnSIM. On the other hand, ChronoSync achieves data synchronization by the broadcast, which causes high performance overhead. Thus, NDN routers take non-caching strategy instead of ChronoSync to suppress the interference in simulation. In the future, we will design a suitable synchronization strategy for *Centaur*.

IV. Application Layer

In this section, we implement a streaming application prototype to verify the feasibility of our proposal and evaluate *Centaur*s performance. In addition, we specify a control cluster module, which controls system and exchanges control messages with hybrid nodes.

A. Media Traffic Module

As for media traffic module, we design and implement a streaming media application, which is used to describe the work flow between media traffic module and Hybrid Transport Layer. Besides we utilize this prototype to verify the feasibility of *Centaur*. This application prototype mainly consists of Tracker and Storage, as is shown in Fig. 3. Tracker stores and keeps the resource Meta, which is fetched through NDN Transport Module. And Storage deposits the resource data, which is transmitted via IP transport module. Benefiting from data transmission mechanism and NDN's advantages, the prototype can fetch resources from publishers nearby. Fig. 3 illustrates the process that how the application prototype achieves an effective resource retrieval through Hybrid Transport Layer.

1) A Consumer/Client sends a request to the Local Tracker for resource discovery. 2) If there is just an entry matched

in resource meta table, the Local Tracker replies to the Consumer/Client with the resource Meta. 3) Using this resource Meta, the Consumer/Client gets resource data from the Local Storage. 4) If there is not any entry matched in resource Meta table, the Local Tracker turns to the Remote Tracker Module(s) for help. 5) The Remote Tracker with this resource will return a Data carrying resource Meta and IP-address of Remote Storage to Local Tracker. 6) The Local Tracker transmits all the information to the Local Storage. 7) By using this information, the Local Storage fetches and caches the resource data directly via IP Transport Module. 8) The Local Storage informs the Meta of this resource to the Local Tracker and the Local Tracker updates its resource Meta table. 9) The Local Tracker replies to the Consumer/Client with the resource information in the same way of step 2. 10) The Consumer/Client request resource data from the Local Storage.

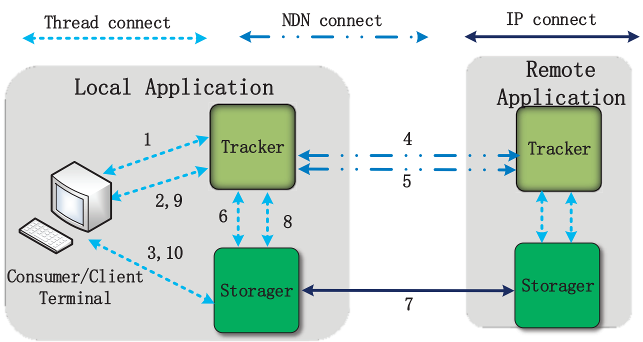

Fig. 3: The Application Prototype

Every time when the application receives a resource, it will publish the resource Meta in NDN network, so that other consumer nearby can access this resource. Besides, the application will upload the resource Meta published to control cluster periodically. Based on the received resource Meta, control cluster maintains information of resource for system. As a result, control cluster can distinguish the popularity and distribution of resources in the entire network. By using this information, control cluster will adjust the distribution of resources to improve the performance of resource retrieval. For instance, when a resource is popular, control cluster will specify some applications to publish this resource in NDN network. If these applications store the resource at the moment, they will first fetch the resource and then provide service.

B. Control Cluster Module

Scalable NDN-based conferencing (SNC) is an NDN-based conferencing architecture for conference control and management [6]. And [7] design and implement an NDN-based scalable control panel for media streaming system. In this paper, we expand the control cluster module from [6], [7], and redesign essential building blocks using (1) light-weighted message for effective control interactions, (2) adaptive and effective resource discovery and management, and (3) secure and distributed access control mechanisms. Control cluster module provides good scalability and adaptivity. And control servers can be added to perfect system services according to user scale. As shown in Fig. 1, control cluster module includes the following 5 types of servers:

- Media Server(MS): MS collects resources and provides resources for consumers as the repository of network.
- Main Control Server: MCS maintains control cluster, and adjusts the number of servers to fit the user scale.
- Agent Register(RA): RA provides user's registration service and allocates key for access control.
- Agent Replicate(PA): PA provides backup function for servers in control cluster.
- Agent Address(AA): AA interacts with users and informs MCS to update user information.

The control cluster mainly provides following services: 1) provides robust control and maintenance services for *Centaur*, e.g., user login, record backups. 2) acts as the repository in the system to provide qualified resources for users. In the future, the control module will provide a preprocessing function which is based on the content's hotness to adjust the distribution of resources.

V. System Evaluations

A. NDN Transmission Overhead

In order to study the overhead which is caused by deploying NDN overlay IP network, we install NFD on IP nodes to realize the experimental scenario. Since overlay network relies on IP channels to translate packets and uses software (NFD) to forward packets. Therefor, transmission effective load and software overhead are selected as the main experimental indexes to measure the performances of overlay network.

1) Effective Load Rate of Transmission: We firstly study the effective load rate of data transmission, which evaluates the ratio between the effective transfer load and the overall load in data transmission. In NDN, the content of Data packet is defined as effective load. Let e and t denote the effective transfer load and the overall load, then the network transmission efficiency f becomes $f = \frac{e}{t}$. Fig. 4 shows the effective load rate of IP and NDN with varying size of packets. the effective load rate of IP transmission is significant higher than that of NDN transmission. In addition, the larger the NDN packet size is, the higher the effective transmission rate is. When NDN transfer data relying on IP tunnels, the transmission load contains not only NDN packets load but also IP packets header which is a communication carrier. Thus, it causes redundant load and inefficient transmission to deploying NDN as IP overlay.

2) Transmission Speed: The next we investigate the effect of NDN packets signature on transmission performance. During the simulation, two cases, NDN packets with signature and NDN packets without signature, are considered to study respectively. And we compare them with IP transmission. As shown in the Fig. 5, the speed of IP transmission is significant faster than that of NDN transmission. In NDN transmission, the speed of packets without signature is much faster than that

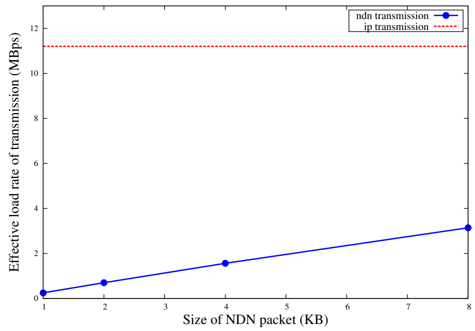

Fig. 4: Effective load rate of transmission

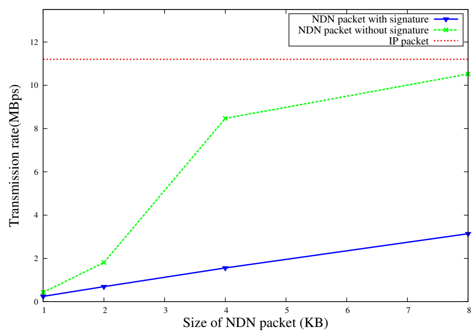

Fig. 5: Transmission speed

of packets with signature. Therefore, we consider that NDN signature operation increases the communication overhead and reduces the transmission speed. There are two reasons for the above phenomenon. Firstly, it brings high overhead of transmission to deploy NDN as IP overlay. Secondly, present NDN just use software (NFD) to handle data forwarding and digital signature. And the performance of NDN communication is inefficient in this way.

B. Centaur Performances

We use a simulation to demonstrate the feasibility of *Centaur*'s architecture and evaluate its performance of data distribution. We implement *Centaur* in ndnSIM and compare it with BitTorrent and NDN-Hippo [8]. BitTorrent is a traditional P2P software based on IP. And NDN-Hippo is a content distribution application based on ideal NDN. We use topology generator BRITE [9] to generate a random network with 10 IP-Routers, n Hybrid-Nodes. Due to ndnSIM's intrinsic limitation, we use a NDN-Router and a User-Node on behalf of a Hybrid-Node. Besides, the NDN-Router and the User-Node are connected by a communication link. Hybrid-Node employs NDN-Router to forward NDN packets. And *Centaur's* application prototype is installed on User-Node. Each IP-Router connects with $n/10$ NDN-Routers through IP channel. In each simulation scenario, we randomly select 5 Hybrid-Nodes as resource servers, which are the only Producers at the beginning of simulation. We run the experiments with a wide range of nodes varying from 0 to 1000 and analyze the resource load rate which represents the load distribution in network.

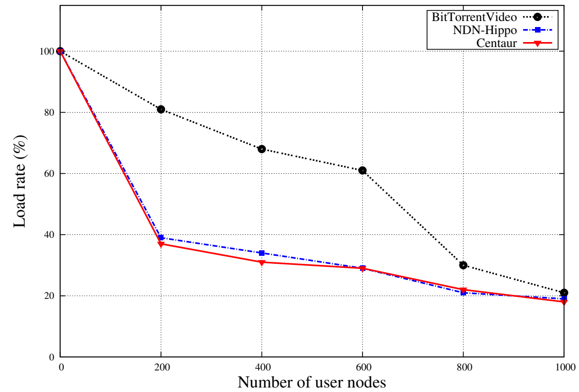

Fig. 6: Load rate of Resource Servers

Fig. 6 shows the resource load rate in three kinds of content distribution applications. Obviously, load rate of *Centaur*'s application is similar to ideal NDN application. When the nodes increase, the load of *Centaur* decreases faster than that of BitTorrentVideo. Benefiting from NDN's advantages, the application based on *Centaur* is more efficient in discovering resources from nearby Nodes than the traditional P2P application. Similar to NDN, *Centaur* has the feature of rapid convergence, and it can achieve load balance. In summary, the performance of *Centaur* is close to that of ideal NDN in content distribution, and *Centaur* can realize lower deployment cost as well as better load balance.

VI. RELATED WORK

How to deploy and implement an efficient and practical NDN [2] has been an increasingly important research topic. Some research efforts have been spent in exploring and embedding NDN into the application framework. nCDN [10], vCDN [11] combined CDN and NDN to propose efficient and scalable CDN architecture. In [10], NDN takes charge of request routing and content delivery while other components of CDN remain changeless. In [11], NDN and SDN are used to achieve programmability in the domain of content delivery. The researches listed above focus on deploying NDN as a component of the existing framework to solve problems. However, this paper focusses on the problems of high overhead and inefficient transmission in NDN streaming application deployment. COPSS [12] introduces the idea of hybrid network to deploy CCN application. [12] also indicates that the processing a CCN packet in a CCN router can be more expensive than forwarding a packet at a normal IP router. Therefore, hybrid network is a more practical solution of NDN deployment.

Several studies are referred to our proposal. NDNx [13] is an open-source of NDN software prototype developed by UCLA, which is a tool for the study of NDN. NFD [14] is a network forwarder that evolves together with the NDN protocol. At present, the above two software are utilized to deploy NDN. This paper designs the control cluster module with self-organization based on SNC [6] which shows an extensible control framework of NDN. ISync [5] and Chronosync [4] provide efficient and robust dataset synchronization strategy for application prototype of *Centaur*. To the best of our knowledge, there are some researches on deployment of other ICN proposals. The experimental system is demonstrated on top of GENI that uses OpenFlow switches to construct the corresponding physical topology [15]. In addition, many ICN projects [16], [17] have attempted to deploy and evaluate their prototypes in SDN. VICN [18] can provide programmability for deploying new network functions and virtualization for co-existing multiple ICN instances on top of the same underlying physical network.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we point out the issues of high overhead and inefficiency transmission caused by deploying NDN as IP overlay, and propose the *Centaur* for streaming application to handle the problems. In *Centaur*, the data transfer mechanism comprehensively employs smart content delivery of NDN and stable data delivery of IP. Benefiting from NDN's advantages, resource discovery and system control can be realized relatively painlessly. Specifically, after receiving resource meta via NDN transport module, IP transport module is employed to obtain resource data. This cooperation promises that our proposal can achieve higher efficiency in media content delivery. We simulate our proposal in ndnSIM and compare it with ideal NDN application and traditional P2P. The result shows that *Centaur* can achieve better load balance with lower overhead. And it is close to the performance of ideal NDN that can achieve in content distribution. In addition, *Centaur* can solve the distribution problems and the control problems in an elegant way. Through experiments and literature review in this field, we believe that *Centaur* is a quite promising candidate for the incremental and compatible deployment of NDN streaming application, and it can shed new light on NDN deployment.

Our future work can be divided into two main aspects. Firstly, we are going to work on enhancements of the *Centaur*, including (1) study suitable synchronization strategy for *Centaur*; (2) substitute SDN for control cluster module and provide more powerful resource management; (3) develop media player for application prototype. Secondly, one thing worth mentioning is that many experiments of *Centaur* can hardly simulate in ndnSIM. We will implement the entity of *Centaur* in practical Internet, and continue our research in performance optimization based on recent NDN routing protocols, design of cache strategy, and so on.

VIII. ACKNOWLEDGMENT

This work has been financially supported by Shenzhen Key Fundamental Research Projects JCYJ20160330095313861, JCYJ20151030154330711, JCYJ20151014093505032 and General Fundamental Research Project No: JCYJ20150626111057728.

REFERENCES

[1] G. Xylomenos, C. N. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos *et al.*, "A survey of information-centric networking research," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, 2014.

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

[4] Z. Zhu and A. Afanasyev, "Let's chronosync: Decentralized dataset state synchronization in named data networking," in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. IEEE, 2013, pp. 1–10.

[5] W. Fu, H. Ben Abraham, and P. Crowley, "isync: a high performance and scalable data synchronization protocol for named data networking," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 181–182.

[6] D. Fesehaye and J. Wei, "Snc: Scalable ndn-based conferencing architecture," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. IEEE, 2014, pp. 872–880.

[7] L. Kai, Y. Longyu, and W. Jun, "Scalable control panel for media streaming in ndn," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 207–208.

[8] K. Lei, Y. Xun, L. Yu, T. Chen, and Z. Tan, "Ndn-hippo: a streaming media system over named data networking," in *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2015, pp. 1–7.

[9] 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.

[10] X. Jiang and J. Bi, "ncdn: Cdn enhanced with ndn," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014, pp. 440–445.

[11] H. Woo, S. Han, E. Heo, J. Kim, and S. Shin, "A virtualized, programmable content delivery network," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on*. IEEE, 2014, pp. 159–168.

[12] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, "Copss: An efficient content oriented publish/subscribe system," in *Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on*. IEEE, 2011, pp. 99–110.

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

[14] Project nfd website. [Online]. Available: http://named-data.net/doc/NFD/

[15] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, "Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed," *Computer Networks*, vol. 57, no. 16, pp. 3207–3221, 2013.

[16] L. Veltri, G. Morabito, S. Salsano, N. Blefari-Melazzi, and A. Detti, "Supporting information-centric functionality in software defined networks," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 6645–6650.

[17] D. Syrivelis, G. Parisis, D. Trossen, P. Flegkas, V. Sourlas, T. Korakis, and L. Tassiulas, "Pursuing a software defined information-centric network," in *Software Defined Networking (EWSDN), 2012 European Workshop on*. IEEE, 2012, pp. 103–108.

[18] J. Ren, K. Lu, S. Wang, X. Wang, S. Xu, L. Li, and S. Liu, "Vicn: a versatile deployment framework for information-centric networks," *Network, IEEE*, vol. 28, no. 3, pp. 26–34, 2014.