Application Design over Named Data Networking with its Features in Mind

Sen Wang, Jianping Wu, Jun Bi
Network Research Center, Tsinghua University
Beijing, China
wangsen@netarchlab.tsinghua.edu.cn; {jianping, junbi}@cernet.edu.cn

***Abstract*—Designed around host-reachability, today's Internet architecture faces many limitations while serving data-oriented applications, which produce most traffic load to the Internet. Many clean-slate designs of the content/data oriented network have emerged to adapt to these needs. Named Data Networking (also known as CCN) is one of these designs to address these limitations from the fundamental level by building network architecture around named data. In this paper, we identify five key features crucial to application design over Named Data Networking and take the voice conference system as an example to show how this features impact the application design significantly in detail. We identify three major challenges facing current voice conference system and illustrate how NDN could help to solve these challenges. A NDN-based design of voice conference system is presented along with discussing its reliability and congestion control.**

Keywords-Named Data Networking; Application Design; Conference System.

I. INTRODUCTION

Internet was designed around a host-to-host model, which is much suitable for most applications at that time (e.g., telnet, ftp, etc.). But, today, most current Internet usage is data-centric [1]. The overwhelming use (>99% according to most measurements) of today's networks is for an entity to acquire or distribute named chunks of data (like web pages or email messages) [2]. Actually, users want to get data or service rather than communicate with the host which holds these data or service. With this insight, some clean-slate redesigns of Internet Architecture have emerged including CCN (Content Centric Networking) [3], DONA (A Data-Oriented Network Architecture) [4], etc. Therefore increasing attention has been attracted into this research area.

Named Data Networking (also known as CCN) [3] is a newly proposed Internet architecture which is designed around named data to address the limitations of today's Internet from the fundamental level. We expect that the success of NDN would largely depend on whether the new architecture can support various application needs more effectively and efficiently as it promises. So, designing applications over NDN is an extremely important issue to solve. In this paper, we identify five key features crucial to application design and take the conference system as an example to show how this features impact the application design significantly in detail.

The rest of this paper is organized as follows. In Section 2, five NDN Features are elaborated. Section 3 identifies three main challenges facing conference system and explains how NDN could help to solve these challenges with its embedded features. Section 4 takes the conference system as an example to show how features of NDN impact the application design significantly in detail. Finally, we conclude in Section 5.

II. NDN FEATURES FOR APPLICATIONS

As a promising, clean-slate network architecture, NDN is designed from a data-centric perspective. Differing from conventional connection-based TCP/IP architecture, NDN has its own features and its effects in design of applications which is summarized in this paper as follows:

First, NDN adopts the Publish/Subscribe communication paradigm to build a data-centric network architecture. The Publish/Subscribe paradigm is a vital ingredient for future services and applications. It allows asynchronous and decoupled many-to-many communication and typically supports data-centric information dissemination [12]. Sending Interests can be viewed as some kind of subscribe and the data delivery can be seen as a publishing process. The Publish/Subscribe paradigm decouples the producers and consumers of data in both time and space [13], which is the nature of most applications [12].

Second, NDN is receiver-controlled by nature. The original objective of the TCP/IP Internet architecture is to interconnect all existing networks and hosts uniformly and efficiently [5]. When a host connects to the Internet, it can communicate with arbitrary host connected to the Internet by its IP address. This enforces today's Internet a sender-control manner naturally. The Publish/Subscribe communication paradigm decoupled the producers and consumers of data. Producers don't need to hold references of consumers and know how many subscribers are participating in this interaction [13], and vice versa. In this paradigm, the conventional sender-controlled manner is not effective. We speculate that a receiver-controlled manner is more suited for NDN. As a clean-slate Internet architecture, this transition of NDN will turn Internet from push mode to pull and impact application design and implementation significantly.

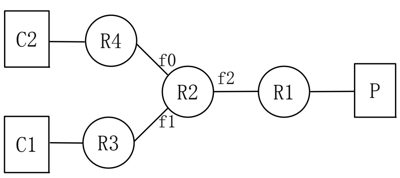
Third, NDN provides an auto-organized and asynchronous multicast distribution mode. In NDN, each chunk of data is named and can be transmitted and stored independently, which provides a substrate for the multicast distribution mode together with the Publish/Subscribe communication paradigm. Specifically, by compressing the interests with the same name and responding to interests with data cached in the intermediate routers, an auto-organized and asynchronous multicast distribution mode is provided in NDN network. As Figure 1 (a) shows, when the two consumers, say C1 and C2, send the Interests to the same datum published by P almost simultaneously, NDN router R2 will compress the two Interests and send just one Interest to R1. After the datum arrives at R2, R2 would find that the Interest requesting this datum has two corresponding interfaces f0 and f1in the PIT table (Pending Interest Table, a recording list of Interests which have been forwarded, while their corresponding Data have not been received yet) and send two copies of the datum through f0 and f1 respectively. The datum will be cached in the Content Store of R2. It should be noted that this

 ISBN: 978-1-61208-183-0

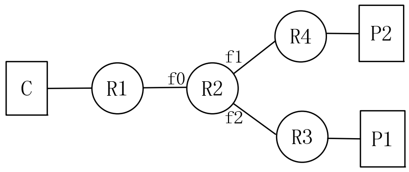
synchronization of the two interests is not necessary. Assume that C1 sends the Interest before C2 does. Before C1 gets the datum, the Interest sending by C2 can always join in the entry in the PIT table of R2, which is generated by the prior Interest. After the C1 gets the datum, the Interest sent by C2 can be satisfied by the cached copy in the Content Store of R2. It is found that this kind of multicast is auto-organized, and there is no need for any extra routing state or control traffic.

Besides this kind of one-to-many distribution mode where many consumers are interested in the same data, NDN also provides some kind of many-to-one distribution mode where many producers publish different data with the same name prefix, and a consumer sends a series of interests with the name prefix to get all these data matching the name prefix. We refer to this kind of one-to-many distribution mode as Enumeration Process. As Figure 1 (b) shows, the two producers, say P2 and P1, publish two data with the two names, ccnx://thu.edu.cn/course-A/homework/sam and ccnx://thu.edu.cn/course-A/homework/alice, and have the same prefix ccnx://thu.edu.cn/course-A/homework/. A consumer, namely C, who wants all the homework of course A, sends an Interest with the name ccnx://thu.edu.cn/course-A/homework/. When the Interest reach NDN router R2, R2 looks up the entry for this prefix in its FIB table (Forwarding Information Base. It is much similar to the FIB of current IP router) and forwards it to R4 and R3 from f1 and f2 respectively. Both P1 and P2 will receive the Interest and respond with its datum respectively. The two data will arrive at R2, and only one of them will be send to the R1 because one interest can just get one datum. Assume that the datum tagged with the name ccnx://thu.edu.cn/course-A/homework/sam is received by C, then C would send another Interest with the same name ccnx://thu.edu.cn/course-A/homework/ but with an attribute Exclude set with the parameter sam which means data with the name constructed by suffixing the interest's name with sam are not viewed as matching this Interest. So, this Interest will get the datum with name ccnx://thu.edu.cn/course-A/homework/alice, which has been cached in R2. Repeating this process, C can get all data tagged with the name prefix ccnx://thu.edu.cn/course-A/homework/. The process will not finish until an Interest gets no datum in an expiration time.

Forth, NDN offers infrastructural support for applications to be designed in a server-less manner. In NDN, named data are the first-class residents and Interests are routed directly according to their names. So for NDN, there is no need to map the wanted data names to their locations. Taking traditional VoIP software based on SIP as an example, the major reason for the existence of a central server is to provide with some kind of name resolution which resolves the human-readable name of a user to current IP address of the host, from which the user register to the server. This kind of complexity of structure and configuration results from a mismatch between the user's goal and the network's means of achieving it [15].



(a) One-to-many



(b) Many-to-one

Figure 1. Simple scenarios for one-to-many and many-to-one distribution mode

Fifth, NDN transmits each piece of data with a signature which is generated by the data's publisher by signing the readable name and its corresponding datum with its public key. The consumer can validate the integrity of the datum received and the association of the datum and its name. Applications can use this signature and some key distribution mechanism based on NDN itself as a foundation to satisfy their own secure demands.

III. SOLVE THE MAJOR CHALLENGES OF CONFEFENCE SYSTEM

In this section, we take the conference system as an example and identify three major challenges facing current conference systems and illustrate how NDN could help to solve these challenges.

IP multicast model is viewed as a scalable and efficient pattern for multi-party communication [6]. But for lack of extensive deployment of IP multicast, designs of conference system based on IP multicast are not accepted widely. Many researchers turned to design conference systems based on centralized server [7]. These designs transfer the scalable problems of endpoint to the server and make the situation even worse for the server must deal with media flows of all the endpoints. In [8][9], it is proposed to construct an application-level multicast overlay over IP to delivery data for conference system. In spite of these solutions, the scalability of conference system is still an open issue. As the VoIP market is growing rapidly, for those who don't want to transfer from traditional phone system to VoIP system, the main concern is the quality of actual VoIP calls [10]. Kushman et al. [10] shows that the qualities of current VoIP systems are unacceptable due to network outages. The main cause was identified as the poor performance of BGP update. Another main concern about conference system as well as many other applications is secure issue. How to keep the privacy and integrity of the calls and allow only granted users to access conference resources is still an open problem in the context of poor secure infrastructure of the Internet. In short, the three major challenges of designing a conference system are i) Scalability, ii) Quality of calls, iii) Security.

With its data-oriented nature, NDN brings enormous potentials and challenges to application design. We argue that NDN provides a substrate for resolve the aforementioned three challenges. First, NDN names data directly instead of naming host and involves Publish/Subscribe paradigm of communication, which make it possible to automatically embrace some kind of multicast providing a scalable and efficient pattern for multi-party communication [6]. Second, NDN does not have routing loops for its data-name-based design [11]. Interest can be forwarded along multiple paths. This feature allows rapid recovery from network outage, as [10] suggests that multi-path routing is a promising direction to

 ISBN: 978-1-61208-183-0

deal with unintelligible quality of VoIP calls caused by BGP update. Third, naming data makes it possible to secure data itself instead of securing the transmission channel. Today's connection-based network architecture does not provide essential infrastructure for securing data, which is the main concern of most applications. As an add-on function, many solutions were proposed to provide various-kind and various-stage security of communication channels. NDN realizes the transition from channel-oriented security to data-oriented security. The task of securing the data can be accomplished by end-to-end cryptographic signatures and encryption (when data secrecy is needed), leaving open only the task of key management among the data sending/receiving parties, but not any channel or boxes in the middle of the data delivery paths [11].

IV. AN EXAMPLE OF COFERENCE SYSTEM DESIGN

In this section, we take the voice conference system as an example to show how this features impact the application design significantly in detail. We identify three main issues of a conference system to be resolved, i) how could a participant know the names of active conferences without centralized server; ii) how could a participant get the name list of other users in a conference; iii) how does a participant get the audio data of other participants in the same conference. We refer to these issues as conference discovery, speaker discovery and voice data distribution respectively. In the following two subsections, some discussions about these three parts are presented, and more details can be found in [14]. After these three main parts, we would discuss some extended features including reliability and congestion control for a conference system and show the real potential of NDN for application design.

A. Conference and Speaker Discovery

Without the existence of central server, a participant needs to communicate with all other conference creators or participants in the same conference for conference discovery or speaker discovery process respectively. These two use cases match the NDN enumeration process aforementioned in Section 2 perfectly. The Interest with the names used for the two processes would be routed by either broadcast or multicast. This kind of multicast in NDN can be achieved by some kind of mechanism where the names like a multicast IP address in that the publishing process resembles the group joining process of IP multicast and the forwarding of Interest resembles data transmission of IP multicast. But it should be noted that this process is used for getting data from multiple parties, and IP multicast is used for sending data to all group members. In contrast, to fetch voice data of other participants, the location-dependent names of participants are used because there is no need for broadcasting Interest. So there would be no additional state imposed to the routing system. Actually the voice data will be efficiently delivered to multiple receivers as Section 2 shows. For conference and speaker discovery and voice data distribution, using separate namespaces makes the system more scalable.

B. Voice Data Distribution

As Section 2 shows, a participant's voice data can go through an automatically-formed spanning tree and arrive at each other participant more efficiently than unicast. This property makes the NDN conference system more scalable than traditional unicast-based conference system. Besides, the producer and consumer of voice data are decoupled in both time and space through Publish/Subscribe communication mode. In terms of time, the producer just publishes its voice data independently and does not need to generate responding datum for the arrival of an Interest designedly. In terms of space, the producer does not know how many and who are receiving its voice data. It can be observed that the Publish/Subscribe mode makes the design simple and efficient. The transmission of real-time stream can be decoupled and appears to be of Publish/Subscribe mode by nature. On the other hand, delivery of voice data is controlled by the consumer in that the consumer controls which chunk of voice data it want to get and how fast these data would be transmitted.

C. Reliability

In this subsection, we discuss the reliability of data distribution of NDN conference system here. Considering the extended function of whiteboard, we could borrow the ideas from literature [16], which designs a reliable framework for IP multicast. IP multicast can be viewed as a special case of Publish/Subscribe communication mode. Joining a group is to express interest in certain subject and delivering data is to publish messages to the interested. The difference is of the granularity in that the NDN makes use of the late-binding technique, but for IP multicast, a receiver keeps a session relationship after joining a group. For IP unicast, the sender has control of data transmission in terms of flow control, reliability, etc. When it comes to reliability of IP multicast, it seems not work well. Floyd et al. [16] shows a transition from sender-based to receiver-based control in the context of reliable multicast due to the fact that the sender cannot keep controlling the transmission any more for so many and delay-diverse receivers. For NDN, a receiver-based reliability mechanism is much more natural. Each receiver is responsible for its reliability of data delivery and keeps independence on correct reception of data.

Besides, as [16] suggests, the "naming in application data units (ADUs)" model works far better for multicast than IP address-based one. NDN architecture provides applications with unique and persistent names, which would eliminate the delay and inefficiency imposed by separate protocol namespace [18]. Furthermore, the performance of retransmission could be improved by data cached in NDN router or other participants who have received the data already. As [18] argues, to design a performance-optimal and efficient transport protocol, some application information (e.g., application data units) should be involved in the protocol design. The concept of networking named data could be viewed as an application of this viewpoint into the network layer, which provides significant efficiency and flexibility for the design of the upper layers.

In summary, with receiver-control mode and application-specified name which are embedded in NDN architecture, reliability can be naturally achieved with some mechanism similar with SRM [16].

D. Congestion Control

The Internet's heterogeneity and scale make multipoint communication design a difficult problem [17]. If a participant generates only one kind of quality of audio data (e.g., with a certain encoding rate), other participants will have a uniform transmission rate of audio data of the participant. This means some low-capacity regions of network suffer congestion and some high-capacity regions are underutilized. To solve this problem in the context of IP multicast, McCanne et al. [17] proposes a receiver-driven layered solution. This solution can be transplanted into NDN circumstance naturally. The NDN is receiver-driven by nature, and its application-specified name is well-suited for a layered solution. We can give different qualities of audio data different names e.g.

Ccnx://thu.edu.cn/bob/audio/high_quality/seq<20>

Ccnx://thu.edu.cn/bob/audio/low_quality/seq<20>

A participant can try to get the audio data of higher quality periodically. If congestion is detected, it would give up this trial and stays on its original quality-level. This process is of lower cost than that in the context of IP multicast in that joining and leaving an IP multicast group is costly, but for NDN, it is costless for NDN's late-binding property.

V. CONCLUSION

It could be found that the many-to-many data distribution mode of NDN allows multi-communication applications, like the voice conference system, to be designed more naturally and efficiently. By sending Interests with different names of voice data, a conference participant can migrate smoothly from one quality level of voice data of other participants to another according to its bandwidth. Furthermore, the content caching mechanism makes the reliability of multicast transmission mode more simply and efficiently. Our future works include studying the reliability and congestion control of NDN for the voice conference system in more detail and extending the implementation presented in [14] with these functions. We will also attempt to address some limitations of NDN in some special application scenarios as our future work. For example, in the scenario of emergent report such as earthquake alarm, data are generated unpredictably. Therefore, either a long-lived Interest is needed, which would occupy the PIT entry for an extremely long time. Or, applications need to send interests periodically. Both solutions aforementioned seem not to be as efficient as current sender-based IP approaches.

REFERENCES

[1] S. Shenker, "We Dream of GENI: Exploring Radical Network Designs," presentation, CRA Computing Comm unity Consortium, 2007;

[2] V. Jacobson, "If a clean slate is the solution what was the problem", Stanford" Clean Slate" Seminar, 2006.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," ACM CoNEXT '09, 2009, pp 1-12.

[4] T. Koponen, M. Chawla, B. G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. "A Data-Oriented (and Beyond) Network Architecture," ACM SIGCOMM Computer Communication Review, 2007, vol. 37, no. 4, pp. 181-192.

[5] J. Wang, E. Osterweil, C. Peng, R. Wakikawa, L. Zhang, C. Li and P. Cheng, "Implementing Instant Messaging Using Named Data," Proceedings of the Sixth Asian Internet Engineering Conference, 2010, pp. 40-47.

[6] D. Pendarakis, S. Shi, D. Verma, M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems, 2001, pp. 5-5.

[7] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol", RFC 4353, 2006.

[8] C. Luo, J. Li, and S. Li, "DigiMetro-an application-level multicast system for multi-party video conferencing," GLOBECOM'04, 2004, vol. 2, pp. 982-987.

[9] X. Wu, K.K. Dhara, and V. Krishnaswamy, "Enhancing Application-Layer Multicast for P2P Conference," Proc. of IEEE Consumer Communications and Networking Conference, 2007, pp. 986-990.

[10] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?! it must be BGP," ACM SIGCOMM Computer Communication Review, 2007, vol. 37, no. 2, pp. 75-84.

[11] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture, 2010, pp. 3-8.

[12] M. Särelä, T. Rinta-aho, and S. Tarkoma, "RTFM: Publish/subscribe internetworking architecture," ICT Mobile Summit, 2008.

[13] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.M. Kermarrec, "The many faces of publish/subscribe," ACM Computing Surveys, 2003, vol. 35, no. 2, pp. 114-131.

[14] Zhenkai Zhu, Sen Wang, Xu Yang, Van Jacobson and Lixia Zhang, "ACT: Audio Conference Tool Over Named Data Networks," ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2011), 2011, vol 11.

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

[16] S. Floyd, V. Jacobson, S. McCanne, C.G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," ACM SIGCOMM Computer Communication Review. 1995, vol. 25, no. 4, pp. 342-356.

[17] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," ACM SIGCOMM Computer Communication Review, 1996, vol. 26, no. 4, pp. 117-130.

[18] D.D. Clark and D.L. Tennenhouse, "Architectural considerations for a new generation of protocols," ACM SIGCOMM Computer Communication Review, 1990, vol. 20, no. 4, pp. 200-208.

 ISBN: 978-1-61208-183-0