Live Streaming with Content Centric Networking

Hongfeng Xu[2,3], Zhen Chen[1,3], Rui Chen[2,3], Junwei Cao[1,3]
[1] Research Institute of Information Technology
[2] Department of Computer Science and Technology
[3]Tsinghua National Laboratory for Information Science and Technology
Tsinghua University, Beijing 100084, China
Email: zhenchen@tsinghua.edu.cn

Abstract—**Media streaming is the killer application in current Internet. There are a variety of media streaming techniques in today's Internet, such as RTSP, HTTP live streaming and Adobe Flash etc. HTTP live streaming (HLS) is a popular and most promising technique as the protocol is based on the Internet workhorse protocol i.e. HTTP, and supported by HTML5 and mobile platform. Most of these media streaming techniques are based on TCP/IP, which is built on the traditional host-to-host network architecture. The host-to-host architecture is proved to be inefficient in content distribution with a lot of bandwidth waste, and it is complicated to deploy network service because of TCP/IP's location-dependence. Content Centric Networking (CCN) is a future Internet architecture which is targeted to solve the above problems by location-independent content naming and universal content caching in router. In this paper, we investigate both HTTP live streaming and CCN, and propose a design of CCN live streaming, which is a media streaming technique base on CCN. Finally, we demo our CCN live streaming on Android client, and conduct evaluation experiments. The results demonstrate that the CCN live streaming is a low-cost scheme and much easier to deploy and configure in operation compared with HTTP live streaming.**

Keywords-Media Streaming; Content Centric Networking; HTTP Live Streaming;Future Internet Architecture

I. INTRODUCTION

The Internet has evolved to be concentrated on content distribution. Especially in recent years, with the rapid development of access network bandwidth, most of the content on the Internet exist as information-intensive form like video or audio. According to the Cisco Visual Network Index [1], video content take 40 percent of all the Internet content in 2010, and that will be 50 percent in 2012.

People's real-time requirement of information retrieval becomes more and more strong, this is why media streaming techniques appear, and the growth of access network bandwidth also makes it possible. We can see media streaming techniques everywhere on the Internet today, online video, Internet radio, Internet TV and so on. However, all these media streaming techniques are based on TCP/IP, which is host-to-host architecture. Using the host-to-host architecture in content distribution may lead to bandwidth waste, because you have to get the content data really from the server, even some hosts nearby may already got the data. Secondly, TCP/IP is a location-dependent protocol [6], which makes it complicated to deploy and configure network service. HTTP live streaming [2,3] is one of these media streaming techniques, which is based on HTTP protocol.

Content Centric Networking [4-6] is a future Internet architecture. The core idea of CCN is named data, that means user do not get content from the data's location, but the data's name. This is a really significant improvement, exchanging the content distribution from "where" to "what". It is important because most action of today's Internet is content distribution, and the users do not care where the data is, but what the data is. Named data architecture make the network easy to cache content, actually we can say that CCN is natively support content caching. Additionally, because named data make CCN location-independent, network service can be much easier to deploy and configure in CCN.

In this paper, we propose a design of CCN live streaming. CCN live streaming splits the video into a sequence of segments, and generates a playlist file, as HTTP live streaming do. The user who wants to play this video streaming can download the small playlist file, and play the video by getting the video segments one by one. All the file delivery is using CCN protocol, instead of HTTP. The last section of this paper contains some experiments that prove CCN live streaming is more efficient and flexible than HTTP live streaming on content distribution.

II. HTTP LIVE STREAMING

A. Introduction to HTTP Live Streaming

Live streaming is technique that the end-user can play the video of an event as it happens, that means the video data is still transmitting or even generating. One of the most widely used live streaming technique is HTTP based live streaming, which implemented by Apple Inc. [3]

HTTP live streaming works by splitting the overall video streaming into a sequence of small HTTP-based downloads. Figure 1 is the architecture of HTTP Live Streaming given by the web site of Apple.

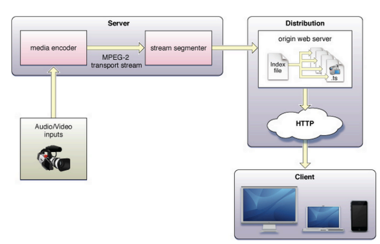


Figure 1. Structure of HTTP Live Streaming System [3]

The media encoder will encode the original audio or video into MPEG-2 stream, and the stream segmenter will break the stream into a sequence of stream segments (ts file), and also generate the playlist which store in the index file (m3u8 file). At the start of the streaming session, the client application will download the index file first, and then start to download the segment files by analyzing the index file, all these data transmitting are using HTTP protocol. The HTTP Live Streaming protocol details can be found in IETF draft [2].

B. Disavantages of HTTP Live Streaming

1) Availability

The HTTP live streaming technique relies on the traditional host-to-host network model, and is a client-server architecture. As we know, one advantage of the host-to-host model is that, if the client-end wants some data from the server-end, then the client's requests have to go through the whole network between them and arrive the server-end, then the server-end will send the data back. That means, when you want to watch a live video on the Internet, you have to get the video stream really from the video server, even that your roommate is watching that video too, and actually he has already download some parts of the video stream. Obviously, this will waste much bandwidth, and make the Internet overwhelmed.

2) Location-dependence

The communication in HTTP live streaming is host-to-host, because the underlying implementation is TCP-IP. That means if you want to watch a live video, you should know the IP address of the video server. The host-to-host network architecture was introduced to fit the problems of the '60s. However, today's Internet has evolved to be dominated by content distribution and retrieval. The users of Internet do not care where the content is, but what the content is. Mapping content to host locations makes the network services complicated to implement and configure.

III. CONTENT-CENTRIC NETWORKING

A. Introduction to CCN

Content-centric networking is a future Internet architecture, which founding principle is that Internet should allow users to get data by the data's name, instead of the data's physical location. Just like TCP/IP, CCN network stack is a thin-waist model, too. However, the thin waist part in CCN is a content chunks instead of IP layer. In the view of network, this means that CCN index data from the data's name but not the physical location. In IP network, the users request data by giving the destination IP. In CCN, the users send a request called Interest which contains the data's name. Additionally, CCN can be layered over everything, even layered over IP. Everything over named data, and named data over everything.

There are two package types in CCN, Interest and Data. The consumer node requests a named data by sending an Interest, which contains the name of the data. Any node that got the Interest can respond the Interest and send the named data back if it has already got the data. The node that responds the Interest may be the producer node, or the middle CCN router node which is caching the data.

Figure 2 is the forwarding model of CCN. Content Store and Pending Interest Table (PIT) are the two components that implements CCN content caching. For content store, it is easy to understand, the CCN router will cache data in it when a new data arrives. For the PIT, it will keep the Interest which sending by this router and still on the way. When a new Interest arrives, the router will firstly check the Content Store. If the named data exists, the router will send this data back to the link where the Interest comes. If the data cannot found in Content Store, the router will check the PIT, if the same pending Interest exists, then add a new item in PIT which records the Interest and the face which the Interest come from. So when the data arrives, the router will check all the items in PIT and forward the data to all of those faces. If the Interest does not match the Content Store and the PIT, then the router will forward the Interest according the FIB, and also add an item in PIT. This is almost how the CCN content caching works.

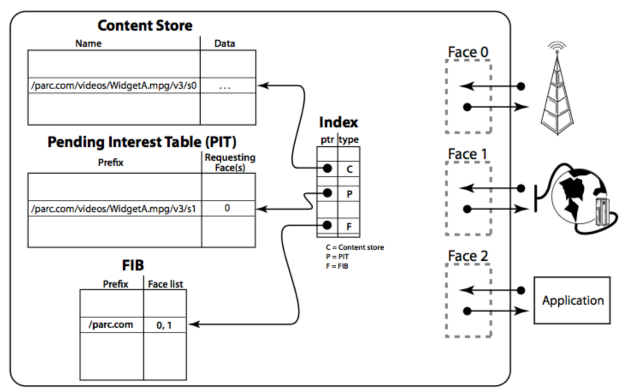


Figure 2. CCN Router Forwarding Model

B. Advantages of CCN

1) Content Caching

CCN natively supports content caching. The named data model of CCN makes the CCN router very easy and natural to cache data that go through the router. Every CCN router has a component called content store which can cache the data, and a Pending-Interest-Table which can remember the Interest history. When a new Interest arrives, the CCN router searches the content store first, if the data not exists then search the PIT. A same interest exists in the PIT means that there is someone sends an interest before but the data is still on the way, so the router will not forward the interest again. This content caching mechanism of CCN can effectively reduce the network latency and bandwidth cost.

2) Location-independence

As we know, TCP/IP network architecture is location-dependent, which makes Internet service complicated to implement and configure. CCN based on named data, not the physical location. CCN protocols do not contain any location information, so it is easy for Internet application to deploy and configure their services.

IV. CCN LIVE STREAMING

A. Introduction

We mentioned above that HTTP live streaming has some disadvantages because it relies on session mechanism of TCP/IP, and we also mentioned some advantages of CCN. In this paper, we want to introduce CCN live streaming. In CCN live streaming, client requests video stream by sending CCN interest, and get the response data using CCN protocol instead of HTTP.

B. Advantages of CCN Live Streaming

Obviously, CCN live streaming can benefit from the content caching mechanism of CCN. Again, we take the same example that mentioned at the start of this paper, imagine that you want to watch a live video, at the same time your roommate is watching that video too, and your computers connect to the same router which is a CCN one. If your video process is later than your roommate, then the video stream segments are already cached in the router. Or, if your process are almost same with your roommate, then the router will only forward one interest to the Internet because the pending mechanism of CCN, this can reduce the network congestion and improve the video stream's delivery speed.

Additionally, for the live streaming provider, CCN live streaming is much easier to implement and configure, because of its location-independence. Current CCN implementation ccnx is based on overlay network. Our CCN live streaming can effectively leverage the overlay routing to detect and adapt the underlay congested path quickly and maintain the continuous service as suggested in RON[10] and CORS[11]. Li Tang and Hui Zhang et al. also have analyzed the benefits and the feasibility to improve the End-to-End QoS in [12-13]. Yin Chen et al. propose an easy-to-follow overlay path selection algorithm in [14].

C. Implementation

We implement a CCN live streaming demo based on the ccnx project [7-9]. Before talking about the implementation details, we should introduce some proper names in the ccnx project.

- *ccnd*, it is a CCN routing daemon. Each CCN node should run its own ccnd.
- *ccn-repo*, it is a CCN based application, which is a repository that can store CCN data.

The demo includes a live streaming service on Linux platform (as shown in Figure 3) and client software on Android (as shown in Figure 4).

1) Server-end Architecture

As show in Figure 3, we are running a ccnd and a ccn repository in the server-end. Just like HTTP live streaming, first we break the input video into a sequence of segments, and also generate an index file, which is actually a playlist. Then we put all the files in the repository using the *ccnputfile* tool given by *ccnx* project.

2) CCN name model design

Actually, out name model design is quite simple. Our server is in the CCN name space ccnx://ccn.tsinghua.edu.cn. So, if Bob is one of our users, he uploads a video named clock.avi, and then the server will generate an index file named clock.m3u8 in the CCN name space ccnx://ccn.tsinghua.edu.cn/bob/clock.m3u8. And all the segment files are in the name space like ccnx://ccn.tsinghua.edu/cn/bob/clock/block1.ts.

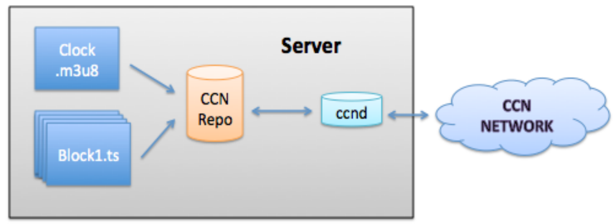


Figure 3. Server-end Architecture

3) Client-end Architecture

We implement a client software base on Android platform. Just like the server-end, we run a *ccnd* on the client. The software includes a media player on Android, which supports HTTP live streaming. The HTTP proxy is the core component, which translates the HTTP request of the media player into CCN interest, then the *ccnd* will forward the interest. When the CCN response arrives the ccnd, the ccnd will send the CCN response to the HTTP proxy, and the proxy will translate the CCN response into HTTP response and send it back to the media player.

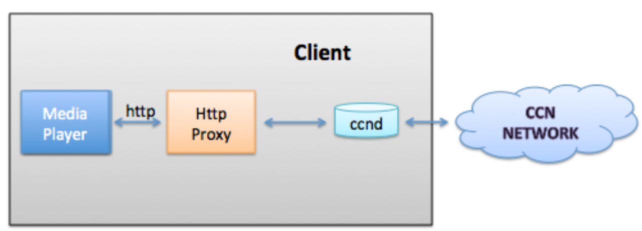


Figure 4. Client-end Architecture

V. EXPERIMENTS

A. Experiments Design

We design and implement a CCN live streaming application as part of the project Omedia. Omedia is a CCN project of Tsinghua NSLab, which is a CCN content delivery system based on social network. Figure 5 shows the Android client implementation and UI of Omedia.

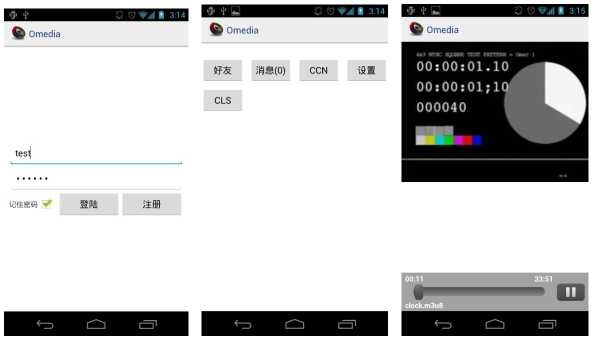


Figure 5. Android Client Implementation

We conduct two kinds of experiments with Omedia. As show in Figure 6, the first experiment is designed to demonstrate that CCN can reduce the network load as the number of simultaneous watch clients increased. All the Android clients are tested in the Tsinghua campus network, and connect to Internet through a CCN router. First, all the clients will play the same video at the same time using both CCN live streaming and HTTP live streaming, then measure the network bandwidth consumption and video buffer delay respectively. This can show the function of the pending mechanism in CCN when content caching. Secondly, the clients will play the video one by one, this will show the function of the content store in CCN when content caching.

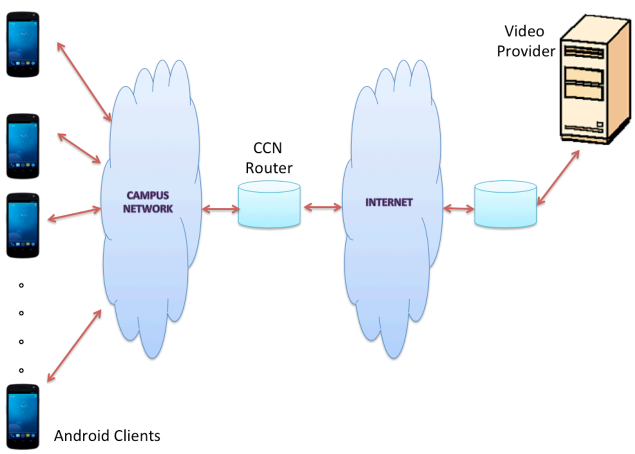


Figure 6. Android Omedia clients retrive video content

Another experiment, shown in Figure 7, is to demonstrate that CCN live streaming can be much easier to deployed and configured. Location-dependence makes TCP/IP routing complicated to configure, or worse, the service provider may even not able to configure the routing because it can only be configured by ISP. Take the path selection for example, the client which using HTTP may choose an unsatisfied default routing path; even that path experiences serve loss ratio and bad latency. Hence, overlay routing to bypass the bad link or router may gain a better performance, while HTTP Live Streaming will not offer such flexibility as the routing is based on IP route. CCN overlay routing is easy to configure to use a new overlay path. Actually, CCN has a strategy layer which can do this work. This experiment will measure the bandwidth when the client chooses the different path.

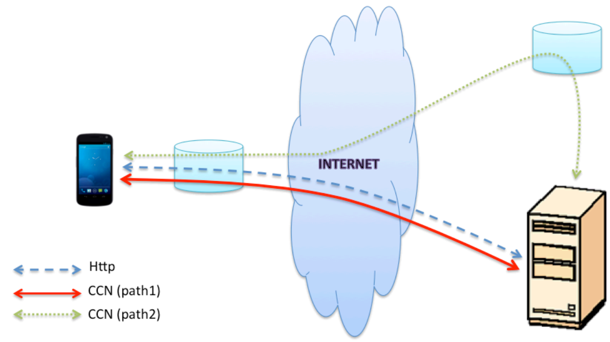
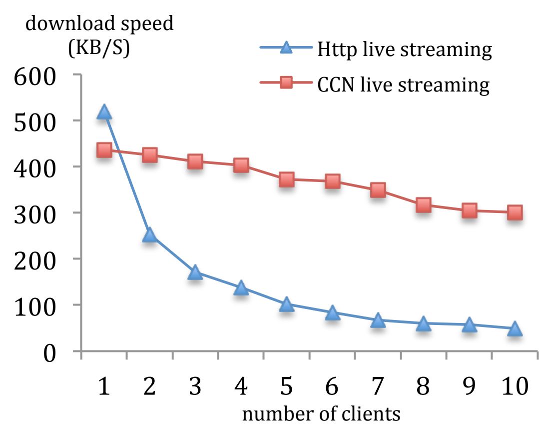


Figure 7. Overlay path selection in live streaming with CCN

B. *Experiment Results Analysis*

As show in Figure 8, from the experiment results we can see that, as the number of clients increasing, the network load will get worse and worse if using HTTP live streaming. CCN can make this condition much better because of content caching.



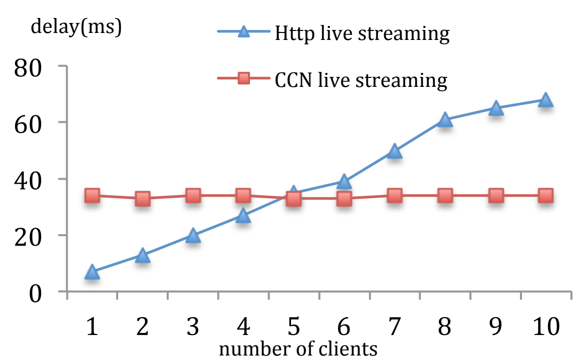
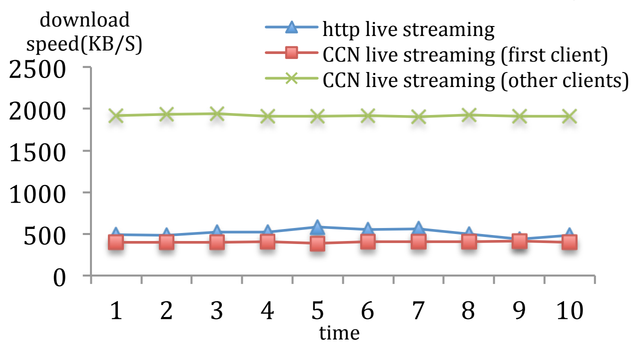


Figure 8. The media buffer delay in playing video simutaneously with lots of Omedia clients

Figure 9 shows the condition that clients play the video one by one, then most of the clients except the first one can have a much better network bandwidth and delay.



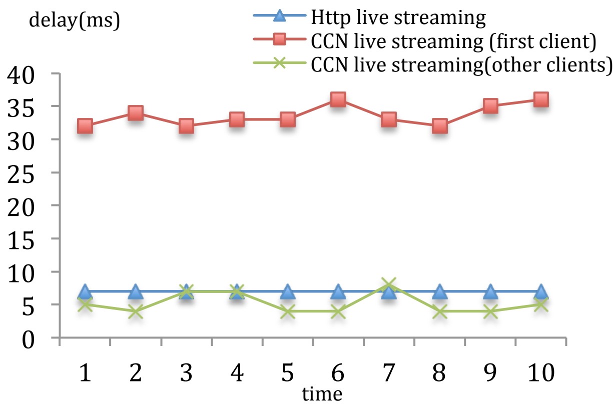


Figure 9. The media buffer delay in playing video one by one with Omedia Clients

Figure 10 is the results of the experiment describes in Figure 7. The results show that CCN live streaming can easily configure a routing path that bring a better network bandwidth.

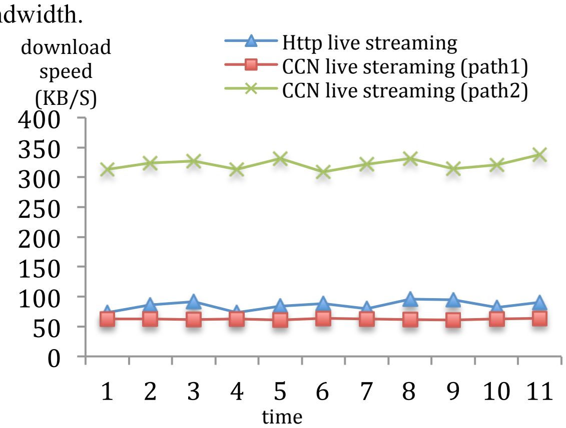


Figure 10. Achieveable network bandwidth with different CCN overlay path

VI. CONCLUSION

Traditional media streaming techniques like HTTP live streaming are based on TCP/IP which is built on host-to-host network model. This model is inefficient in content distribution, and also makes the streaming service complicated to deploy and configure. This paper proposes CCN live streaming, which is a media streaming technique base on Content Centric Networking. CCN live Streaming is easy for content caching, and overcome the location-dependent problem. We demonstrate and evaluate a CCN live streaming implementation based on CCN overlay testbed and Android Omedia client, and the experiments show that CCN live streaming is low-cost for video distribution and much easier for streaming service to deploy and configure.

VII. ACKNOWLEDGMENT

This work is supported by Ministry of Science and Technology of China under National 973 Basic Research Program Grant No. 2011CD302600, Grant No.2011CB302805 and China NSFC A3 Program (No.61161140320).

VIII. REFFERENCES

[1] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015.

[2] IETF draft, HTTP live streaming, HTTP://tools.ietf.org/html/draft-pantos-HTTP-live-streaming-08 .

[3] Apple HTTP live streaming site, HTTPs://developer.apple.com/resources/HTTP-streaming.

[4] Van Jacobson's talk at PARC Forum: "The Good, Bad, and Ugly of Digital Distribution: A Content-centric Networking Perspective on Evolving Network Architecture", February 2011.

[5] Van Jacobson's tutorial at Future Internet Summer School (FISS09) Short course, July 22, 2009.

[6] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass,Nicholas H. Briggs, Rebecca L. Braynard, Networking Named Content, CoNext,Rome, Italy, 2009.

[7] CCNx Project, www.ccnx.org

[8] CCN4B project, HTTPs://github.com/truedat101/ccn4b.git

[9] PyCCN project, HTTPs://github.com/remap/PyCCN

[10] D. G. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan, Resilient overlay networks, In Proceedings of the 18th ACM Symposium on Operating Systems Principles, 2002.

[11] Li Tang, Zhen Chen, Hao Yin, Jun Li, CORS A cooperative overlay routing service to enhance interactive multimedia communications, Journal of Visual Communication and Image Representation, February 2010.

[12] Hui Zhang, Li Tang and Jun Li, Impact of Overlay Routing on End-to-End Delay, Proceedings of 15th International Conference on Computer Communications and Networks, 2006. ICCCN 2006.

[13] Li Tang, Zhi-jie Sun, Zhen Chen, Jun Li, On the Feasibility of Enhancing Interactivity for Real-time Communications using Overlay Routing, ICNS 2007.

[14] Yin Chen, Li Tang and Jun Li, Heuristic Relay Node Selection Algorithm for One-hop Overlay Routing, The 28th International Conference on Distributed Computing Systems Workshops, 2008.