Vol.2015-AVM-89 No.6
2015/8/3

大規模フィールド実験におけるコンテンツ先回り配信の性能評価

金井謙治[†1]　武藤健史[†1]　甲藤二郎[†1]　山村新也[†2]　古殿知之[†2]　齋藤孝文[†1]
三上博英[†1]　草地薫[†1]　津田俊隆[†1]　亀山渉[†1]　朴容震[†1]　佐藤拓朗[†1]

これまで筆者らは、次世代ネットワーク技術として知られる NDN を活用した先回りコンテンツ配信システムの提案を行っている。しかしながら、必ずしもその実証評価が十分ではなかった。そこで、本稿では、最大 50 人に協力して貰い、実際の営業線上にシステムを展開し、その性能評価を行った。その結果、最大同時に 50 人までのユーザに途切れない映像配信を提供できることが分かった。

Performance Evaluation of Proactive Content Caching for Mobile Video through 50-User Field Experiment

Kenji Kanai[†1], *Takeshi Muto*[†1], *Jiro Katto*[†1], *Shinya Yamamura*[†2], *Tomoyuki Furutono*[†2],
Takafumi Saito[†1], *Hirohide Mikami*[†1], *Kaoru Kusachi*[†1], *Toshitaka Tsuda*[†1]
Wataru Kameyama[†1], *Yong-Jin Park*[†1], *Takuro Sato*[†1]

Providing robust content delivery service, such as Video on Demand, with efficient wireless resource usage is important for mobile users. To achieve this, we have proposed a proactive content caching scheme utilizing transportation systems (e.g., train.) Because our previous prototype is immature and has a scalability issue, in this paper, we introduce efficient transmission mechanism of Interest messages in order to improve throughput efficiency in mobile NDN. We develop two prototypes based on HTTP and NDN and then evaluate prototype performance by performing larger-scale field experiments on actual commercial railroad line. Evaluations conclude that our system can achieve high efficient content transfer and fully utilize wireless network bandwidth. Therefore, our system can provide high-quality video streaming for up to 50 users simultaneously.

1. Introduction

Providing robust content delivery service, such as Video on Demand, with efficient wireless resource usage is important for mobile users and mobile carriers. According to [1], Cisco forecasts that mobile data traffic, especially mobile video, will be the highest growth rate among any mobile applications. We expect that Named Data Networking (NDN) [2] is one of promising approaches to address this fact because of its in-network cache mechanism.

Because the original NDN implementations, such as NDNx [3] and NDN-JS [4], do not take into account user mobility, we have proposed a proactive content caching scheme that uses transportation systems [5, 6] which is an application layer approach to manage mobile video. This scheme consists of mainly two phases; proactive caching by stations and wireless video streaming in a train. In [5], we find that our scheme can provide higher-speed, higher-reliable and lower power consumption video delivery without playback interruption (i.e. freezing) than traditional streaming on cellular networks. However, in [6], our previous prototype shows low performance in NDN content delivery because our previous prototype applies original NDNx's congestion control that emulates legacy TCP-Tahoe. Our previous prototype also has a scalability issue in the number of users to be accommodated (i.e., a single user we assumed.)

In this paper, in order to improve wireless throughput efficiency in mobile NDN, we introduce efficient transmission mechanism of Interest messages. We develop two different prototypes based on HTTP and NDN, and then evaluate the system performance by performing larger-scale field experiments. We use actual train vehicles of commercial railroad line, build a high-speed backbone network to connect railroad stations, and install cache servers and Wi-Fi access points at each station and in a train. Then, a maximum of 50 users participate in the experiment to download video contents simultaneously.

2. Related work

2.1 Named Data Networking

In NDN delivery, two message types called Interest and Content Object are exchanged. Interest messages are used to request data by specifying the content chunk name. Such messages also contain a name prefix to limit the data that is most suitable from the collection of the same prefix. Content Object messages are used to supply data. Such messages are mainly composed of a name, publisher, and chunk of data; they also contain data payload, cryptographic signature, publisher

---

†1 早稲田大学
Waseda University
†2 富士通九州ネットワークテクノロジーズ

identification, and other information about signing. In communication, a data consumer broadcasts an Interest message over all available connectivity, and any nodes with the content that satisfies the Interest must respond with at most one Content Object message. In order to satisfy the Interest, the Content name in the Interest message has to be a prefix of the Content name in the Content Object message. One of the key features of NDN is the router content caching mechanism. When content goes through a NDN router, it caches itself. This feature provides congestion reduction and fast content delivery because clients can fetch content from the nearest cache rather than the origin content server. Different implementations of NDN are known as NDNx and NDN-JS.

2.2 MPEG-Dynamic Adaptive Streaming over HTTP

MPEG-DASH [7] is a streaming technology, standardized by MPEG that is capable of a continuous playback by changing the bitrate dynamically and adaptively while observing the network bandwidth. Video content is encoded into multiple bitrates and resolutions, and is divided into segments. URLs of each segment are written in the Media Presentation Description (MPD), which also has information on encoded bitrates, resolution, minimum buffer time, etc. Clients access this MPD file at the start of the streaming session, and refer to it in order to select the optimal bitrate according to the network conditions. Note that every segment can be accessed individually by the client via HTTP GET requests.

DASH-JS [8] is one of the libraries integrated for the DASH standard. This is a JavaScript-based implementation that uses the Media Source API from Google's Chrome browser, and has no need for further plugins. Furthermore, this feature allows DASH playback on multiple devices and makes it easier to develop mobile applications.

2.3 Proactive caching

Many studies have been conducted on proactive caching schemes. An opportunistic content pushing scheme was proposed in [9] that predicts the moving routes of roaming users and pre-locates content to Wi-Fi spots along their routes. Lobzhanidze [10] proposed a proactive video caching scheme based on video popularity prediction using a topic modeling tool called Latent Dirichlet Allocation and a frequent pattern mining algorithm called Apriori. Vasilakos [11] proposed a method for Information Centric Networking (ICN) called Selective Neighbor Caching that enhances seamless mobility in ICN, which selects an optimal subset of neighbor proxies that consider user mobility behavior. Similar to [11], Rao [12] proposed a proactive caching approach for seamless user-side mobility support in NDN.

3. Proactive content caching utilizing transportation systems

3.1 Methodology

The main feature of our approach is that mobile users no longer need to access content servers directly in order to download video contents. This can reduce delay and packet losses than traditional streaming on cellular networks. We place servers with cache capability (e.g., NDN) in every trains and stations. Every station server connects to content servers via high-speed wired backbone networks, and also connects to a train server via high-speed wireless networks. Our approach has two phases that are "Proactive caching phase" and "Video streaming phase." An overview of our scheme is shown in Figure 1.

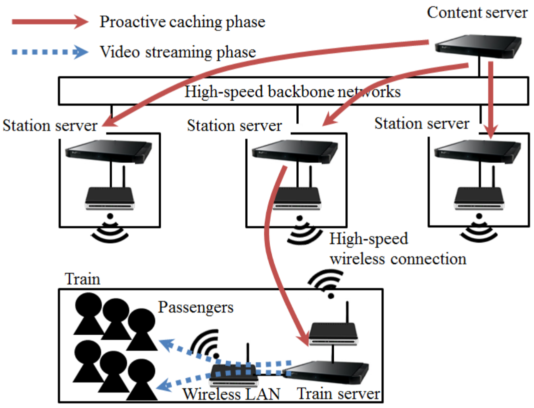

Figure 1: Overview of our system.

Proactive caching phase indicates how to deliver contents to the nearest cache router/server, which is a train server in this case, using IP and/or NDN protocol. The detail of this phase is shown in [5, 6]. When the content server receives a requirement, the required content is divided into several segments and then proactively delivered to station servers according to train's time-table. Content quality and the number of content segments should be adaptively changed by the train's time-table and network conditions. Once the trains arrive at the stations, they download pre-fetched content segments from the station servers via high-speed wireless networks, such as millimeter-waves, until leaving the stations.

Then, the received video segments are streamed to the users on trains via high-speed wireless networks such as wireless LANs. We call this phase "Video streaming phase." Because the users can access the nearest cache server/route in a single-hop wireless network, our approach can achieve to provide higher QoS mobile video service compared with traditional video streaming on cellular networks.

3.2 Smart scheduler

The key function is a delivery scheduler that we call "smart scheduler." Smart scheduler determines content quality and the amount of content segments and also selects delivery locations and timing. Adaptations for content quality and number of segments are important for robust video streaming. Therefore, the video content is encoded by multiple bitrate, divided into several segments, and encapsulated by MPEG-DASH. In MPEG-DASH, one of the layers in the video quality hierarchy is called "representation," and each representation consists of several "segments" that correspond to a few seconds of video content. Thus, the smart scheduler simply selects the representation by referring to calculated content bitrate. The content bitrate is computed by fulfilling the three conditions named "Proactive caching," "Continuous playback," and "Smooth streaming."

Similarly, the number of segments is also calculated by the train's time-table in order to provide continuous video playback. Delivery locations can be predicted by the current train location and time-table. The more detail of smart scheduler is shown in [5].

3.3 Communication protocols

In order to implement our scheme, we need to determine the communication protocols. With regards to NDN architecture, there is no standard protocol for control packets (i.e., signaling). Therefore, we use the IP for control packets, and use IP (HTTP) or NDN for data packets. The basic procedures of our prototype are demonstrated in Figure 2, and summarized as follows:

1. *Signaling (IP):* once a train server or a content server receives a request message from a user, the servers forward the request to the smart scheduler. Then, the smart scheduler collects the train's time-table from a database and determines content delivery schedule. This scheduler also collects information of network conditions from stations and trains periodically and stores them into another database.
2. *Proactive caching (IP and/or NDN):* the station servers download video segments by referring to the delivery schedule. Once the station servers receive video segments via IP and/or NDN networks, they cache the received segments.
3. *Video segment download (IP and/or NDN):* when the train arrives at a station, the train server download video segments via IP and/or NDN networks. Therefore, the train server can obtain video segments.
4. *Video Streaming (IP and/or NDN):* finally, the user devices receive video data via the IP or NDN networks.

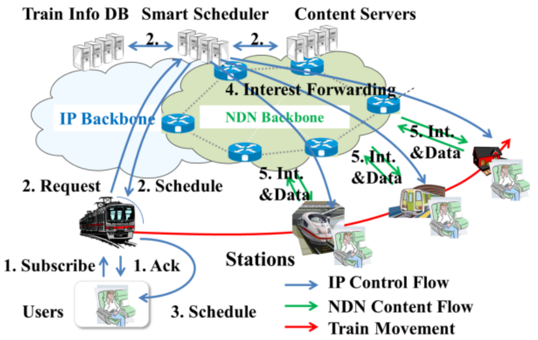

Figure 2: Basic protocol procedures of our prototype.

3.4 Implementation

We develop two prototype implementations based on NDN and HTTP aiming to field experiments described in Sections V. The NDN-based prototype is written in JavaScript, and we call it DASH-NDN-JS [9]. DASH-NDN-JS integrates DASH-JS and NDN-JS to allow playback of DASH content on web browsers. This is convenient for large-scale experiment because participants can use browsers they are familiar with. In order to increase throughput efficiency by underlying TCP, Interest message aggregation and Secure Copy Protocol (SCP) are also used in the NDN and HTTP based prototypes for proactive caching phase, respectively. In addition, in order to perform fully automatic implementations, scripts are developed to switch Wi-Fi connection by checking signal strength and the train time-tables.

On implementation of the smart scheduler, we manage the train time-tables and network conditions by constructing data bases on an HTTP server, which also supports NDN transport. The smart scheduler itself is written in JavaScript in both prototypes.

4. Efficient content transfer in NDN

Efficient content transfer in NDN is particularly important for our system because, without cellular connections, the train server have to transfer large video data until left from a station. Because current NDN implementations, such as NDNx and NDN-JS, emulate a legacy congestion control, such as TCP-Tahoe, their network bandwidth utilization is insufficient [5, 6]. In this section, we introduce efficient transmission mechanism of Interest messages to improve throughput efficiency in NDN.

First, we send NDN packets over TCP, specifically CUBIC-TCP, which is implemented into the Linux kernel. It is well known that CUBIC-TCP provides higher-throughput and better link utilization than TCP-Tahoe. Because TCP provides reliable data transport capability by retransmission, we no longer worry about packet drops of Interest and Content Object messages.

Second, to fully utilize CUBIC-TCP's efficient congestion control, Interest messages are aggregated and transmitted in a single burst per station. In the current video streaming application with NDN capability, Interest messages are transmitted intermittently. However, TCP connection per Interest is inefficient, and Interest aggregation contributes to throughput improvement. This is also true for the Content Object case, and we also attempt video data aggregation in the HTTP based video delivery using SCP.

Third, to reduce unnecessary Interest messages, we allow the smart scheduler to calculate the amount of Interest messages that the consumer needs to send. In current NDN, a consumer does not know how many Interest messages it should send because NDN divides video content into several chunks by its own rule (e.g., chunk size is 4096 Byte in NDN-JS). For this reason, the consumer continues to send Interest messages until it can receive Content Objects. Therefore, our smart scheduler calculates the amount of Interest messages by referring to chunk size and content size in advance. In MPEG-DASH, content size can be managed by the MPD file.

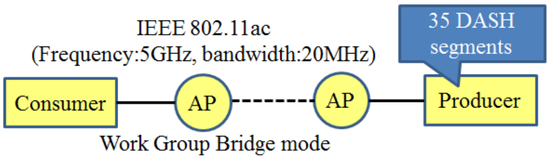

Figure 3: An image of experimental environment for bandwidth utilization evaluation.

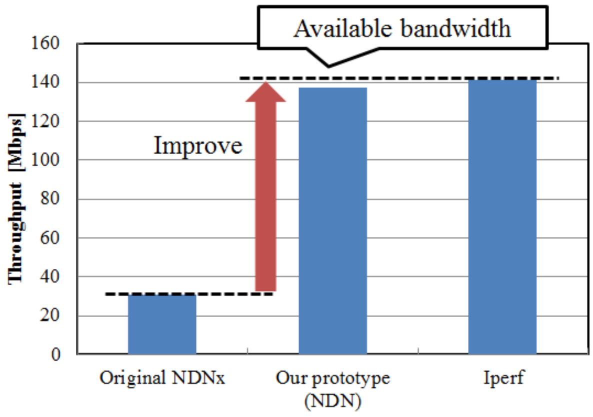

Figure 4: Throughput comparisons of original NDNx and our NDN-based prototype. Result of iperf shows the available bandwidth.

We then evaluate and compare link utilizations between the original NDNx and our system over a wireless network. An evaluation environment is shown in Figure 3. A consumer is connected to a producer via IEEE 802.11ac wireless LAN. We use two Cisco Aironet 3700 Access Points (APs) that function in work bridge mode. The channel frequency and bandwidth are 5GHz and 20MHz, respectively. We place 35 DASH segments that equal 300 MB in total, and the segments are divided into several chunks by NDNx. We also observe the available network bandwidth using iperf. Then, as shown in Figure 4, we compare the throughputs of original NDNx, our proposal, and the available bandwidth given by iperf. As shown in Figure 4, our proposal can achieve higher throughput than original NDNx and almost reach the available bandwidth.

Figure 5: Snapshots of installed equipment at a station.

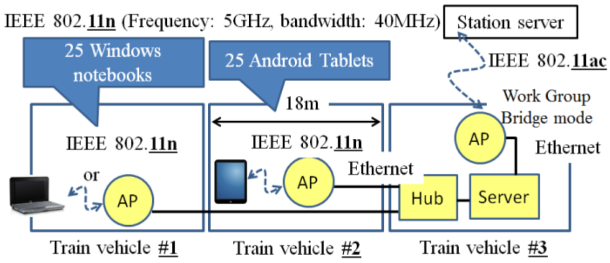

Figure 6: Experiment setup inside train.

5. Performance evaluations through 50-user field experiments

5.1 Field experiment environment

We evaluate the performance of NDN and HTTP based prototypes by performing a larger-scale field experiment using a commercial railroad line, called Keikyu Daishi line, in Kawasaki, Japan. This line has seven stations and requires 10 min for a one-way trip. We select three train stations from seven, Minatocho, Suzukicho, and Kawasakidaishi. As shown in Figure 5, we set a station server, 802.11ac AP, and backbone router in a box, and install the box to the inbound side platform of each station. Inside a train, as shown in Figure 6, we place a train server, 802.11ac AP, and two 802.11n APs, and connect them by LAN cables. The 802.11ac AP is used to communicate with stations, and the two 802.11n APs are used to deliver video content to train passengers. Each station server is connected via high-speed optical backbone networks. This backbone network is constructed by NTT East Corporation, and it is also connected to Waseda University to monitor the backbone. The station servers at Minatocho (outbound case) and Kawasakidaishi (inbound case) also perform content servers. Each AP is assigned different channels to avoid radio wave interference. We randomly select 25 different 2 K videos from YouTube. Each video is encoded by H.264/AVC and divided into several segments. The encoding rates are 100, 200, 400, 600, and 800 kbps and 1 Mbps, and the segment length is 2 sec, which is an original DASH parameter.

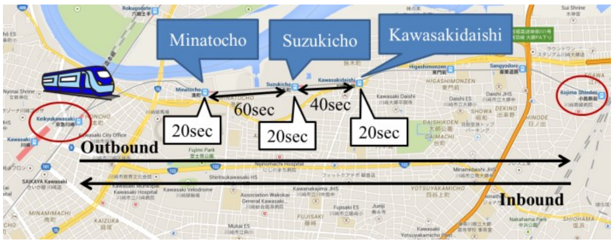

Figure 7: Train movement map for Keikyu Daishi line in Kawasaki, Japan.

5.2 Evaluation scenarios

The train movement schedule is shown in Figure 7. The train stops 20 sec at each station, and requires 40 sec to 1 min to move between stations. A total of 50 participants are separated into two groups; the first group consists of 25 Android users, and the second group consists of 25 Windows users. The two groups board different trains, each of which has one 802.11n AP. Up to 50 users request video content simultaneously before the train arrives at Minatocho (outbound) and Kawasakidaishi (inbound). We employ the following three different evaluation scenarios:

A) *25 users and one AP:* in this scenario, 25 Android users connect to the 802.11n AP located on train #2.

B) *50 users and two APs:* in this scenario, 25 Android users connect to the AP located on train #2, and 25 Windows users connect to the AP located on train #1.

C) *50 users and one AP:* in this scenario, all users connect to the AP located on train #2.

Because we prepare 25 videos, Windows users request the same videos as the Android users.

We can conduct six round trips for the field experiment, which requires approximately three hours. We develop two prototypes: HTTP-based and NDN-based (DASH-NDN-JS), both of which runs automatically without manual operations. We have three scenarios and two directions (outbound and inbound). Therefore, we attempted twelve (2×3×2) measurements in total according to the prototypes, scenarios, and train directions. All packets are captured by Wireshark.

5.3 Performance characteristics of proactive caching between the train and stations

First, we observe throughput characteristics of proactive caching between the train and stations for our two prototypes. Figure 8 shows an example of throughput characteristics between the train and station servers. In this figure, throughputs of the outbound case become lower and fluctuate because the distance between the train and station APs is farther than the inbound case. From this figure, we can recognize that both prototypes perform well and approach the iperf results that indicate available bandwidth. This achievement is provided by the video data aggregation at the content server for the HTTP-based prototype, and by Interest message aggregation into a single burst for the NDN-based prototype. CUBIC-TCP also contributes to maintain high bandwidth utilization. We can also notice that most data transfers by the prototypes are completed before a train left the station. This is because our prototype designs are conservative in order to avoid experiment failures. The results encourage us to send more video data in our future design.

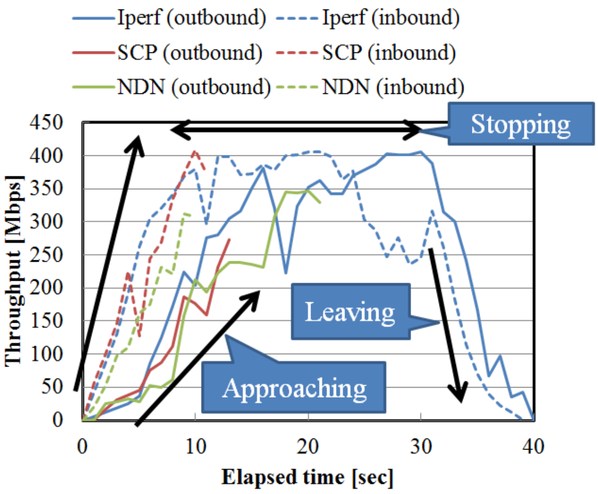

Figure 8: Example of throughputs for proactive caching observed between train and station servers at Suzukicho station.

5.4 Performance characteristics of video streaming

Second, we observe the performance for video streaming in the train. An average received data size from the train server, with the exception of Ack packets, is shown in Figure 9. As shown in this figure, both HTTP and NDN show similar characteristics in middle traffic load scenarios, such as “25 users one AP” and “50 users and two APs” cases. However, in the high traffic load scenario (50 users one AP), we can notice that received data sizes are reduced. This is because the capacity of IEEE 802.11n that we use inside a train is limited. In addition, in this high traffic load scenario, we can recognize that Windows users obtain lower performance than Android users for both prototypes. This is because Windows users attempt to access the AP that is placed at train #2 where Android users are sitting down as shown in Figure 6. We also notice that the NDN-based prototype degrades more severely than the HTTP-based. This is because the NDN needs more overheads (i.e. Interest messages) to get video contents, and the packet collisions might be triggered many times in wireless LAN.

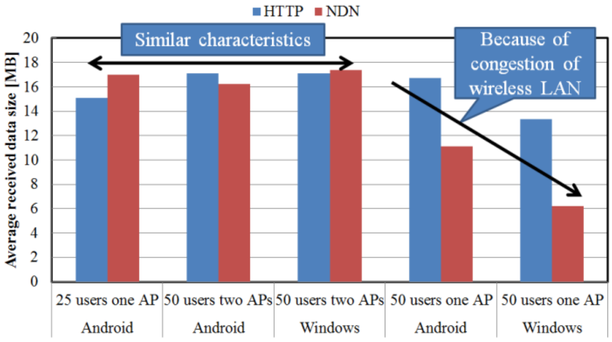

Figure 9: Averaged received data sizes, with the exception of Ack packets, from train server to client terminals.

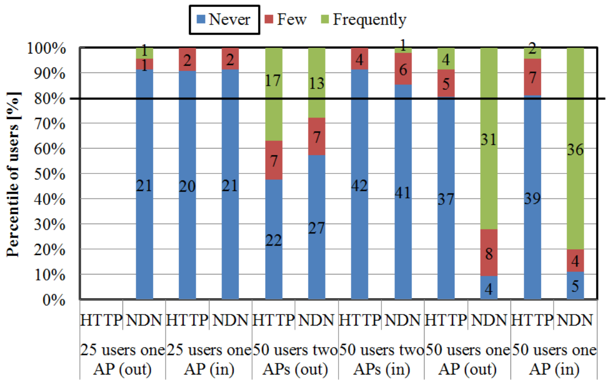

Figure 10: QoE evaluation results of video playback impression (in: inbound, out: outbound).

5.5 QoE characteristics of our prototypes

Finally, in order to confirm the subjective effectiveness of our systems, we evaluate QoE using questionnaires. Among the many questions we provide, Figure 10 shows the results of the video playback interruption experienced by users. We employ three choices; "never freezing," "few times," and "frequently freezing." The results show that more than 80% of users experience smooth video playback without interruption in most scenarios. When the train fails to obtain video segments from the stations or the single AP must accommodate 50 users, some trials indicate that video does not play back smoothly and many users experience video that froze frequently. In the high traffic load case (i.e., 50 users one AP), we need more investigations why NDN-based video delivery leads to low quality.

6. Conclusions

In this paper, we introduced a proactive content caching scheme that uses transportation systems for robust video streaming and efficient utilization of wireless resources. We developed two prototypes based on HTTP and NDN and evaluated QoS and QoE characteristics for both prototypes by performing larger-scale field experiments, where maximally 50 users downloaded video contents simultaneously. Evaluations concluded that our system can provide video streaming without interruption for up to 50 users. In near future, we will explore efficient content delivery in NDN and reduce overhead of NDN-based wireless video to improve its QoS. We will also extend our framework to automobile cases.

Acknowledgements This paper has been supported by MIC's Project for Promotion of Advanced Communication Applications Development and also supported by the FP7/NICT EU-JAPAN Green ICN project. The authors thank to Keikyu Corporation, a train operating company in Japan, who provided us field experiment environment. The authors also thank to NTT East Corporation, who provided us high-speed backbone network for the field experiment.

References

1) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019.
2) L. Zhang, et al. "Named Data Networking," ACM SIGCOMM Computer Communication Review, vol.44, Issue 3, pp.66-73, Jul. 2014.
3) NDN project [online]: http://named-data.net/
4) W. Shang, et al. "NDN.JS: A JavaScript Client Library for Named Data Networking," Proc. IEEE INFOCOM Workshop 2013, pp.399-404, Apr. 2013.
5) K. Kanai, et al. "Proactive content caching utilizing transportation systems and its evaluation by fied experiment," Proc. IEEE Globecom 2014, pp. 1399-1404, Dec. 2014.
6) T. Muto, et al. "Implementation Evaluation of Proactive Content Caching using DASH-NDN-JS," Proc. IEEE WCNC 2015, Mar. 2015.
7) I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," IEEE Multimedia, vol.18, Issue 4, pp.62-67, Apr. 2011.
8) B. Rainer, et al. "A seamless web integration of adaptive HTTP streaming," in Proc. Singla Processing Conference (EUSIPCO) 2012, pp.1519-1523, Aug. 2012.
9) T. Han, et al. "Opportunistic content pushing via WiFi hotspots," Proc. IEEE IC-NIDC 2012, pp.680-684, Sept. 2012.
10) A. Lobzhanidze, et al. "Proactive caching of online video by mining mainstream media," Proc. IEEE ICME 2013, pp.1-6, July, 2013.
11) X. Vasilakos, et al. "Proactive selective neighbor caching for enhancing mobility support in information-centric networks," Proc. ACM ICN '12, pp.61-66, Aug. 2012.
12) Y. Rao, et al. "Proactive caching for enhancing user-side mobility support in named data networking," Proc. IMIS 2013, pp.37-42, July, 2013.