Interest アグリゲーションと再生バッファ制御を利用した NDN における省電力ビデオストリーミング

石津裕也[†1]　金井謙治[†1]　甲藤二郎[†1]　中里秀則[†1]　広瀬真里枝[†2]

概要：無線通信において、動画配信における省電力化は重要な要素である。本稿では、NDN(Named Data Networking)における省電力動画ストリーミングを提案する。この提案では、Interest アグリゲーションによるスループットの向上と再生バッファ制御によるオーバーヘッド電力の削減の二点に焦点を当てている。消費電力の評価にはハードウェア消費電力計測ツールを利用する。本提案は、Interest アグリゲーションにより約 50%の消費電力量を削減し、再生バッファ制御により最大で約 37%の消費電力量の削減を可能とすることを確認した。

キーワード：省電力, NDN/CN, MPEG-DASH, 再生バッファサイズ制御

Energy-Efficient Video Streaming over Named Data Networking using Interest Aggregation and Playout Buffer Control

YUYA ISHIZU[†1]　KENJI KANAI[†1]
JIRO KATTO[†1]　HIDENORI NAKAZATO[†1]　MARIE HIROSE[†2]

Abstract: In wireless networks, it is important to realize energy-efficient video delivery. To do this, we introduce energy-efficient video streaming over named data networking (NDN). In our proposed approach, we focus on two areas, namely, to improve the throughput performance by Interest aggregation, and to reduce the overhead energy using playout buffer-size control. We evaluate the power savings realized by our method using a hardware power-measurement tool. Our results show that Interest aggregation can realize an approximately 50% reduction in energy compared to conventional NDN implementation, and a large playout buffer size can reduce the energy by approximately 37% compared to the case of a small playout buffer size.

Keywords: energy efficient, NDN/CCN, MPEG-DASH, playout buffer-size control

1. Introduction

Recently, there has been an increased demand for energy-efficient video delivery in wireless networks because of the poor battery life of current mobile devices that run power-hungry multimedia applications, such as video streaming, with high quality [1].

Content-centric networking [19]/named data networking [5] is considered to be a promising approach for future network architecture because of its in-network caching mechanism, which distributes and caches video content on edge routers. Because of this mechanism, CCN/NDN is able to provide energy-efficient video delivery because its caching mechanism can reduce network latency and network traffic [2]. However, current CCN/NDN implementations exhibit low performance [3], and there have been few studies regarding CCN/NDN-based energy-efficient video transfer on the client-side.

Therefore, in this paper, we propose an energy-efficient video-delivery mechanism in NDN. Our approach involves the use of two key technologies, namely, efficient content transfer and playout buffer-size control on the client-side. Efficient content transfer in NDN reduces the transmission time when downloading video content, and it reduces energy consumption because energy consumption during transmission depends greatly on the transmission time. Meanwhile, the use of playout buffer control can reduce unnecessary energy consumption in video streaming.

In our study, we implement these methods into NDN-JS [4], which is a JavaScript-based implementation of NDN, and we evaluate the performances of these methods in real environments.

The rest of this paper is organized as follows. Section II presents related work. In Section III, we propose efficient video delivery in NDN, while in Section IV, we propose energy-efficient video streaming in NDN by controlling the playout buffer size. Finally, we conclude the paper in Section V.

†1 早稲田大学　基幹理工学研究科　情報理工・情報通信専攻
Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo,
†2 パナソニック　アドバンストテクノロジー(株)
Panasonic Advanced Technology Development Co., Ltd., Hiroshima,

2. Related work

2.1 Content-Centric Networking/Named Data Networking

CCN/NDN has been previously proposed as a candidate for future network architecture in order to adapt to recent changes in current Interest usage [5]. Most Internet users are primarily concerned about the content itself (i.e., the content name), and not the actual location in which the content is stored (i.e., IP address or URL). However, this approach is unsuitable for current Internet architecture, which incorporates host-to-host networks, and users can request content by specifying the IP address (i.e., URL) of a resource. Therefore, future network architecture, CCN/NDN, requires the use of information-centric networks, and users will be expected to request content by name, and not URL.

In CCN/NDN delivery, two message types are exchanged, namely *Interest messages* and *content object messages.*

- *Interest* messages are used to request content by specifying that content's name, which generally has a hierarchical structure.
- *Content object* messages are used to supply data. They contain the identifying content name, a cryptographic signature, and the data payload.

In communication, a data consumer broadcasts an Interest message over all available links, and nodes that have content satisfying the Interest message 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 CCN/NDN is the router content-caching mechanism. When content goes through a CCN/NDN router, it caches itself. This feature minimizes congestion and enables fast content delivery because clients can fetch desired content from the nearest cache rather than the origin content server.

There are several open-source applications that employ CCN/NDN, CCNx [7], NDNx [6], and NDN-JS. CCNx and NDNx are written in C language as the basis for architecture research. NDN-JS is the first JavaScript implementation of NDN [8], and it supports the basic NDN functions of content fetching and publishing by using Interest/content object exchange. It is wire-format compatible with CCNx and NDNx routing and forwarding.

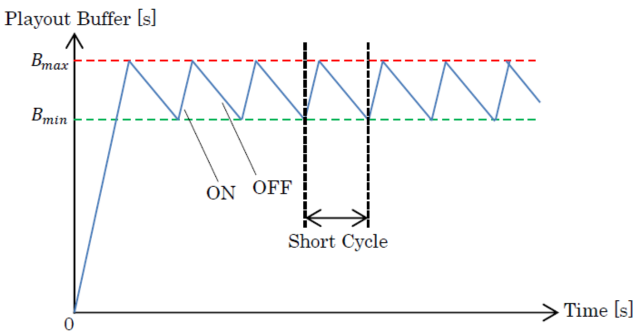
2.2 MPEG Dynamic and Adaptive Streaming over HTTP

MPEG-DASH is a very popular video streaming technology that uses HTTP [9]. It provides continuous video playback by observing the network bandwidth and dynamically changing the video content's bitrate. Video content is defined by a hierarchical structure called "representation," which is explained as follows. Video content is encoded with multiple bitrates and resolutions, and it is also divided into several segments to enable constant playback time (the original parameter is 2s). This hierarchical structure is managed by media presentation description (MPD). The client accesses this MPD file at the beginning of the streaming session, and selects the optimal bitrate by referring to the MPD according to the network bandwidth.

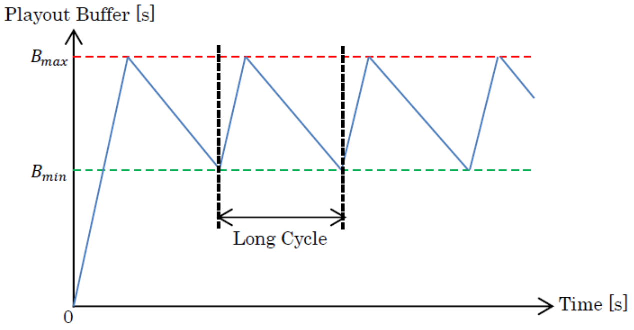
2.3 Playout Buffer Strategy for Video Streaming

Much research has been conducted on the client-side playout buffer strategy in video streaming. In [10], the authors revealed the buffering mechanisms on YouTube and Netflix by capturing and analyzing these traffic patterns, and they defined three types of buffering mechanisms, namely "no ON-OFF Cycles," "short ON-OFF Cycles," and "long ON-OFF Cycles." Note that the "ON" phase means that the video segments are downloaded from content servers, and the "OFF" phase refers to the decoding of video segments for playback and display purposes.

The no ON-OFF cycle strategy is considered to be a traditional buffering strategy. In this strategy, entire video segments are obtained during the start-up phase. In the short ON-OFF cycle strategy, a small number of video segments are downloaded and decoded repeatedly over a short period of time. In other words, there is a high frequency of switching between the OFF and ON cycles. On the other hand, in the long ON-OFF cycle strategy, many video segments are downloaded and decoded repeatedly over a long period. Because of this, the frequency of switching between the OFF and ON cycles is lower than in the previous case. Images of these buffering strategies are shown in Figure 1.These strategies can be controlled by the values of the maximum and minimum buffer size (B_{max} and B_{min}).



(a) One of the examples of the "short ON-OFF cycle" strategy.



(b) One of the examples of the "long ON-OFF cycle" strategy.

Figure 1: Examples of client-side playout buffer strategies in video streaming.

According to [10], these strategies depend on a combination of the browser and container employed. For example, when Google Chrome is used, YouTube for HTML5 is classified as a long ON-OFF cycle strategy, while Netflix for Silverlight is classified as a short ON-OFF cycle strategy. The no ON-OFF cycle strategy is used by YouTube for Flash HD regardless of the browser used. In [11], the authors have also defined a buffering strategy for YouTube called the "block sending algorithm," which is similar to the long ON-OFF cycle strategy. In [12, 13], the authors have also defined the short and long ON-OFF cycle strategies as "Zippy pacing" and "Sawtooth pacing," respectively.

3. Efficient Video Delivery over NDN

3.1 Method

Efficient content transfer is important in order to provide green video delivery because energy consumption depends greatly on the throughput performance. Current NDN implementations, such as NDNx and NDN-JS, emulate legacy congestion control techniques such as TCP-Tahoe, which has an insufficient network utilization bandwidth. In this section, we introduce efficient content-transfer mechanisms for Interest message aggregation to improve the throughput performance of NDN. Our approach has three elements.

First, we send Interest messages over TCP, specifically CUBIC-TCP, which is implemented in a Linux kernel. By sending Interest messages over TCP, we no longer consider packet drops of Interest and data packets on the application layer because TCP provides reliable data transport using the packet retransmission mechanism.

Second, in order to maximize the use of the efficient congestion control of CUBIC-TCP, Interest messages are aggregated and transmitted in a single burst. In the current NDN implementations, Interest packets are transmitted intermittently. Therefore, the current NDN implementation is not able to fully use CUBIC-TCP, and the throughput is lower than in our proposed method.

Third, in order to reduce the number of unnecessary Interest messages, the consumer (i.e., client) should be aware of the size of the desired content. In NDN, all content is divided into several segments, each of which has a constant value (e.g., 4096 Kbytes by default), and each chunk has unique name. Using this mechanism, the consumer has to transmit several Interest messages to obtain the desired content. However, in current NDN, the consumer is unaware of the number of Interest messages required to obtain the content. Therefore, the consumer has to send superfluous Interest messages. If the consumer does not receive data packets that correspond to the redundant Interest messages for a fixed time, the consumer stops sending Interest messages. To overcome this problem, an MPD file, which determines the content information in MPEG-DASH, should have the video segment size, and the consumer can then become aware of the size of the desired content. By referencing this MPD file, the consumer can calculate the number of Interest messages using Equation (1). Note that I, $S_{content}$, and S_{chunk} are the number of Interest messages, the desired content size, and the chunk size, respectively.

$$I = {}^{S_{content}}/_{S_{chunk}} \tag{1}$$

3.2 Experiment Environment

Then, we evaluate and compare original NDN applications and our proposed method over a wired network. The experiment environment used is shown in Figure 2. A consumer and producer are connected via a 100 Mbps wired connection. The producer pre-installed three videos, each of which was divided into several 4096-Kbyte segments. We used three videos with sizes of 11.4 MB, 44.5 MB, and 231.6 MB, respectively. We observed the available network bandwidth using iperf [14]. In addition, we also evaluated the energy consumption during transmission using a hardware power-measurement tool called "Watts UP? PRO" [15]. We performed three trials, and the results show the average of these trials.

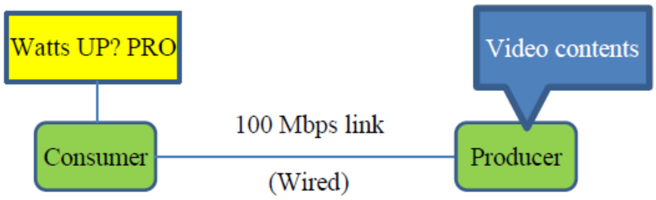


Figure 2: Experiment environment for bandwidth-utilization evaluations.

3.3 Results

Figures 5 and 6 show comparisons of the results of the link utilization and energy consumption for communication between the original NDN application and our proposed method, respectively. As shown in Figure 3, our proposed approach has a higher throughput than the original NDN application in the three video cases. In addition, as shown in Figure 4, our proposed approach realizes more energy savings during communication than the original NDN application because energy consumption depends greatly on the transmission time. Therefore, by realizing efficient content transfer, we can realize energy-efficient content delivery.

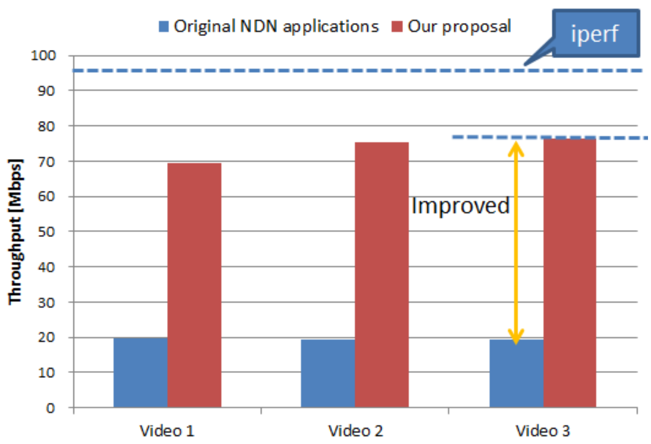


Figure 3: Throughput comparisons between original NDN application and our proposed method for three different videos.

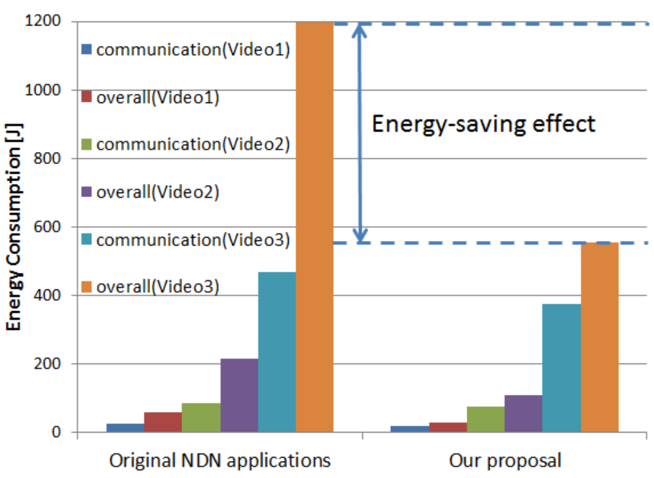


Figure 4: Energy consumption comparisons between original NDN application and our proposed method for three different videos.

4. Energy-efficient Video Streaming by Controlling Playout Buffer

In order to reduce the energy consumption for video streaming in wireless networks, we propose an energy-efficient buffering strategy on the client-side. In this paper, we focus on the access network, as shown in Figure 5. As shown in this figure, we assume that the video content is distributed and stored in an edge router in the same manner as in NDN, and the client transmits the aggregated Interest packets, as described in the previous section, to the nearest edge router in order to download the video segments. Although we assume video delivery in NDN, we believe that our approach can also provide energy-efficient video delivery for traditional video streaming (e.g., CDN).

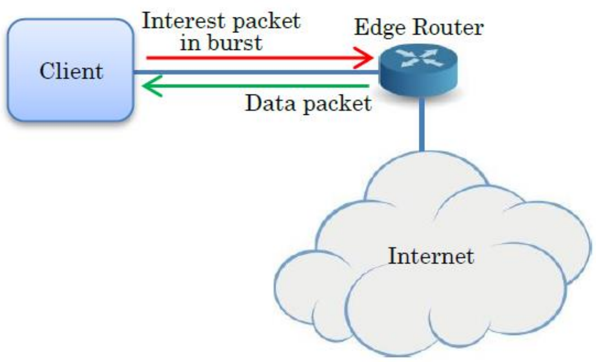


Figure 5: Assumed network topology.

4.1 Method

In [18], the authors reported that there is an overhead energy required for communication, which is called the "maintenance energy." This energy is used when a device communicates via Wi-Fi. An illustration of the maintenance energy is shown in Figure 6. As shown in this figure, the energy consumption is greatest when the device transmits and receives a packet. However, this peak energy does not immediately decrease, but gradually decreases to the idling energy (P_{idle}) after ending the communication. Based on this phenomenon, we expect that intermittent transmissions (i.e., when small packets are transmitted for short durations of time) would lead to inefficient transmission in terms of energy consumption because the overhead maintenance energy is used after every transmission.

This phenomenon is also true for the client-side playout buffer strategy in video streaming. The client needs to buffer several video segments in order to achieve smooth playback without video playback interruption. As described in Section II.D, a larger playout buffer size, which is similar to a long ON-OFF cycle, achieves a longer idle time because the client is able to playback video segments for longer periods of time. In such cases, the larger playout buffer size decreases the energy consumption lost to idling energy (P_{idle}), as shown in Figure 7. On the other hand, if the playout buffer size decreases, similar to the short ON-OFF cycle, the client remains idle for a shorter period, and has to continuously obtain video segments from a content server. In such a case, the energy consumption cannot fully decrease to the idling energy (P_{idle}), and maintains a high value, as shown in Figure 7. Therefore, controlling the playout buffer size is a key factor in the realization of energy-efficient video delivery. In our proposed method, in order to provide energy-efficient video delivery in wireless networks, we used the long ON-OFF cycle as a playout buffer strategy. Moreover, as shown in Figure 7, realizing energy savings by using the idle state, and additional energy by using a long transmission are illustrated in areas "1" and "2," respectively. These two areas show the trade-off characteristics. As area 1 increases relative to area 2, the client device can save energy, and vice versa.

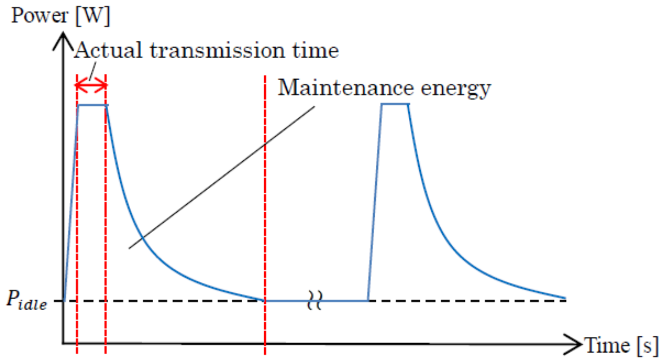


Figure 6: Image of energy-consumption behavior for wireless communication. The maintenance energy is defined as the overhead energy in [18].

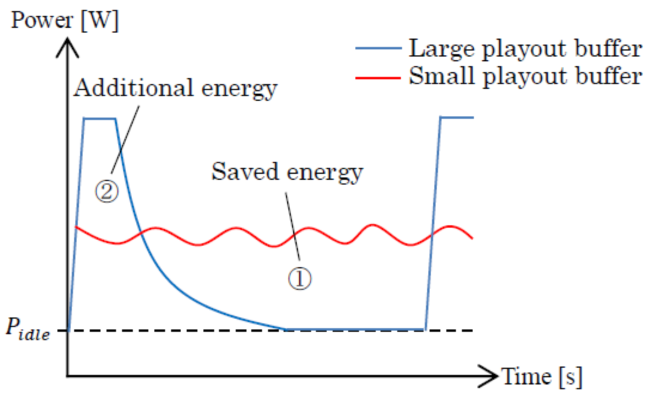


Figure 7: Image of energy-consumption characteristics for different playout buffer sizes.

4.2 Experiment Environment

We evaluated and compared the energy consumption characteristics for different playout buffer strategies in a real environment. In order to download video segments as in the NDN, we installed NDNx and NDN-JS on the producer and consumer, respectively. The experiment environment is shown in Figure 8. In this evaluation, we connected the producer to the AP via a wired connection, and we connected the IEEE 802.11ac AP to the consumer via wireless connection. We used a Cisco Aironet 3700i AP and a note PC as the consumer device. We observed the available network bandwidth using iperf, and it was found to be approximately 295 Mbps. The video used was a 4K video clip called "Tears of Steel [16]," and we pre-encoded it by H.264/AVC using DASHEncoder [17] and chopped into several segments in the same manner as in DASH. The segment length was 2s. The encoding rates were 3 Mbps, 7 Mbps, and 20 Mbps. Although we used MPEG-DASH, the selected bitrate remained constant (i.e., no rate adaptation) because we evaluated only the pure effect of the playout buffer strategy. As shown in Table II, in order to use the playout buffer strategy, we set the minimum playout buffer size to 20s, and set the maximum playout buffer size to 22, 80, and 140s. This means that the idle time was 2, 60, and 80s, respectively. We measured the energy consumption for the consumer device using Watts UP? PRO. We ran the evaluations three times for each parameter, and the results show the averages obtained.

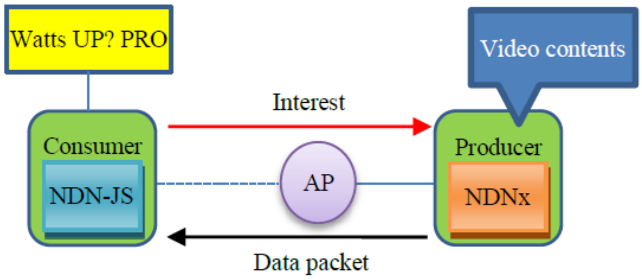


Figure 8: Experiment environment for energy-consumption evaluations using different playout buffer strategies.

TABLE I: PARAMETERS FOR PLAYOUT BUFFER STRATEGIES.

Playout buffer strategy	**Minimum buffer size**	**Maximum buffer size**
Short ON-OFF cycle	20 [s]	22 [s]
Long ON-OFF cycle	20 [s]	80 [s]
Long ON-OFF cycle	20 [s]	140 [s]

4.3 Results

Figure 9 shows a comparison of the energy-consumption results obtained for different playout buffer strategies. As shown in this figure, for most cases, the energy consumption decreased as the buffer size increased. In the case where the content bitrate was 20 Mbps, the energy consumption of the highest playout buffer size (140 s) was larger than that of the middle playout buffer size (80 s). This may be due to other components, such as physical memory. Generally, a device consumes energy as it reads and writes data into physical memory.

From this experiment, the long ON-OFF cycle can reduce the energy consumption by approximately 37% energy compared to the short ON-OFF cycle. In addition, the longer playout buffer size has a greater impact on energy-efficient video delivery regardless of the content bitrate.

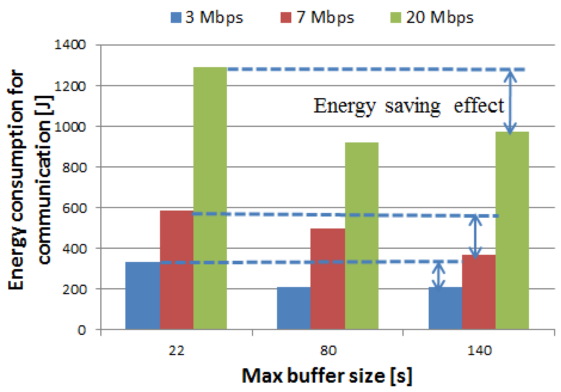


Figure 9: Comparison results of energy-consumption characteristics for different playout buffer strategies.

5. Conclusions

In this paper, we introduced energy-efficient video delivery over NDN using Interest aggregation and playout buffer-size control. We implemented these mechanisms into an NDN application called NDN-JS, and we evaluated the energy-consumption characteristics on a client device using a hardware power-measurement tool. We performed evaluations, which show that Interest aggregation can result in a significantly higher throughput of approximately 3.5 times compared to the original NDN implementation. This improvement can also lead to energy savings of 50% in video delivery. Next, by varying the playout buffer size, we can also realize energy-efficient video delivery. From our evaluations, the long ON-OFF cycle with a large playout buffer size can reduce the energy consumption by 37% compared to the case of a short ON-OFF cycle with a small playout buffer size. In the near future, we aim to perform a more in-depth analysis of the energy consumption, including hardware components such as the CPU and physical memory, and we will build an energy-consumption model for varying the playout buffer size in order to determine the optimal playout buffer size.

6. Acknowledgement

This paper has been supported by the FP7/NICT EU/JAPAN GreenICN Project and Grant-in-Aid for Scientific Research (A) (15H01684) of JSPS in Japan. The authors also thank to Mr. Kenichi Nakamura of Panasonic Corporation for his helpful comments.

Refferences

1) Ranoma Trestian, Arghir-Nicolae Moldovan, Olga Ormond, and Gabriel-Miro Muntean, "Energy Consumption Analysis of Video Streaming to Android Mobile Devices," IEEE NOMS 2012, pp. 444-452, Apr. 2012.
2) Nakjung Choi, Kyle Guan, Daniel C. Kilper, and Gary Atkinson, "In-Network Caching Effect on Optimal Energy Consumption in Content-Centric-Networking," Proc. IEEE ICC 2012, pp. 2889-2894, Jun. 2012.
3) Haowei Yuan and Patrick Crowley, "Experimental Evaluation of Content Distribution with NDN and HTTP," Proc. IEEE INFOCOM 2013, pp. 240-244, Apr. 2013.
4) Wentao Shang, Jeff Thompson, Meki Cherkaoui, Jeff Burke, and Lixia Zhang, "NDN.JS: A JavaScript Client Library for Named Data Networking," [online]:
http://web.cs.ucla.edu/~lixia/papers/NOMEN13-ndnjs.pdf
5) Lixia Zhang, Jeffrey Burke, Van Jacobson, Beichuan Zhang, K.C. Claffy, Christos Papadopoulos, Lan Wang, and Patrick Crowley, "Named Data Networking," ACM SIGCOMM Comput. Communication, pp. 66-73, Jul. 2014.
6) NDN Specification Documentation. [online]:
http://named-data.net/wp-content/uploads/2013/11/packetformat.pdf
7) CCNx. Ccnx software. [online]: http://www.ccnx.org/
8) Hongfeng Xu, Zhen Chen, Rui Chen, and Junwei Cao, "Live Streaming with Content Centric Networking," Proc. ICNDC 2012, pp. 1-5, Oct. 2012.
9) Christopher Mueller, Stefan Lederer, and Christian Timmerer, "ITEC - Dynamic Adaptive Streaming over HTTP," [online]:
http://www-itec.aau.at/dash
10) "Network characteristics of video streaming traffic," Proc. ACM CoNEXT 2011, Dec. 2011.
11) Shane Alcock and Richard Nelson, "Application flow control in YouTube video streams," ACM SIGCOMM Comput. Commun. Rev., vol. 41, Issue 2, pp. 24-30, Apr. 2011.
12) Kevin J. Ma, Radim Bartos, and Swapnil Bhatia, "Scalablity of HTTP Pacing with Intelligent Bursting," IEEE ICME 2009, pp. 798-801, Jun. 2009.
13) Kozo Satoda, Hiroshi Yoshida, Hironori Ito, and Kazunori Ozawa, "Adaptive Video Pacing Method Based on the Prediction of Stochastic TCP Throughput," Proc. IEEE GLOBECOM 2012, pp. 1944-1950, Dec. 2012.
14) iperf. [online]: https://iperf.fr/
15) Watts UP? PRO. [online]:
https://www.wattsupmeters.com/secure/products.php?pn=0
16) Test media. [online]. https://media.xiph.org/
17) DASHEncoder. [online]:
http://www-itec.uni-klu.ac.at/dash/?page_id=282
18) Niranjan Balasubramanian, Aruna Balasubramanian, and Arun Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," ACM SIGCOMM Conf. on Internet Meas. Conf., pp. 280-293, Nov. 2009.
19) Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, "Networking Named Content," in Proc. ACM CoNEXT 2009, pp. 1-12, Dec. 2009.