An NDN IoT Content Distribution Model With Network Coding Enhanced Forwarding Strategy for 5G

Kai Lei, *Member, IEEE*, Shangru Zhong, *Student Member, IEEE*, Fangxing Zhu, Kuai Xu, *Senior Member, IEEE*, and Haijun Zhang, *Senior Member, IEEE*

***Abstract*—The challenging requirements of fifth-generation (5G) Internet-of-Things (IoT) applications have motivated a desired need for feasible network architecture, while Named Data Networking (NDN) is a suitable candidate to support the high density IoT applications. To effectively distribute increasingly large volumes of data in large-scale IoT applications, this paper applies network coding techniques into NDN to improve IoT network throughput and efficiency of content delivery for 5G. A probability-based multipath forwarding strategy is designed for network coding to make full use of its potential. To quantify performance benefits of applying network coding in 5G NDN, this paper integrates network coding into a NDN streaming media system implemented in the ndnSIM simulator. The experimental results clearly and fairly demonstrate that considering network coding in 5G NDN can significantly improve the performance, reliability, and QoS. Besides, this is a general solution as it is applicable for most cache approaches. More importantly, our approach has promising potentials in delivering growing IoT applications including high-quality streaming video services.**

***Index Terms*—Content delivery, internet-of-things (IoT) streaming system, named data networking (NDN), network coding, probability multipath forwarding.**

Manuscript received October 1, 2017; revised November 12, 2017; accepted November 26, 2017. Date of publication December 8, 2017; date of current version June 1, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61471025 and Grant 61771044, in part by the Young Elite Scientist Sponsorship Program by CAST (2016QNRC001), in part by the Research Foundation of Ministry of Education of China & China Mobile (MCM2018-1-8), Beijing Municipal Natural Science Foundation (L172025), and the Fundamental Research Funds for the Central Universities (FRF-GF-17-A6), and in part by the Shenzhen Key Fundamental Research Projects under Grant JCYJ20160330095313861 and Grant JCYJ20170412151008290. Paper no. TII-17-2327. *(Corresponding author: Haijun Zhang).*

K. Lei, S. Zhong, F. Zhu are with the Shenzhen Key Laboratory for Cloud Computing Technology Applications, Institute of Big Data Technologies, School of Electronics and Computer Engineering, Peking University, Shenzhen 518055, China (e-mail: leik@pkusz.edu.cn; shangru@sz.pku.edu.cn; 914332429@qq.com).

K. Xu is with the School of Mathematical and Natural Sciences, Arizona State University, Tempe, AZ 85281 USA (e-mail: kuai.xu@asu.edu).

H. Zhang is with the Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China (e-mail: haijunzhang@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2781372

I. Introduction

In THE fifth-generation (5G) networks, the emerging applications [e.g., internet-of-things (IoT) and media streaming] with large scale users lead to the explosive growth of traffic. In 2021, IP video traffic will be 82% of all consumer Internet traffic, up from 73% in 2016, and content delivery network traffic will deliver nearly 71% of all internet video traffic by 2021 [1]. In addition to media entertainment, a large number of industrial IoT devices (e.g., security cameras) also contribute to the massive video traffic, which causes huge network load problems and operation cost in internet service provider (ISP). Therefore, effective content distribution for data-intensive IoT applications such as streaming media is a key challenge to be solved in the future Internet for 5G.

5G is expected to support high-density deployment of latency-aware applications such as autonomous driving, virtual/augmented reality especially IoT devices, and services with meeting ultralow latency requirements. The challenging service requirements of 5G have motivated a need for feasible network model. Meanwhile, a large-scale deployments of Industry IoT applications also require a suitable network architecture to provide storage, mobility and security, etc.

However, the existing transmission control protocol (TCP)/IP faces many challenges and obstacles in supporting content-oriented IoT applications [2] mainly due to its weakness in supporting broadcast, mobility, and heterogeneous network. Besides, TCP/IP can only provide connection-based security mechanism which may bring intolerant latency for massive IoT devices. By contrast, named data networking (NDN) [3] is an important research direction among future Internet architecture proposals developed to address these problems in 5G and IoT scenarios [4]–[6]. The core mechanism of NDN is to send a named `Interest` unrelated to the location, thereby obtaining the corresponding `Data` via multipath and hop-by-hop forwarding strategy.

NDN has become a suitable candidate for serving IoT applications with feasibility and scalability.

1) NDN proposes a data-centric name-based routing schema, making it feasible to identify IoT devices and manage access control with hierarchical meaningful name.

2) Location-independent and pull-based receiver-driven communication paradigm that NDN offers natively enables mobility (e.g., mobile consumer user) and publish-subscribe model (e.g., message broadcasting in IoT scenarios).
3) And the features of in-network caching and multipath in NDN can reduce latency and enhance the efficiency of massive content delivery in IoT.
4) NDN identifies data packet by its name rather than location, enabling data-oriented encryption security at the network layer where IoT devices are generally exposed.

As for massive content delivery in network architecture, network coding is a widely used content-oriented approach to improve network capacity [7]. In NDN network, the default forwarding strategy, Best-Route Forwarding, may not allow network coding to use its full potential since Best-Route Forwarding mainly gets data from one path. Then we design a probability-based multipath forwarding called *Pinform*, to improve network coding performance and to optimize big data applications delivery.

The main idea of our design is to explore network coding in large volumes of data to encode and decode data chunks at producers and consumers, respectively, thus avoiding modifying packets in the middle NDN routing nodes. This method follows NDN fundamentals without excess modification. The combination of network coding (NC) and NDN Pinform forwarding strategy, referred to as *NDN-NC-Pinform* in this study, increases the diversity of the data blocks, and facilitates efficient utilization of cached data in NDN routers. More importantly, NDN-NC-Pinform effectively improves the performance, reliability, and QoS. This proposed solution could be used in kinds of applications such as streaming media system and content delivery network (CDN). What is more, this is a general solution that most of the cache strategies could be used in our solution with good performance.

To evaluate the performance of the proposed NDN-NC-Pinform scheme, we develop a network coding enhanced streaming media system in the ndnSIM simulator. To quantify the benefits of applying network coding, we compare a variety of performance metrics between NDN-NC-Pinform with the basic NDN-forwarding scheme and the NDN-NC-BestRoute scheme.[1] The contributions of this paper are as follows.

1) We design a probability-based multipath forwarding strategy, and propose the idea of applying network coding to optimize the forwarding mechanism via multipath by reducing the network bandwidth. Unlike other methods, our approach does not change the NDN fundamentals.
2) The NDN-NC-Pinform framework is designed based on probability-based multipath forwarding and network coding, while the NDN forwarding policy was modified to accommodate it in order to adapt to the way the data request after adding the network coding. Adding network coding to reduce the network load is theoretically analyzed in NDN-Forwarding and NDN-NC-Pinform.
3) We implement an NDN streaming system incorporating NDN-NC-Pinform to simulate IoT applications, and systematically quantify the benefits of applying network coding on improving cache hit ratio, jitter, network throughput, and network transmission efficiency which contributes to its application in 5G IoT environments.

[1] Applying network coding in NDN with Best-Route forwarding

The remainder of this paper is organized as follows. Section II discusses related work on network coding and NDN for 5G IoT. Section III describes our proposed solution combining network coding with multipath forwarding strategy. Section IV introduces the design of a network coding enabled streaming media system in NDN, while Section V presents the experimental results. Section VI concludes this paper and future work.

II. Related Work

With the increasing need of multimedia traffic and energy consumption in 5G networks, Device-to-Device and cloud-radio access network are general 5G-enabled solutions to address the challenges of network service (e.g., low delay, efficient energy, and bandwidth utilization) [8]–[10].

Except for the access techniques, recent years also witnessed the progress of network architecture for 5G and information-centric networking (ICN), an NDN-like architecture, is one of the promising candidates. Nishiyama *et al.* proposed a routing-based mobility architecture to support seamless mobility management for 5G cellular and noncellular networks, addressing the signaling overhead, and anchor-based challenges [11]. Ravindran *et al.* proposed 5G-ICN framework by leveraging the features in ICN, and showed the flexibility of mobility as a service enabled slicing in 5G [12].

NDN, or ICN is suitable for the deployment of IoT applications and various NDN-based IoT demos have been proposed, ranging from access control (security) [13], [14] to smart home scenarios [15], [16]. Zhang *et al.* realized the control of authentication and access for data security via an NDN-enabled mobile health device case [14]. Amadeo *et al.* implemented a NDN-based smart home demo [16] and demonstrated that IoT can benefit in energy consumption and storage efficiency from NDN when compared with IoT standard 6LoWPAN/RPL/UDP.

Network coding is a widely used technique for improving network throughput, scalability and data transmission efficiency [7], and providing information protection. In recent years, several studies have applied network coding theory in the content-centric networks, e.g., content distributions [17] and NDN networks [18], [19]. However, the papers [19] have a misunderstanding of the definition of NDN multipath forwarding, which does not mean to send the *same* Interest simultaneously to multiple upstream routers continuously. A recent research [18] proposes "CodingCache" concept to apply network coding for improving the cache hit ratio by about 60%. Although this concept brings significant benefits of improved hit ratio, it violates the basic NDN security principles due to requiring intermediate routers to perform network coding. Unlike these prior studies, this paper just encodes the data in the source and decodes it when the user gets it, while the router does not involve encoding.

Compared to our previous study [20], this paper has designed a probability-based multipath forwarding strategy for network coding to achieve better performance for IoT distribution. Besides, this paper is compared with NDN-NC-Pinform, NDN-

NC-BestRoute, and NDN-Forwarding in the experiments. And the experiments also show that our work is feasible to most cache approaches, which is an important feature to IoT application.

III. Network Coding and Multipath Forwarding in NDN

In this section, we first briefly describe the basic NDN forwarding strategy and design a probability-based multipath forwarding strategy in NDN. Subsequently, we present the idea of the proposed network coding forwarding strategies in NDN for effective content delivery. Finally, we present the theoretical analysis of network load in NDN with single-path topology and multipath topology with path diversity.

A. NDN Multipath Forwarding

Departing from the conventional single path forwarding strategy, named-data link state routing protocol (NLSR) [21], the routing protocol for NDN, provides multiple forwarding options for each name prefix. Adaptive forwarding [22], the basic forwarding strategy in NDN, has two approaches to explore multiple paths available in NDN networks. The first approach is to choose a single best path as long as the selected path can handle all the traffic. However, when a problem such as network congestion (packet loss) occurs to the path, the router will forward excess traffic to other alternative paths. The second method takes a proactive approach to first split traffic along multiple paths for forwarding, so the router would measure the data plane performance from multiple paths. In addition, a single failure only affects a small portion of the overall traffic.

The first approach is also called the best route forwarding strategy, which mainly uses one path and could not allow network coding to make full use of its potential. In this study, we use the main idea of the second approach to explore the ability and benefits of NDN multipath forwarding, which is just like that there are several expressways from New York to Chicago and each driver can select only *one* of them, but not select two or several of them simultaneously such as the papers [19]. Based on this idea, we design a probability-based multipath forwarding strategy called Pinform, for network coding to effectively utilize the idle paths in the network, to balance traffic load over multiple paths, and to distribute named data contents to a broader set of routers for in-networking caching.

B. Probability-Based Multipath Forwarding

Here we describe the design of Pinform, our proposed probability-based multipath forwarding for network coding in NDN. Pinform is inspired by the Q-routing algorithm [23] and INFORM [24], a dynamic request forwarding in ICN.

In the Pinform, each node runs independently and learns the delivery times of different files. The key notations have presented in Table I. Every node will update $Q(f,i)$ value according to Q-learning function once it receives a data belongs to file f through interface i, i.e., $Q_x(f,i) = (1-\alpha)Q_x(f,i) + \alpha(rtt_{x,y} + \sum_{m\in I_y(f)} P_y^{'} * Q_y(f,m))$. Then, we get the basic forwarding probability

TABLE I
Key Notation

$Q_x(f,i)$	Estimated delivery time for a package belongs to file *f* through interface *i* at node *x*.
$P_x(f,i)$	The basic forwarding probability for a package belongs to file *f* through interface *i* at node *x*.
$P_x^{'}(f,i)$	The modified forwarding probability.
$\overline{Q_x(f)}$	The expectation of estimated delivery time for a package belongs to file *f* at node *x* during exploration phase.
$\overline{Q_x(f)_{cur}}$	The expectation of estimated delivery time for a package belongs to file *f* at node *x* during exploration phase.
α	The Q-function learning parameter.
γ	The threshold probability to decide whether to forward data through the interface.

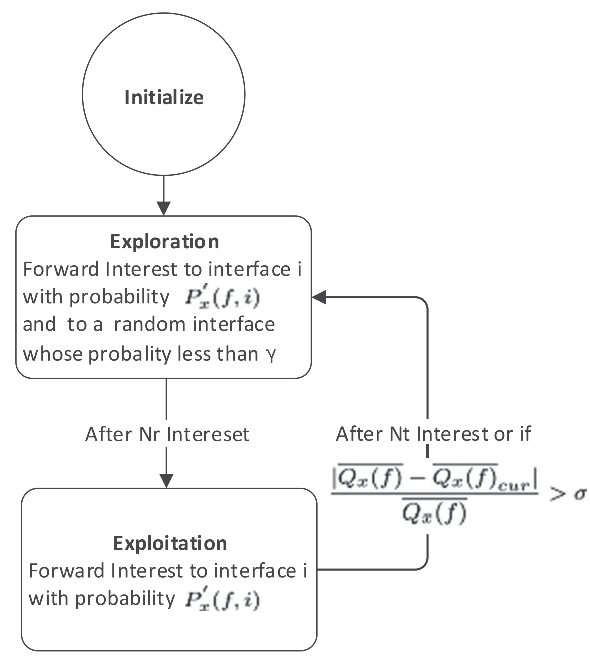


Fig. 1. Transition between phases.

$P_x(f,i)$ according to a set of $Q(f,i)$ value, i.e., $P_x(f,i) = (1)/(Q_x(f,i))/(\sum_{m\in I_x(f)}(1)/(Q_x(f,m)))$. If the interface i has a small $p_x(f,i)$ value, the interface i will have a large delivery time and it is not feasible to forward data. We set a threshold value γ to avoid the data forwarding to the bad interface by modifying the $p_x(f,i)$ value as follows,

$$P_x^{'}(f,i) = \begin{cases} 0, & P_x(f,i) \le \gamma \\ \frac{P_x(f,i)}{\sum_{m\in I_x(f) \text{ and } P_x(f,m)>\gamma} P_x(f,m)}, & P_x(f,i) > \gamma \end{cases}.$$

Once Pinform starts, it will initialize according to the routing costs. As shown in Fig. 1, Pinform has two main phases for forwarding Interest, exploration phase, and exploitation phase. In the exploration phase, the router not only forward Interests to an interface according to the $P_x^{'}(f,i)$, but also to a random interface whose basic probability $P_x(f,i)$ less than γ, which can explore new paths with limited overheads. The exploration phase lasts N_r Interests (N_r is a small number) and at the end

Algorithm 1: Operations on receiving an Interest from interface i at node x.

```
if requested Data is present in the cache then
    forward Data through interface i;
else
    if request is present in the PIT then
        add interface i to list of requesting interfaces;
    else
        create a new entry in the PIT;
        j=select interface m with probability P'_x(f,m);
        if Exploration phase then
            k=random select a interface m whose P_x(f,m)
            less than γ;
            forward Interest through interfaces j,k;
        else
            forward Interest through interface j;
        end if
    end if
end if
```

Algorithm 2: Operations on node x after receiving a Data through interface i from node y.

```
cache.store(data);
Q_x(f,i) = (1 − α)Q_x(f,i) + α(rtt_{x,y} +
    Σ_{m∈I_y(f)} P'_y(f,m) ∗ Q_y(f,m));
for all interface j ∈ I_x(f) do
    P_x(f,j) = (1/Q_x(f,j)) / (Σ_{m∈I_x(f)} 1/Q_x(f,m));
end for
for all interface j ∈ I_x(f) do
    if P_x(f,j) ≤ γ then
        P'_x(f,j) = 0;
    else
        P'_x(f,j) = P_x(f,j) / Σ_{m∈I_x(f) and P_x(f,m)>γ} P_x(f,m);
    end if
end for
for all interface j ∈ PIT.entry[data] do
    forward data through interface j;
end for
```

of the phase the expectation of estimated delivery time $\overline{Q_x(f)}$ is identified, i.e., $\overline{Q_x(f)} = \sum_{m\in I_x(f)} Q_x(f,m) * P_x^{'}(f,m)$. In the exploitation phase, the Interests will forward to only one interface according to the modified probability $p_x(f,i)$. The exploitation phase will last at most N_t Interests or until the network status changed a lot, i.e., $(|\overline{Q_x(f)} - \overline{Q_x(f)_{cur}}|)/(\overline{Q_x(f)}) > \sigma$. Algorithm 1 showed the operations (forwarding Interest) on receiving an Interest from interface *i* at node *x* and Algorithm 2 showed the operations (updating probability) on node *x* after receiving a Data through interface *i* from node *y*.

C. Exploring Network Coding in NDN Multipath Forwarding

The process of combing network coding in NDN could also be applied to any big files. Assume that a given file f has a number of segments, each of which consists of n data chunks (n is set as 4 in the experiments), i.e., $C_1, C_2, C_3, \ldots, C_n$. The data chunk is the unit of Interest request. With network coding at the original data source, the data owner could encode the data chunks into E_1, E_2, E_3, $\ldots$, E_n, with $E_i = v_1^i C_1 + v_2^i C_2 + v_3^i C_3^i + \cdots + v_n^i C_n^i$, and $\{v_1^i, \ldots, v_n^i\}$ is a randomly generated coefficient vector. Equation (1) summarizes the encoding formula for all data chunks of the segment. This encoding method can reduce the dependence on the blocks and increase the reliability of transmission

$$\begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} = \begin{pmatrix} V_1^1 & \cdots & V_n^1 \\ \vdots & \ddots & \vdots \\ V_1^n & \cdots & V_n^n \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} \tag{1}$$

The user requests data in the units of chunks. For example, the user sends an Interest request with the name of `/ndn/vod/netcod/seg_01/01`, indicating that the user does not have the coded chunks belonging to the first segment. The coefficient vectors of all data chunks corresponding to the first segment could be simply appended in the *coef* domain of Interest packets, e.g., `/ndn/vod/netcod/seg_01/01/coef`, and the producer or the data owner returns data packets with coefficient vectors in the name field. Once collecting all coded chucks, the user could apply the decoding formula in (2) to obtain all of the original n data chunks for constructing the entire segment

$$\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} = \begin{pmatrix} V_1^1 & \cdots & V_n^1 \\ \vdots & \ddots & \vdots \\ V_1^n & \cdots & V_n^n \end{pmatrix}^{-1} \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix} \tag{2}$$

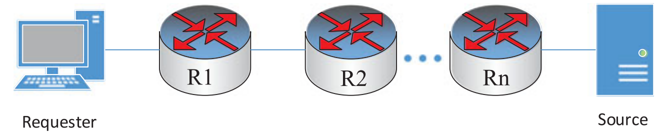


Fig. 2. Single-path topology with n NDN routers.

D. Theoretical Analysis of Network Load

Before using experimental results to demonstrate the benefits of NDN-NC-Pinform, we first perform theoretical analysis on network load for NDN-Forwarding and NDN-NC-Pinform with single-path topology and multipath topology, respectively. For simplicity, we use t_1 and t_2 to refer to these two topologies. Fig. 2 illustrates a single-path topology with n NDN routers, one data requester, and one data source, whereas Fig. 3 shows a multipath topology with 4 NDN routers, which form two disjoint paths `R1-R2-R4` and `R1-R3-R4` to dispatch traffic from the data source to the requester. We assume that both of the paths have the same status and same probability to forwarding data.

Assume the data source has a file with m data chunks, each of which has a size of c bytes. The data requester sends an `Interest` request, which is forwarded through one of the paths to the data source. The data source returns encoded data chunks via network coding technique along the reverse paths.

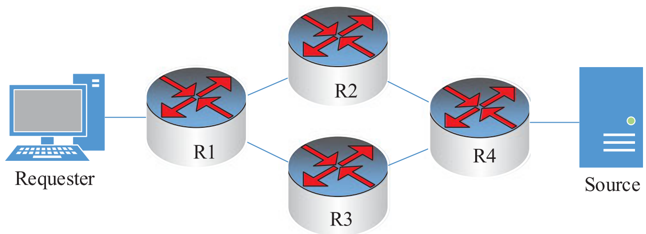


Fig. 3. Multipath topology.

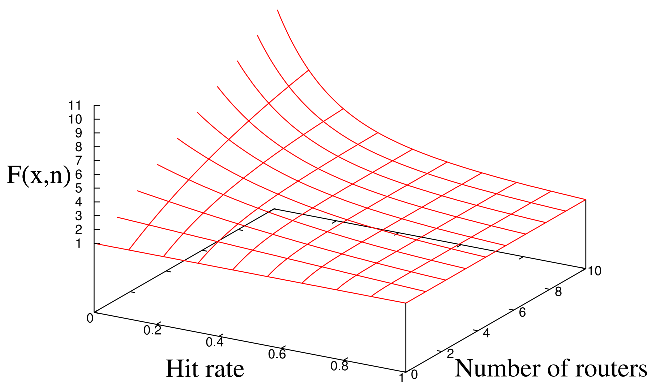


Fig. 4. Correlation between cache hit ratio and network coding with single-path topology.

Thus, the network load of a data request is defined as the total traffic carried by all the links to transfer all m data chunks of the file from the data source to the data requester. In this analysis, we ignore the size of `Interest` packets due to their small size.

Assume the cache hit ratio is p in NDN-Forwarding, and the ratio is p' in NDN-NC-Pinform. As each coded block is formed by all m blocks, thus $p \leq p' \leq mp$. That is $p' = \lambda p$, where $1 \leq \lambda \leq m$.

Now we analyze the network load for NDN-Forwarding and NDN-NC-Pinform under the single-path topology.

Single-Line Topology: For the above scenario, the network load for NDN-Forwarding, $f(p, t_1)$ is

$$\begin{aligned} f(p,t_1) &= \{p + 2(1-p)p + \cdots + (n+1)(1-p)^n\}mc \\ &= \left\{\frac{1-(1-p)^n(np+1)}{p} + (n+1)(1-p)^n\right\}mc. \end{aligned} \tag{3}$$

Similarly, the network load for NDN-NC-Pinform, $f(p', t_1)$ is

$$\begin{aligned} f(p',t_1) &= \{p + 2(1-p')p' + \cdots + (n+1)(1-p')\}mc \\ &= \left\{\frac{1-(1-p')^n(np'+1)}{p'} + (n+1)(1-p')^n\right\}mc \\ &= \left\{\frac{1-(1-\lambda p)^n(n\lambda p+1)}{\lambda p} + (n+1)1-\lambda p)^n\right\}mc. \end{aligned} \tag{4}$$

Assume $F(x,n) = (1-(1-x)^n(nx+1))/(x+(n+1)(1-x)^n)$, where x could be either p for NDN-Forwarding or λp, i.e., p', for NDN-NC-Pinform. Fig. 4 illustrates a decreasing trend of network load as a result of improving cache hit ratio. More importantly, Fig. 4 also shows that NDN-NC-Pinform has less network load than NDN-Forwarding, indicating the benefits of network coding on reducing network load under the single-path topology.

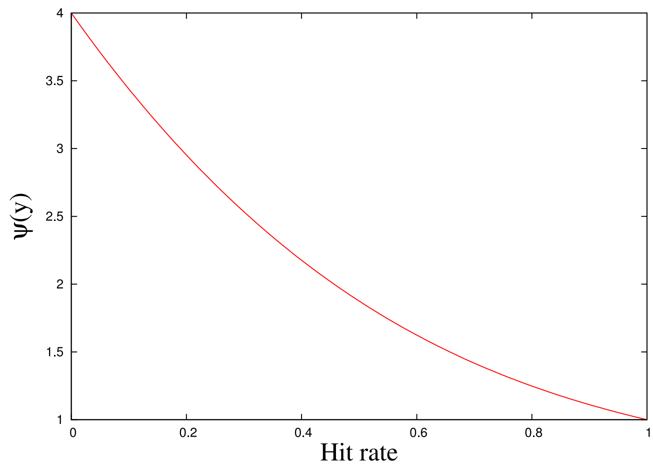


Fig. 5. Correlation between hit ratio and network load with network coding in multipath topology.

Multipath Topology: For the same scenario of Interest request and data chunks, the network load for NDN-Forwarding, $f(p, t_2)$ is

$$\begin{aligned} f(p,t_2) &= \{p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3\}mc \\ &= \{4 - 6p + 4p^2 - p^3\}mc. \end{aligned} \tag{5}$$

The network load for NDN-NC-Pinform, $f(p', t_2)$ is

$$\begin{aligned} f(p',t_2) &= \{p' + 2(1-p')p' + 3(1-p')^2p' \\ &\quad + 4(1-p')^3\}mc \\ &= \{4 - 6p' + 4p'^2 - p'^3\}mc \\ &= \{4 - 6\lambda p + 4(\lambda p)^2 - (\lambda p)^3\}mc. \end{aligned} \tag{6}$$

Assume $\Psi(y) = 4 - 6y + 4y^2 - y^3$, where y could be either p for NDN-Forwarding or λp, i.e., p', for NDN-NC-Pinform. Fig. 5 shows that the network load $\Psi(y)$ decreases with improving cache hit ratio. After introducing network coding to enable the diversity of transferring data blocks in parallel, the cache hit ratio increases, which effectively reduces network load in multipath topology.

In summary, our theoretical analysis on network load reveals that adding network coding reduces network load for both single-path topology and multipath topology. In the next two sections, we will design a network coding enabled NDN streaming media system and evaluate the performance benefits of network coding for effective NDN content delivery in experiments.

IV. Network Coding Enhanced NDN Streaming System

To explore the benefits of network coding on reducing the network bandwidth and maximizing the use of in-network caching, we design and implement the proposed NDN-NC-Pinform scheme. We use NDN-NC-SM to refer to the entire system model prototype in this paper. In this section, we first outline the overall system architecture of NDN-NC-SM, and subsequently present the design of system interactions, naming scheme, network coding processes, and forwarding module.

A. System Architecture

NDN-NC-SM is developed based on an existing streaming system in IP networks. NDN-NC-SM retains the advantages of the existing system including the separation of control flows and data flows and peer-to-peer data transmission. The main change is to implement NDN multipath forwarding mechanism and network coding into the system architecture, and adjust the characteristics of the system architecture based on NDN multipath forwarding mechanism and network coding, reducing the bandwidth and maximizing the use of network cache.

1) *Preprocessing Node*: The preprocessing node is a publishing and maintaining system for preprocessing streaming media files. The preprocessing node is built with three major functions:
   a) processing the video data based on the NDN protocols and the relevant video file types, encoding and transmitting data to the source node;
   b) managing the program online and offline;
   c) managing source nodes within the NDN networks.
2) *Directory Index*: The directory index maintains the directory information of video files in the storage system. The directory index provides users a list of the current browsable videos and a user interface for user operations, such as adding and deleting interested videos.
3) *Source Node*: Source node stores the video data and responds the requests from the users.
4) *User*: The user sends an Interest to obtain the videos and sends the video data to a video player after decoding.

B. System Interactions

To use NDN-NC-SM, the users mostly interact with directory indexes and the source node. Above all, a user will first send an `Interest` to the directory indexes to request the program list after the login process, and subsequently send another `Interest` to the source node to request data chunks after selecting a particular video for watching. The interactive process steps between the directory indexes and the source node are as follows:

1) The preprocessing node sends an `Interest` to request program changes, such as online, update, offline, to the directory indexes, notifying the directory indexes to change the relevant programs. An example of such request is as "`/hippo.pkusz.edu.cn/vod/dirsvr/stream_id/online`."
2) Upon receiving the Interest request, the directory indexes validate whether the relevant fields meet the requirements, and return a different `Interest` to the preprocessing node for requesting the changed data. An example of such request is "`/hippo.pkusz.edu.cn/vod/presvr/stream_id/consist_id/desc`."
3) Preprocessing node will return the relevant program information to directory indexes after receiving the request of the changed data.

When a new video file is released on the preprocessing node, the meta information of the video will be extracted and integrated into the directory on the directory indexes. After the successful release of directory information, preprocessing node will send video metadata, key frame indexes, and data blocks to the source node in order. Their interactive process is similar to the aforementioned steps between the directory indexes and the source server.

C. System Design

1) Network Coding Module:
The NDN-NC-SM system applies random network coding for maximizing link utilization and NDN network throughput. By following the principles outlined in (1) and (2), the `source node` applies coding coefficient vectors on the original data chunks to produce coded chunks, while the `user` uses the inverse matrix of the coding coefficient vectors, forwarded from the source, on the coded chunks to obtain the original data chunks. The encoding and decoding operations are implemented on the Galois field with $GF(2^n)$ for handing floating point calculations and ensuring the linear independence.

2) Forwarding Strategy:
Network coding enhanced NDN improves the diversity of the blocks as the coded block contains the plurality of original chunks' information. Several Interest of a segment could be responded by a same coded block in the case of independent coding coefficient vectors.

We change NDN forwarding policy based on the existing one to take advantage of the benefits brought by network coding to improve network throughput and reduce overall network traffic. NDN network we built is consistent with the basic architecture. We just change the query processing operations of CS, pending interest table (PIT), forwarding information base (FIB) in forwarding process. Specific changes of forwarding policy are mainly divided into two parts.

1) *Searching strategy after the router receives Interest:* The router will first determine whether to request the coded data blocks or not after receiving an Interest packet. It will use the basic NDN forwarding strategy if the Interest requests the noncoding data blocks. Otherwise, the content store will be first searched, returning the data if matched (same name and the coefficient independent) while the same name is the only requirement in basic NDN. The router thereby will search in PIT and FIB then adopt the same strategies in NDN-Forwarding. The process is shown in Fig. 6. For example, an Interest with name "`/ndn/vod/nc/seg_1/[coef_x]`," meaning requesting to the coded chucks due to the flag "nc," reaches a router. The router will first check its Content Store, in which if a data named "`/ndn/vod/nc/seg_1/[coef_2]`" has the same name"`/ndn/vod/nc/seg_1/`" with the Interest and the coefficient vector "coef_2" is linear independent to the Interest's coefficient vectors "coef_x," then the data

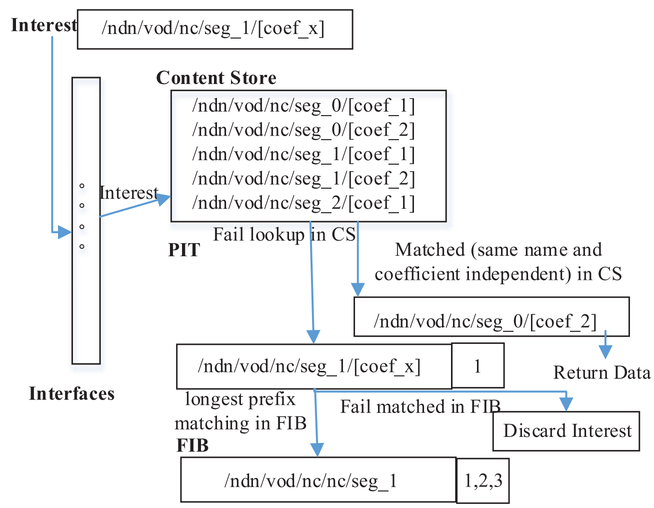


Fig. 6. Forwarding Interest.

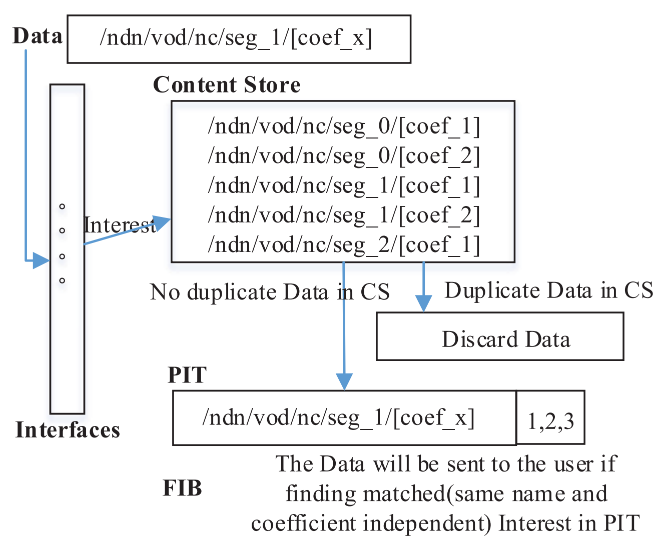


Fig. 7. Forwarding data.

is matched to the Interest and will be returned to the user. Otherwise, it fails in lookup in CS and will check the PIT and FIB as it deals in NDN-Forwarding.

2) *Searching strategy after the router receives data:* The router will first determine whether to request the coded chucks or not after receiving a packet. It will use the basic NDN forwarding strategy if the data packet is uncoded. Otherwise, the packet will be discarded if finding a duplicate data (with the same name and the coefficients dependent) in the Content Store. If finding the matches (with the same name and the coefficients independent) in the PIT, the router will send the data to all faces corresponding to the face list in PIT except the packet's entrance face. The process is shown in Fig. 7. For example, a data with name "`/ndn/vod/nc/seg_1/[coef_x]`," meaning requesting to the coded chucks due to the flag "nc," reaches a router. Then the router will first check its Content Store, in which if a data named "`/ndn/vod/nc/seg_1/[coef_2]`" has the same name"`/ndn/vod/nc/seg_1/`" with the reached data and the coefficient vector *"coef_2"* is linear independent to the reach one's "coef_x," the reached data is called to a duplicate data and will be discarded. Otherwise, the router will check its PIT and if an Interest named "`/ndn/vod/nc/seg_1/[coef_y]`," meaning the Interest and the reached data have the same name "`/ndn/vod/nc/seg_1/`" and their coefficient vectors, "coef_y" and "coef_x," are linear independent; Then the Interest is matched to the reached data, which thereby will be stored in the router and be sent to the faces corresponding to the face list.

3) Overhead Analysis of Space and Time: Applying network coding in NDN multipath forwarding brings the space and time overheads. The space cost is mostly caused by the transfer of coefficient vectors from source code to the user. In our prototype system, we set the coefficient in the range of [0, 255] for achieving effective codec in the NDN networks. Thus, this codec operation is implemented in a finite field with $GF(2^8)$ with each factor occupying a single byte. In the streaming media application, we set the number of data chunks between 4 and 10, thus creating a small number of coefficient vectors. In other words, the space overhead caused by coefficient vectors is very small.

The time overhead is mostly contributed by the encoding process at the source node and the decoding process at the user. As the source code could pre-encode data chunks for saving end-to-end latency, we need to consider the decoding time at the user for calculating the original data blocks. Our NDN-NC-SM applies singular value decomposition (SVD) algorithm to decode data chunks and determine linearly independence of coefficient vectors, thus the time complexity is $O(n^3)$ [17], where n is the number of data chunks in a segment. Given the number of data chunks set between 4 and 10, the time spent on decoding is in an acceptable range.

V. Performance Evaluation

A. Network Topology For Experiments

To systematically evaluate the performance of our proposed NDN-NC-Pinform scheme, we design and implement a prototype system and evaluate the system with ndnSIM [25] simulation environment. The NDN network topology is generated via a widely used "Brite topology generator," which uses Waxman model to produce router-only network topology.

Table II outlines the parameters used in our experiments. For our experiments, we generate a simulated NDN network with 100 nodes and 200 edges. The average degree (i.e., the number of connected neighbors) of node is 4, and the degrees among all the nodes range from 2 to 11. The link bandwidths of the network follow an even distribution of 100–10240 kbps, while the latency follows an even distribution of 1–40 ms. In addition, we simulate a simultaneous downloading behavior for a variety of users in a time windows of 60 s.

B. Reliability and QOS

We explore reliability and QoS by checking jitter, throughput, and congestion. Jitter is an important factor of QoS in

TABLE II
PARAMETER VALUES USED IN THE SIMULATIONS

Parameter	Value
Link bandwidth	100–10 240 kbps
Link delay	1–40 ms
Queue length	100
PIT replacement policy	LRU
PIT size	1000
CS replacement policy	LRU
CS size	4000
Interest frequency	200
Data (payload size)	1024 B

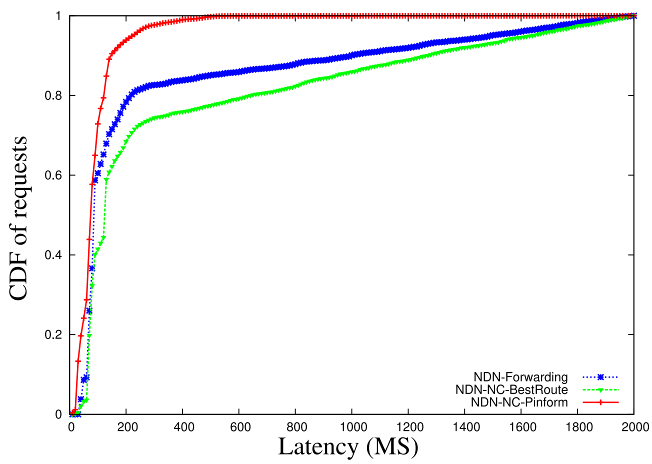


Fig. 8. Jitter of NDN-Forwarding, NDN-NC-BestRoute and NDN-NC-Pinform.

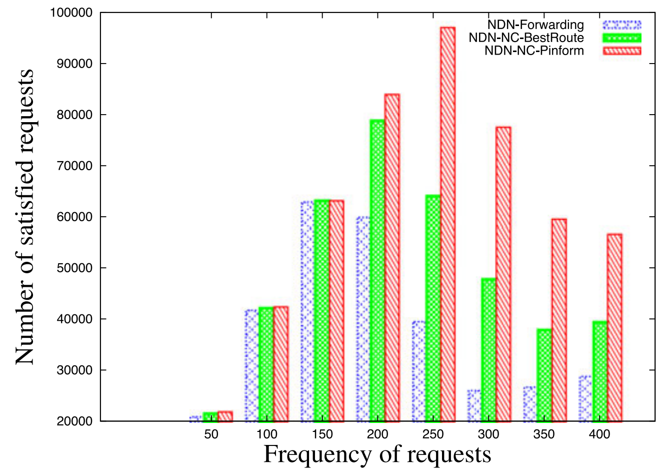


Fig. 9. Throughput of NDN-Forwarding, NDN-NC-BestRoute and NDN-NC-Pinform.

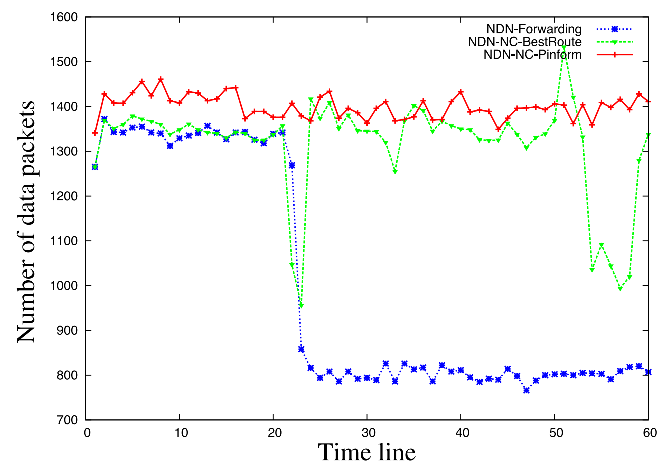


Fig. 10. Congestion.

streaming media system, and we evaluate the jitter by the cumulative distribution of requests latency. The latency is defined as the time from the node requesting a new segment to the node being able to decode the received packets. The experimental results of NDN-Forwarding and NDN-NC-BestRoute serve as a baseline performance benchmark for evaluating our proposed NDN-NC-Pinform.

The jitter in NDN-Forwarding, NDN-NC-BestRoute, and NDN-NC-Pinform is illustrated in Fig. 8. And 95% of the requests in NDN-NC-Pinform have the latency less than 200 ms, whereas in NDN-Forwarding the ratio decreases to 80% and in NDN-NC-BestRoute the ratio decreases to 70%. As a result, the requests are mainly distributed in low latency both in NDN-Forwarding and NDN-NC-Pinform, but a large amount of them in two baselines still distribute in high latency, resulting a worse jitter compared to our NDN-NC-Pinform.

Next we evaluate the network throughput of NDN-NC-Pinform via measuring the number of successfully transmitted data packets in various request frequencies in NDN. As is shown in Fig. 9, the throughputs are mainly increased first, then decreased, and at the end keep in a stable level as the request frequency increasing. When the network traffic is light, the throughputs in two baselines are very closed to our NDN-NC-Pinform as almost all the packets are transmitted successfully with few packets dropped. However, when the network traffic is heavy, there is a significant difference in the throughputs of three solutions as NDN-NC-Pinform suffers little influence by congestion and performs much better, increasing the overall network throughput by approximately 50% and 100% compared with NDN-NC-BestRoute and NDN-Forwarding, respectively.

Fig. 10 shows the number of successfully transmitted packets per second in above three solutions. We can see that the number of the transmitted packets are very close in the first 22 s, but the curve in NDN-Forwarding decreases rapidly and keeps in a low level while that in NDN-NC-Pinform does not have a big fluctuation. It is because of the congestion causing a large amount of packets dropped, resulting a low level packets transmission in NDN-Forwarding. Although the curve of NDN-NC-BestRoute keeps in a high level in most time, it always has a large shock with little time higher than that of NDN-NC-Pinform. As a result, NDN-NC-Pinform can perform much better in a heavy network traffic scenario where massive IoT devices suffer by avoiding congestion, reflecting a much better reliability and QoS.

C. Performance

1) Cache Hit Ratio: We first study the benefits of NDN-NC-Pinform and the increased data chunks on the cache hit ratio. Fig. 11 illustrates the relationships between the number

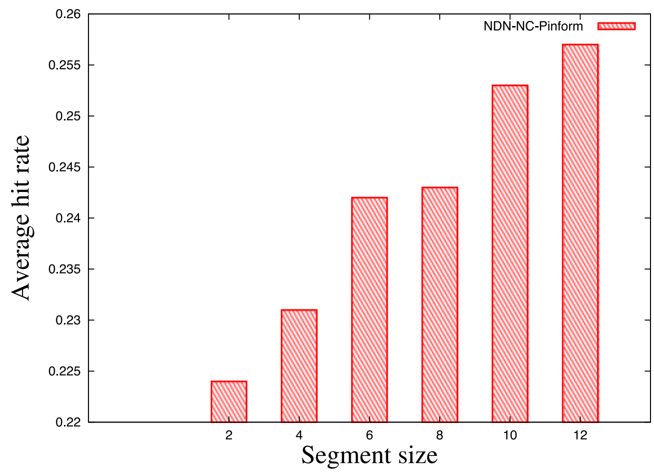


Fig. 11. Correlation between the number of chunks in a segment and the cache hit rate.

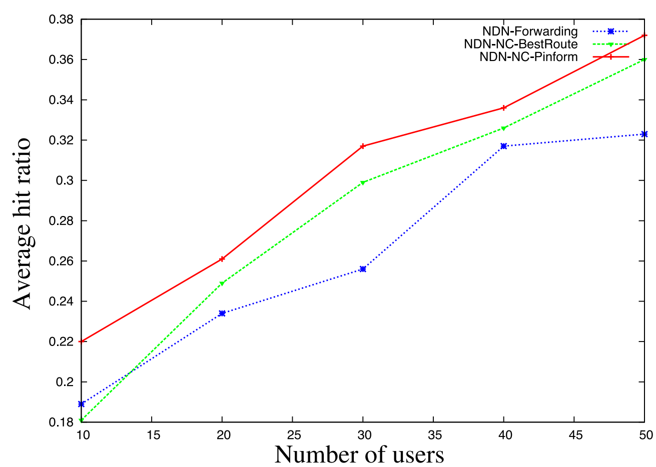


Fig. 12. Relation between user numbers and the cache hit ratio.

of encoded data blocks and cache hit ratio. Apparently, as the number of encoded chunks in a segment increase, the cache hit ratio increases. This trend starts to stabilize once the number of blocks reaches a certain threshold, i.e., 10 blocks observed in our simulations. The hit ratio increase is largely contributed by a coded block containing more chunks' information, improving the probability of Interests being met.

Given the prior observations, we set 4 as the number of encoded chunks for each segment in our simulations. To quantify the benefits of NDN-NC-Pinform, we compare the cache hit ratio for NDN-Forwarding, NDN-NC-BestRoute, and NDN-NC-Pinform under a variety of user numbers. As shown in Fig. 12, cache hit ratio naturally increases as the user number increasing. By introducing network coding strategies, NDN-NC-BestRoute scheme achieves 10% higher cache ratio than the basic NDN-Forwarding scheme. More importantly, by introducing a feasible forwarding strategy, NDN-NC-Pinform achieves 10% higher cache ratio than NDN-NC-BestRoute.

2) Network Transmission Efficiency: Network transmission efficiency is defined as the ratio of the actual transmission of the useful data transmission and the actual network traffic. The

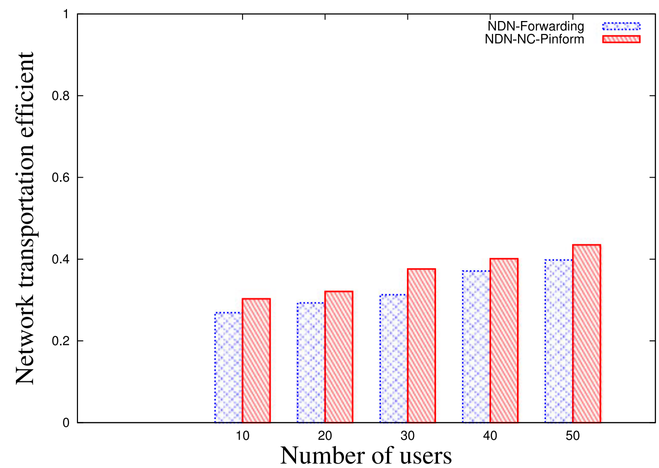


Fig. 13. Network transmission efficiency.

amount of the actual network traffic is equal to the total flow forwarded by all participating network nodes. Fig. 13 illustrates the network transmission efficiency of both NDN-Forwarding and NDN-NC-Pinform. Clearly, network transmission efficiency of NDN-NC-Pinform is higher than NDN-Forwarding especially in the large amount users scenario. It is improved from 10% to 20%, which is consistent with the analysis in Section III. Using basic NDN-Forwarding, each Interest may lead to ineffective responses, which is effectively resolved with a very low probability of data being discarded, leading a higher proportion of the effective amount of data transmission in NDN-NC-Pinform.

D. Generalizability

In this section, we verify that most of cache strategies could be used in our solution and give full play to the characteristics of the cache strategy and network coding. We select a simple cache strategy, which is called as lifetime-based greedy caching approach (LBGC) [26], to replace the least recently use (LRU) default cache strategy in NDN. In LBGC, the new arrival contents only replace the expired cache items and simulation results indicate that LBGC has better performance compared with other caching approaches. To simulate different scenarios, the users in this experiment start to request data with different time and the results will be different compared with the former experiments.

In Fig. 14, 92% of the requests with LBGC in NDN-NC-Pinform have the latency less than 200 ms, while with LRU cache approach, the ratio decreases to 77%. In Fig. 15, in the first 30 s, the number of satisfied packets gradually increases as the users starting to request one by one. At the end of 27 s, all the users have joined to request the data and the number of satisfied packets keep in a high level in the following time. We can see that NDN-NC-Pinform with LBGC cache approach is better at the most time from the fluctuating curves. Therefore, our NDN-NC-Pinform is a general solution to apply network coding in NDN as it is well fit to most of the cache approaches.

In summary, our experiments show that the performance, reliability, and QoS of NDN-Forwarding are significantly improved by NDN-NC-Pinform in various aspects such as jitter, throughput, congestion, hit ratio, and transmission efficiency,

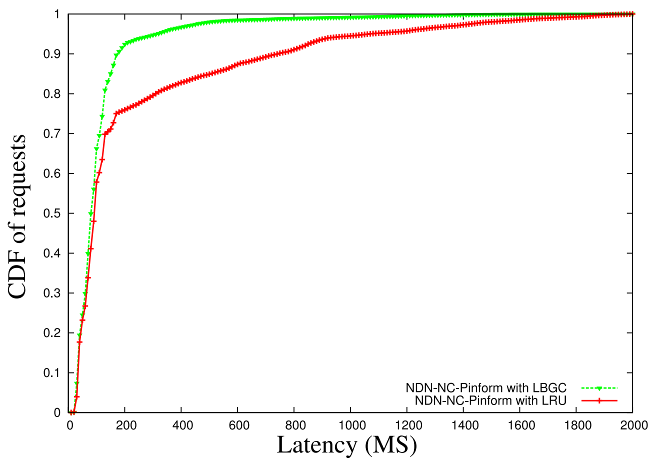


Fig. 14. Jitter of NDN-NC-Pinform with LBGC and LRU.

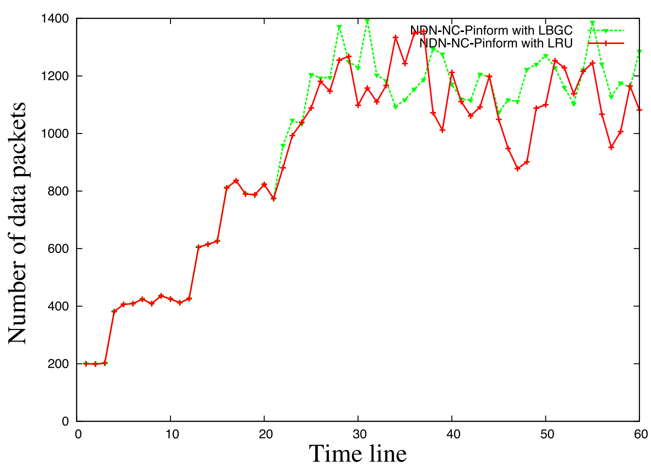


Fig. 15. Throughput of NDN-NC-Pinform with LBGC and LRU.

which indicates that NDN-NC-Pinform is an effective NDN content delivery strategy and can be well matched in streaming system. As for massive deployment of IoT in 5G, our NDN-based content distribution model can serve as a suitable solution for its higher efficiency in large scale content delivery and reliable QoS guarantee.

VI. Conclusion and Future Work

This paper introduced a network coding enhanced content distribution model (NDN-NC-Pinform) for 5G-NDN, which combined with probability multipath forwarding to fully utilize idle path network and improve the performance of content delivery in IoT scenario. This method strictly followed NDN design fundamentals. To demonstrate the feasibility and evaluate the performance of this model, we developed a network coding enhanced NDN streaming system in the ndnSIM simulator. The experimental results revealed that NDN-NC-Pinform outperforms the basic NDN forwarding by improving performance (i.e., network throughput and transmission efficiency, etc.) and reflecting a much better reliability and QoS. Content-based IoT applications can benefit from such high efficiency in massive content delivery of our NDN-based model. With the maturing of the prototype system, the idea of this article can be realized in a real demo of Industrial IoT in the future, e.g., intelligent security and defense system with cameras.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2016-2021," Cisco, San Jose, CA, USA, Tech. Rep. No.1465272001663118, Sept. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[2] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," NDN Tech. Rep. NDN-0038, Named Data Networking Project, Feb. 2016.

[3] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Review*, vol. 44, no. 3, pp. 66–73, 2014.

[4] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. Leung, "Network slicing based 5G and future mobile networks: mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[5] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. C. M. Leung, "Fog radio access networks: Mobility management, interference mitigation and resource optimization," *IEEE Wireless Commun.*, 2017, to be published.

[6] K. Lei, C. Hou, L. Li, and K. Xu, "A rcp-based congestion control protocol in named data networking," in *Proc. IEEE Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, 2015, pp. 538–541.

[7] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Mar. 2012, pp. 280–285.

[8] Moiz, M. I. Ashraf, K. S. B. Liaqat, S. Mumtaz, and M. Katz, "Implementation of D2D enabled mobile cloud-based content distribution architecture in 5G networks," in *Proc. 22nd Eur. Wireless Conf.*, Oulu, Finland, 2016, pp. 1–6.

[9] D. Zhang, S. Mumtaz, Z. Zhou, and T. Sato, "Integrating energy efficiency mechanism with components selection for massive MIMO based CRAN," in *Proc. IEEE Int. Conf. Commun. Workshops*, Kuala Lumpur, Malaysia, 2016, pp. 74–79.

[10] K. M. S. Huq, S. Mumtaz, J. Rodriguez, P. Marques, B. Okyere, and V. Frascolla, "Enhanced CRAN Using D2D Network," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 100–107, Mar. 2017.

[11] Y. Nishiyama, M. Ishino, Y. Koizumi, T. Hasegawa, K. Sugiyama, and A. Tagami, "A Proposal on routing-based mobility architecture for ICN-based cellular networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2016, pp. 467–472.

[12] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, "5G-ICN: Delivering ICN services over 5G using network slicing," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 101–107, May 2017.

[13] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L Zhang, "Securing building management systems using named data networking," *IEEE Netw.*, vol. 28, no. 3, pp. 50–56, May/Jun. 2014.

[14] H. Zhang, Z. Wang, and L. Zhang, "Sharing mHealth data via named data networking," in *Proc. 3rd ACM Conf. Inf.-Centric Netw.*, pp. 142–147, 2016.

[15] E. Baccelli, C. Mehlis, and O. Hahm, "Information centric networking in the IoT: Experiments with NDN in the Wild," *ACM Conf. Inf.-Centric Netw.*, 2014, pp. 77–86.

[16] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Information centric networking in IoT scenarios: The case of a smart home," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 648–653.

[17] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE Comput. Commun. Soc.*, Mar. 2005, pp. 2235–2245.

[18] Q. Wu, Z. Li, and G. Xie, "Codingcache: Multipath-aware CCN cache with network coding," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw.*, 2013, pp. 41–42.

[19] S. Miyake and H. Asaeda, "Network coding and its application to content centric networking." 2013. [Online]. Available: http://www.me.inf.kyushu-u.ac.jp/witmse2013/proceeding_files/W7.pd

[20] K. Lei, F. Zhu, C. Peng, and K. Xu, "Network coding for effective NDN content delivery: Models, experiments, and applications," in *Proc. 44th Int. Conf. Parallel Process.*, Sep. 2015, pp. 101–110.

[21] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "Nlsr: Named-data link state routing protocol," in *Proc. 3rd ACM SIGCOMM Workshop Inf.-Centric Netw.*, 2013, pp. 15–20.

[22] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 42, pp. 62–67, Jun. 2012.

[23] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 671–671.

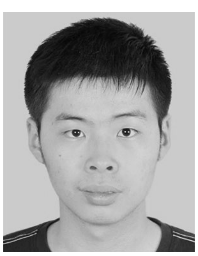
[24] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "Inform: A dynamic interest forwarding mechanism for information centric networking," in *Proc. 3rd ACM SIGCOMM Workshop Inf.-Centric Netw.*, 2013, pp. 9–14.

[25] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Tech. Rep. NDN-0005, Named Data Networking Project, Oct. 2012.

[26] H. Qian, W. Muqing, W. Dongyang, and G. Song, "Lifetime-based greedy caching approach for content-centric networking," in *Proc. 21st IEEE Int. Conf. Telecommun.*, 2014, pp. 426–430.

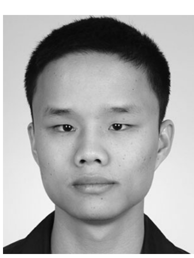
Kai Lei (M'14) received the B.Sc. degree from Peking University, Beijing, China, in 1998, the M.Sc. degree from Columbia University, New York, NY, USA, in 1999, and the Ph.D. degree from Peking University, Beijing, China, in 2015, all in computer science.

He worked for companies including the IBM T.J. Waston Research Center, Citigroup, Oracle, Google, from 1999 to 2004. He is currently an Associate Professor at the School of Electronics and Computer Engineering, Peking University, Shenzhen, and participates in the CENI project supported by National Development and Reform Commission since 2016. His research interests include named data networking and big data technologies.



Shangru Zhong (S'17) received the B.Eng. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2015. He is currently working toward the M.S. degree in computer science at Peking University, Shenzhen, China.

His research interests include future networks, recommender system, and social networks.



Fangxing Zhu received the M.S. degree in computer science from Peking University, Shenzhen, China, in 2016.

His research interests include named data networking and content delivery network.

Kuai Xu (M'02–SM'15) received the B.S. and M.S. degrees in computer science from Peking University, Beijing, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 2006.

He is currently an Associate Professor at Arizona State University, Tempe, AZ, USA. His research interests include network security, cloud computing, and social networks.

Dr. Xu is a member of the Association for Computing Machinery (ACM).

Haijun Zhang (M'13–SM'17) received the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2013.

He is currently a Full Professor at the University of Science and Technology Beijing, Beijing, China. He was a Postdoctoral Research Fellow at the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC, Canada. From 2011 to 2012, he visited the Centre for Telecommunications Research, King's College London, London, U.K., as a Visiting Research Associate. He has authored and coauthored more than 100 papers and authored two books.

Dr. Zhang is an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE 5G TECH FOCUS, and the JOURNAL OF NETWORK AND COMPUTER APPLICATIONS, and serves/served as a Leading Guest Editor for the IEEE COMMUNICATIONS MAGAZINE, and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING. He serves/served as the General Co-Chair of 5GWN 2017 and GameNets 2016, Track Chair of ScalCom 2015, Symposium Chair of GameNets 2014, TPC Co-Chair of INFOCOM 2018 Workshop on Integrating Edge Computing, Caching, and Offloading in Next Generation Networks, General Co-Chair of ICC 2018 (ICC 2017, Globecom 2017) Workshop on 5G Ultra Dense Networks, and General Co-Chair of Globecom 2017 Workshop on LTE-U. He has served as a TPC member in a numerous international conferences. He was the recipient of the IEEE ComSoc Young Author Best Paper Award in 2017.