A CRC-32 Name Prefix Encoding in Named Data Networking

Jenq-Muh Hsu\textsuperscript{1}, Jui-Yang Chang\textsuperscript{2}

Department of Computer Science and Information Engineering, National Chiayi University
No. 300, Syuefu Rd., Chiayi City 60004, Taiwan (R.O.C.)
\textsuperscript{1}hsujm@mail.nannyu.edu.tw
\textsuperscript{2}s1010441@mail.nannyu.edu.tw

Abstract—Named Data Networking (NDN) is a content-centric network. It is not similar to traditional IP-based network. It uses the content name, not IP address, to route and transmit information in NDN. Therefore, NDN focuses on WHAT (data content) instead of WHERE (data location). A content provider publishes content with a unique named prefix, called as named data, into a content-space (CS) in a NDN node. If a user wants to access the named content, the name prefix is used to query and find the corresponding content in NDN. This paper adopts the CRC-32 (Cyclic Redundancy Check) scheme to encode the named prefix in a fixed length encoded form in order to efficiently enhance the query mechanism of named content based on the named prefix. Through a fixed-length query mechanism to lookup the named data, it can improve the lookup efficiency in NDN indeed.

Keywords—Named Data Networking, Named Data, Name Prefix, CRC-32, Content Lookup.

1. INTRODUCTION

With the rapid development of the Internet, various types of application services are developed and vivid information are generated. It believes that the growth of information traffic will become much huge in the Internet. A traditional IP-based network could not handle much huge information them for storing, querying and searching, and deliver, especially the meaningful content, in the future Internet.

In traditional IP network, an IP address scheme is used to address and route network information. That is, it is a host-to-host based data transmission. A user wants to access the data, such as online video and newest news, in traditional IP network. At first, the user should understand the IP address of the host having the information through the Internet searching service. The host-to-host communication link between the data requester and the data provider is established and then the user can access the information from the originated source. At the same time, the other users want to access the same information at next movement. They will do the same operations to access the same information many times. That is, if there are a lot of users want to access the same information in a period time. It will lead to waste huge volume of network traffic to transmit the same information from the content provider. This case will also decrease and affect the network performance. Thus, several innovative transmission frameworks are discussed and proposed for meeting the new features in the future Internet.

Named Data Networking[1], NDN, is a content-centric network[2][3]. It adopts the content name to be the network address. The NDN nodes, also called NDN routing nodes, use the prefix name to handle the issues, including content addressing, content routing, and content transmission in NDN. Besides, NDN nodes have the caching ability to cache the content packets which are routed through them. If the other information requests are reaching to the NDN nodes which have cached the content in their caches, the nodes will immediately reply the requesting contents to the requesting users without accessing the contents from the original content providers. Thus, the NDN provides a content-to-content-based routing mechanism to efficiently reduce the network traffic of the same requested content while they are accessed too huge.

A unique name prefix is used to identify the content in NDN. Different contents have different name prefixes. The name prefix is also used for addressing, searching, routing, and accessing the contents. That is, the routing mechanism of NDN is a content-to-content routing.

The name prefix can clearly identify the content in NDN, it also allows using a meaning
International Journal of Advanced Information Technologies (IJAIT), Vol. 8, No.2

name to assign the content. The length of name prefix is variable and not limited. Although content can be assigned with a meaning prefix name, the length is not fixed. It will not search, query, route, and access the content with the variable length of name prefix efficiently. The same problem is also occurred in the notation of URL. Thus, some providers and research devote to develop the URL-shorten techniques[4][5], such as tiny URL, for efficiently and fast searching and accessing the web pages. For the same reason, an efficient name searching and content accessing mechanism is necessary in NDN.

In this paper, a CRC-32-based encoding scheme is adopted to shorten the variable length of name prefix into a fixed length for improving content searching in NDN. The main reason is to fix the name length of the content that can efficiently reduce the response time of content search without affecting the original data structure for storing information content in NDN. Using the shortened and encoded name prefix in NDN, the prefix name in NDN packets will be shortened more than the originated one. Thus, it can efficiently speed up the searching time of name prefix and save the storing space of name prefix by shortening the length of name prefix.

This paper is organized as follows. The related issues of NDN will be briefly introduced in Section 2. Section 3 describes our proposed and integrated approach to shorten the variable length of prefix name into the fixed length based on CRC-32 encoding scheme. The experimental result of our proposed approach is described in Section 4. Finally, conclusions are made in Section 5.

2. NAMED DATA NETWORKING

The mainstream of Named Data Networking (NDN) is the NDN project[1] derived from Content Centric Networking (CCNx)[6] developed by PARC Inc. Content in NDN has a unique name prefix. The structure of name prefix is similar to the form of URL (Uniform Resource Locator). It combines a series of meaningful identifiers concatenated by slash symbol. For example, the name prefix of content may be presented as the following notation, “/ndn/media/video1”.

The routing mechanism of traditional Internet is IP-address routing. That is, it is a host-to-host-based transmission for data delivery. In contrast, NDN adopts the innovative routing concept, content-to-content routing. It deals the name prefix of the content acting as routing information, not the IP address which host has the content. NDN has a content-aware routing to transmit the information based on the content name. An example of content publishing and access in NDN is shown in Fig. 1.

Fig. 1 An example of content publishing and access in NDN.

There are two content providers, A and B, and a content requester in Fig. 1. Provider A publishes a content which name prefix is “/ndn/media/movie01” to node A. Provider B also publishes two contents named as “/ndn/media/art01” and “/ndn/media/movie01” to node B respectively. The two providers only deal the content publishing processes without telling others where they are in NDN. When a content requester wants to the content which name is “/ndn/media/movie01”. It sends a request with the name prefix of requesting content to NDN. The nodes in NDN will flood the request to their neighbour nodes to query which one has the content. If a node has the content, it will reply the content to the original requester through the opposite direction of the content requesting path. Besides, each node located at the path will cache the content for quickly respond the same content requests from other requesters. Thus, the requester does not understand the original location of content providers. It just sends the content request based on name prefix to NDN and then it will get the content from NDN.

There are two main packets in NDN, interest packet and data packet shown in Fig. 2[7]. The structure of interest packet has three fields, content name, selector, and nonce. The field of content name stores the name prefix the user wants to access the content. The selector field...
indicates the preference setting of content requesting and the nonce field is used to verify the content integrity for the security issue. Thus, Interest packet is used to encapsulate the content requesting with name prefix to request the content in NDN. In contrast, the data packet is used to deliver the requested content sending back to the original requester. The content name field indicates the name prefix for responding content. The two fields of signature and signed info are used for data signature and encryption. The last data field stores the data chunk. NDN content may be divided into several chunks to fit the limitation of packet size. They have the same name prefix. It can easily to combine data chunks restoring the original content in NDN.

<table>
<thead>
<tr>
<th>Interest Packet</th>
<th>Data (Content) Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Name</td>
<td>Data</td>
</tr>
<tr>
<td>Signature</td>
<td></td>
</tr>
<tr>
<td>(digest algorithm, witness, …)</td>
<td></td>
</tr>
<tr>
<td>Nonce</td>
<td></td>
</tr>
<tr>
<td>(order preference, publisher filter, scope, …)</td>
<td></td>
</tr>
<tr>
<td>Singed Info</td>
<td></td>
</tr>
<tr>
<td>(publisher ID, key location, stale time, …)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2 Structures of Interest and Data Packets.

A NDN node has three main components, Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB)[7]. CS is storage for storing the published content in NDN. Data chunks routed by a node are also cached in CS for sequentially responding data request of the same content without directly accessing from the originated provider.

PIT keeps the routing information of Interest packets which content is not found in the NDN node. If the node has the content matching to the content name field of Interest packet, the node will immediately reply the requested content retrieved from CS to the content requester. In contrast, if the requested content is not in the node, PIT will store the packet information and flood the packet to other outgoing interfaces for finding the requested content. When the requested data packet comes back to the node and its content name is matched to the pending information of Interest packet in PIT, the entry will be removed. The reason is that node has cached the requested content. It can directly respond the following requests of the same content without flooding it to other nodes.

The functionality of FIB is similar to the routing table of a traditional IP router. It stores the routing information of content-interface mapping to decide which outgoing interface will send the packet out. Fig. 3 shows the processing flows of Interest packet and Data packet based on name prefix in a NDN node.

From above described, name prefix is a key feature for publishing, searching, requesting and replying the named content in NDN. However, the length of name prefix is variable and not limited. If the length of name prefix is too long, the space of content name field in Interest packet and Data packet will be larger. The size of packet is also too large respectively.

To shorten the length of name prefix, it can increase the efficiency of the prefix lookup and saving the storage of name prefix respectively. Cyclic Redundancy Check (CRC)[8] is a hash function used to generate a fixed length of numeric number to verify the transmitted data is correct or not over the communication network. Equation (1) shows a generic CRC function.

\[ M(x) \cdot x^t = Q(x) \cdot K(x) - R(x) \tag{1} \]

\( M(x) \) is a dividend, the original information with shifted n-left digits. \( K(x) \) is the divider which represents a CRC polynomial, \( Q(x) \) is a quotient polynomial, and \( R(x) \) is a checksum. That is, information of \( M(x) \) divided by \( K(x) \) can be calculated the checksum \( R(x) \). There are several CRC functions, such as CRC-8, CRC-16, CRC-32, and CRC-64. The main difference is the value and length of hash functions. Therefore, a better hash function can efficiently hash the longer information into a fixed length of checksum.

This paper adopts the CRC function to encode the name prefix and shorten a fixed length. The main reason is the CRC-based hash function is well-known and widely used in the world of Information and Communication Technology (ICT). It can also speed up the encoding process.

Fig. 3 Processing Flows of Interest and Data packets in a NDN node.
with hardware support[9]. But it is an important issue which kind of CRC function is better and that can efficiently encode the name prefix in NDN.

3. CRC-32 NAME PREFIX ENCODING

The length of Information can be reduced after CRC-based hashing computation. The encoded information can be represented the original information while data collision is not occurred with other information.

This paper adopts the version of CRC function, CRC-32-IEEE 802.3[10], to shorten the name prefix. The polynomial of CRC-32-IEEE 802.3 is $x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. It also can represent the value, 0x04C11DB7 in hexadecimal form. It will be inverted to 0xEDB88320 for conveniently calculating the hash value of name prefix.

A name prefix generally consists of several segments concatenated by slash symbol, such as"/com/onlinemd/media/video/funny/movie01.mpg". The length of name prefix is forty-five bytes. The first two segments indicate the content provider, the third, fourth and fifth segments describe the type of media categories, and the last segment is content name. The name prefix length is 45.

In order to retain the original naming structure of name prefix, the slash symbol will not be encoded and each segment will be encoded by CRC-32-IEEE 802.3.

After segment encoding, all of encoded segments will be concatenated again to form an encoded name prefix with the same name structure. The above name prefix is encoded to \texttt{/64B6C6E6/531DB98B/6A2CA10C/7CC7DA2C/1C5F4CC5/2188CC4B"} in hexadecimal fashion. In fact, the length of each encoded segment is four bytes. The total length of encoded name prefix included concatenated slash symbols is twenty-four bytes. Thus, it can reduce twenty-one bytes, about 46.47 percentage of the length of name prefix used by CRC-32-IEEE 802.3. Fig. 4 shows the operational flow of encoding name prefix based on CRC-32 IEEE 802.3.

4. EXPERIMENT OF NAME PREFIX ENCODING

For evaluating the efficiency of CRC-based name prefix encoding, two factors, storage space and searching time of name prefix, are used to measure the efficiency of our proposed integrated approach.

DMOZ (http://www.dmoz.org/) is an open directory project which is a human-edited directory of the website. It collects and categorizes a lot of available URLs for easily accessing the website. There are 16 top-level
categories defined by DMOZ including arts, games, kids and teens, reference, shopping, business, health, news, regional, society, home, recreation, science, and sports.

DMOZ is also an M-way tree hierarchical structure. It is identical to the storing structure of name prefix used in NDN. Thus, many researches[11][12][13] have adopted the collected URLs of DMOZ to be an experimental dataset for evaluating the searching and matching performance of name prefixes and URLs.

A web spider is used to crawl along the web content of DMOZ for collecting the dataset in this paper. The URLs collected in DMOZ are too huge. This paper only selects a subset of URLs which content language is English to crawl. There are 1417230 crawled URLs in March, 2014.

The collected URLs from DMOZ are constructed in an M-way tree simulating to store the name prefixes into CS in a NDN node. After setting the dataset, ten thousands Interest packets are generated. The content name of Interest packet is randomly assigned from the collected URLs of DMOZ. A NDN node will receive the Interest packets and then search the related named data based on name prefix in its CS. The experimental result is shown in Table 1.

<table>
<thead>
<tr>
<th>Methods</th>
<th>NDN</th>
<th>CRC-32-NDN</th>
<th>Reducing Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total size of name prefixes</td>
<td>970KB</td>
<td>618KB</td>
<td>26.29%</td>
</tr>
<tr>
<td>Average length of name prefixes</td>
<td>98.32B</td>
<td>34.63B</td>
<td>64.77%</td>
</tr>
<tr>
<td>Average segments of name prefixes</td>
<td>6.92 Segments</td>
<td>6.92 Segments</td>
<td>0%</td>
</tr>
<tr>
<td>Average times of name prefix lookup</td>
<td>7.93 Times</td>
<td>7.93 Times</td>
<td>0%</td>
</tr>
<tr>
<td>Average access time of name prefix lookup</td>
<td>51278.8ns</td>
<td>40188.2ns</td>
<td>21.63%</td>
</tr>
</tbody>
</table>

The total size of original name prefixes is 908 kilobytes. After CRC-32 encoding, the total encoded size of name prefixes is 618 kilobytes. It reduces the storage space of name prefix over 26.29% based on our integrated CRC-32 encoding scheme. Average length of originated name prefix is 98.32 Bytes. Average length of encoded name prefix is 34.63 Bytes. It also reduces over 64.77% length of name prefix. Thus, our proposed name prefix encoding scheme can efficiently reduce the storing space in NDN.

There are two experimental values are identical between original and encoded name prefixes. Average segments of name prefixes are 6.92 segments and average times of name prefix lookup are 7.93 times. It indicates that our proposed approach adopts the same data structure and operation flow of name prefix lookup compared with the original structure in NDN.

The average access time of original name prefix lookup is 51278.8 nanoseconds. The average access time of encoded name prefix lookup is 40188.2 nanoseconds. The reducing rate of average access time is 21.63%. It can efficiently improve the access time of name prefix lookup in our integrated CRC-32 encoding scheme.

As described above, the experimental result indicates that our integrated name prefix encoding scheme based on CRC-32 performs better than original one used in NDN. It also uses the originated data structure and operational flow of name prefix without additional design. Thus, our proposed approach can easily realize in the NDN implementation issues.

5. CONCLUSIONS

Named Data Networking is a content-centric network. It uses the content name, called name prefix, being the network address. That is, a requester sends an Interest packet containing the name of requesting content to NDN. It does not understand the location of requesting content and which nodes can provide the requesting content. Thus, name prefix is a key feature for accessing the named content in NDN.

The length of name prefix is variable and not limited in NDN. In order to enhance the efficiency of name prefix lookup, this paper adopts a CRC-32 encoding scheme to shorten the length of name prefix. The experimental results show that our proposed approach can efficiently reduce the storing space and improve the access time of name prefix lookup.

Although the length of name prefix encoded by CRC-32 can be shortened, the collision of encoded name prefix may be occurred. It may lead to access the wrong named content. For avoiding the encoded collision of different name prefixes, the process of false positive should be considered to choose a proper CRC function for
shortening the length of name prefixes without collision occurring in the future.

**ACKNOWLEDGMENT**

This work was partially supported by National Science Council of Taiwan (R.O.C.) under Grants NSC 102-2221-E-415-017 and MOST 103-2221-E-415-020.

**REFERENCES**


