

## Research on the Method of Implementing Named Data Network Interconnection Based on IP Network

Yabin Xu<sup>1,2,\*</sup>, Lufa Qin<sup>2,\*</sup> and Xiaowei Xu<sup>3</sup>

<sup>1</sup>Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University, Beijing, 100101, China

<sup>2</sup>School of Computer, Beijing Information Science and Technology University, Beijing, 100101, China

<sup>3</sup>Department of Information Science, University of Arkansas at Little Rock, Little Rock, 72204, USA

\*Corresponding Author: Yabin Xu. Email: xyb@bistu.edu.cn

Received: 05 March 2022; Accepted: 08 April 2022

**Abstract:** In order to extend the application scope of NDN and realize the transmission of different NDNs across IP networks, a method for interconnecting NDN networks distributed in different areas with IP networks is proposed. Firstly, the NDN data resource is located by means of the DNS mechanism, and the gateway IP address of the NDN network where the data resource is located is found. Then, the transmission between different NDNs across the IP network is implemented based on the tunnel technology. In addition, in order to achieve efficient and fast NDN data forwarding, we have added a small number of NDN service nodes in the IP network, and proposed an adaptive probabilistic forwarding strategy and a link cost function-based forwarding strategy to make NDN data obtaining the cache service provided by the NDN service node as much as possible. The results of analysis and simulation experiments show that, the interconnection method of NDN across IP network proposed is generally effective and feasible, and the link cost function forwarding strategy is better than the adaptive probability forwarding strategy.

**Keywords:** NDN; IP network; named data network interconnection; adaptive probability forwarding strategy; link cost function forwarding strategy

### 1 Introduction

At the beginning of the Design of the Internet, the application requirements are mainly file transfer and sharing between hosts. After decades of development, the application requirements of today's Internet are mainly content distribution and data acquisition, which is quite different from the original design goal. As a result, data transmission over the Internet is inefficient, network security problems are prominent, and IP addresses are exhausted. In order to fundamentally solve these problems, Zhang et al. [1] proposed the concept of NDN for the first time. Since then, NDN has been widely recognized and entered a period of rapid development. NDN aims at efficient distribution and



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

acquisition of information and takes content name instead of IP address as routing core, thus realizing the transformation of network from “machine interconnection” to “information interconnection”.

In NDN, there are two types of packets: interest packets and data packets. When users request data, they send interest packets to the network, and the network uses data packets to meet interest packet requests. Interest packets and data packets contain globally unique content names, and they are matched by content names. Each routing node in the NDN maintains three tables: PIT table, FIB table and CS table. The PIT table records the content name and access interface of each Interest package passing through this node. The FIB table records the forwarding interfaces corresponding to different content names. The CS table is used to cache data. Because the data is no longer associated with the host IP address and port number, the cached data in the routing node can meet the requirements for the same request. In this way, NDN can greatly reduce the transmission of redundant data in the Internet. In addition, NDN also has better performance than traditional networks in mobility and security.

With the deepening of the research, the theory of NDN and related protocols gradually mature, but how to practical application of NDN has become an urgent problem to be solved. The analysis shows that there are two main reasons restricting the development of NDN:

- (1) The current Internet infrastructure and applications are designed based on TCP/IP mode, which is fundamentally different from NDN. Although NDN has many advantages over IP network, it is not realistic to directly replace the existing network with NDN.
- (2) The data transmission form of NDN is “pull” data. That is, the data requester first sends interest packets and “pull” back the data packets to be requested. However, IP network is an end-to-end “push” mode, which leads to the incompatible deployment of NDN and IP network.

If NDN wants to be applied in practice, it must go through the stage of integration and coexistence with IP network. Therefore, the research on the fusion of two networks has become one of the key issues in the field of NDN research. At present, the academic circle has put forward some methods about NDN and IP network fusion. They fall into two categories.

- (1) On the premise of not changing the traditional terminal equipment, the main body of the network is gradually replaced with NDN. The key of this data conversion mode is how to convert the data format of IP network into the data format of NDN.

Moiseenko et al. [2] implements the method of transmitting TCP segments in the NDN by placing the IP address, TCP sequence number, TCP acknowledgement number, and fragmentation information in the name. Refaei et al. [3] proposed a method of converting the transport layer data and the NDN data in the IP network by using a configuration file to provide a socket-to-name mapping. Kim et al. [4] realized the integration of IP network and information center network by applying SDN (Software Defined Network) switch and SDN controller to the information center network. The literature [5–8] proposed a method for converting the data format of the application layer HTTP protocol with the data format of ICN.

Since traditional terminal devices and applications are designed based on TCP/IP architecture, this converged deployment method only converts IP network data packets into a form for transmission, and cannot effectively play the advantages of NDN.

- (2) Without change the main part of the network, and gradually deploy NDN at the edge of the network. In this way, the IP network changes little and the existing network devices can be well protected. Therefore, there is little resistance to the practical application of NDN, which is a

feasible method. However, the key is how to transmit NDN data without IP address through IP network.

Zuraniewski et al. [9] and Vahlenkamp et al. [10] proposed a scheme for integrating NDN and IP network based on SDN. The forwarding path of the NDN data is generated by the SDN controller, and the transmission of the NDN data across the IP network is realized. However, since SDN is not promoted in IP network, deploying NDN based on SDN is not realistic. Further, the method of NDN deployed at the network edge, which can only exert a small advantage in the interior of the NDN, could not play a role in a wider range.

Of the two schemes, we prefer the second.

Inspired by the evolutionary deployment mode of IPv6 in traditional network, a design scheme to realize the interconnection between different NDN by means of IP network is proposed. First, a gateway is set at the junction of each NDN and IP network to realize data conversion between the two networks. Then, by means of DNS mechanism, NDN data resources are located to find the gateway IP address of the NDN network where the data source resides. Finally, NDN data is transmitted through tunnels between different NDN gateways.

In addition, in order to improve the efficiency of transmitting NDN data in the IP network, we also added a small number of NDN service nodes to the IP network. Adaptive probability forwarding strategy and link cost function forwarding strategy are proposed. The adaptive probabilistic forwarding strategy adjusts the forwarding probability of different links according to the real-time delay, and finally aggregates NDN data to the link with high cache hit ratio, low latency and NDN service. The link cost function forwarding strategy firstly calculates the data forwarding cost of different links and selects the link with low forwarding cost to achieve efficient data transmission.

## 2 NDN Transmission Method Across IP Networks

### 2.1 Locating Resources

For edge NDN, the traditional IP network in the middle is almost transparent. However, with the increase of the number of local NDN networks, how to realize the location of NDN data resources is a problem [11]. That is, when a gateway needs to send NDN data to an IP network, it needs to determine the NDN data source at which gateway is connected to the NDN.

In fact, if NDN goes into practical use, there will be many local NDN on IP network. In order to realize the interconnection between different NDN, the gateway needs to know that the NDN data request containing a certain content name prefix can be satisfied at which NDN connected by gateway. Therefore, a resolution mechanism between NDN names and IP addresses of gateway needs to be designed.

Because the content name in the named data network is also a hierarchical structure, a top-level domain name branch can be added to the domain name system of the IP network by means of Domain Name System (DNS) to provide the mapping service between the content name and the IP address of the gateway.

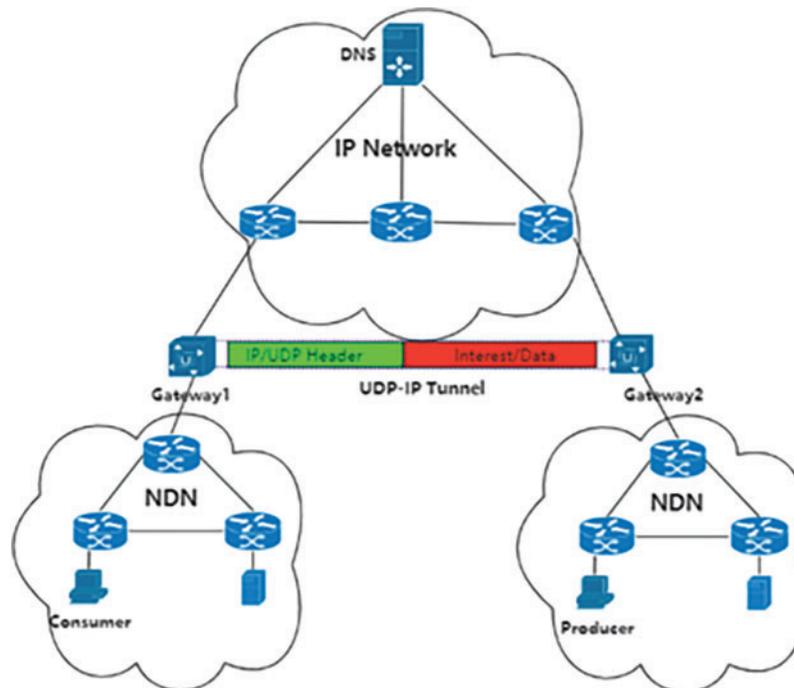
Assume that the full name of an NDN interest packet request is “/www.ndnsim.ndn/doc/%FD%23”. When the gateway receives an interest packet containing the name, it first resolves the name prefix of the interest packet, which is “/www.ndnsim.ndn/”. Then, the NDN name prefix is stored in the query question field of DNS data, and DNS request packets are sent to the domain name system.

According to the recursive URL resolution process on traditional networks, the DNS system resolves the IP address of the data source (for example, 10.1.2.3), encapsulates the IP address in the answer area, and sends DNS packets to the gateway. The gateway then obtains the data source IP address (10.1.2.3) contained in the DNS response.

At this point, the positioning of NDN resources is complete.

## 2.2 Data Transmission

For the convenience of expression, Fig. 1 shows a schematic diagram of data transmission between two NDN's across IP network.



**Figure 1:** Diagram of NDN transmission across IP networks

When a named data network generates a new content name prefix, the owning gateway notifies the domain name system to record the mapping between the new name prefix and the gateway IP address. When other named data networks request this content, the gateway is first responsible for sending resolution requests to DNS. After receiving the resolution result from the DNS server, the gateway obtains the IP address of the destination gateway. The gateway then encapsulates the NDN interest packet into the UDP data portion and sends it to the Internet. After receiving the UDP data, the destination gateway extracts the UDP data, that is, the NDN data, and records the mapping between the source IP address and the content name, and then sends the data to the local named data network. Data is returning in the same way.

### 3 Data Forwarding Based on NDN Service Node

#### 3.1 Discovery of NDN Service Nodes

The main applications of the current Internet, including WWW, video, audio, file download and other data transmission services, all have a large number of repeated data transmission problems [12], resulting in the increase of unnecessary transmission delay and low efficiency of data transmission. Therefore, we can add a few NDN service nodes into the traditional network. However, how to make the gateway aware of the presence of these devices and take advantage of them is a key problem.

Therefore, we maintain a corresponding relation table between IP address and geographical location in the domain name system. When the gateway queries the IP address of the target gateway in the domain name system, the IP address of the NDN service node can be added to the field of additional resource record.

When the domain name system receives a request from the gateway, it can calculate some NDN service nodes whose geographical locations are close to the NDN service nodes by traversing the mapping table between IP addresses and geographical locations, and give their corresponding IP addresses. The returned multiple NDN service nodes can be selected by the strategy proposed later.

Assume that two NDN service nodes are obtained and their corresponding IP addresses are 100.4.5.6 and 200.7.8.9 respectively. The response instance of DNS carrying the IP address of NDN service node is shown in Fig. 2. In Fig. 2, the IP addresses of two NDN service nodes are encapsulated in additional region fields.

|                |    |       |    |
|----------------|----|-------|----|
| 0              | 15 | 16    | 31 |
| Transaction ID |    | Flags |    |
| 1              |    | 1     |    |
| 0              |    | 2     |    |
| www.ndnsim.ndn |    |       |    |
| 100. 1. 2. 3   |    |       |    |
| 100. 4. 5. 6   |    |       |    |
| 200. 7. 8. 9   |    |       |    |

Figure 2: NDN service node carries the IP address response

#### 3.2 Data Forwarding Strategy

Because named data network can greatly reduce repetitive information transmission, and delay is the most concerned indicator of user experience, this paper focuses on the delay factor.

In order to effectively utilize NDN service nodes distributed in traditional networks, the adaptive probabilistic forwarding strategy and link cost function forwarding strategy are designed.

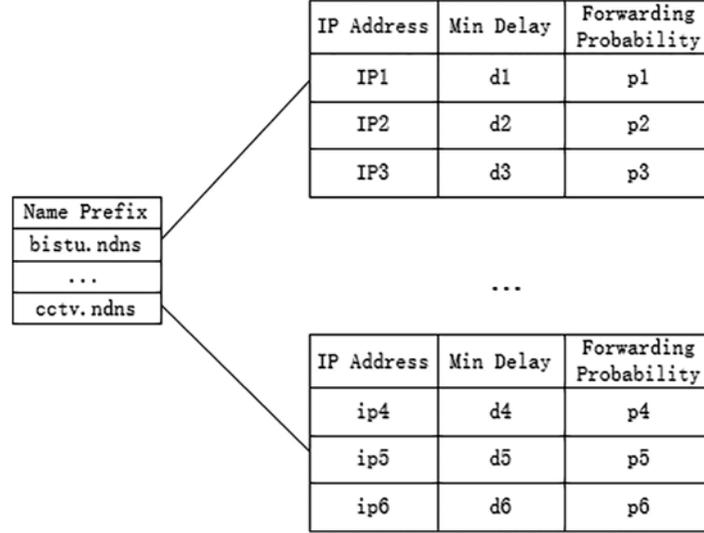
##### 3.2.1 Adaptive Probabilistic Forwarding Strategy

The analysis shows that, under the condition of no link congestion, the links with NDN service nodes, high cache hit ratio and fast processing speed, have naturally low latency, so such links should be preferred. Therefore, we design an adaptive probabilistic forwarding strategy.

Firstly, gateway node and NDN service node maintain the forwarding information table as shown in Fig. 3. If name prefixes are not distinguished, requests with different name prefixes will be forwarded

to the same NDN service node, resulting in frequent cache updates, which in turn reduces cache hit ratio and increases latency. To do this, we distinguish between different prefixes. Different name prefixes are maintained in the name prefix table. Each name prefix corresponds to a forwarding information table. Among them,

$$\sum_{i=1}^n p_i = 1 \quad (1)$$



**Figure 3:** Forward information table

In the initial state,  $p_i = 1/n$ ,  $n$  is the number of optional paths for a name prefix. The gateway preferentially selects the link that has not passed. That is, the link of  $d_i = -1$ . It sends out the data, and records the time when each packet is sent. When a data packet returns, calculate the delay and update  $d_i$ .

When a link with an NDN service node hits the cache, the latency of the request is much lower than the normal delay of the request to the data source. Therefore, the link minimum delay can be recorded to determine whether the cache is hit. In addition, due to the fluctuation of network delay, the following probability update strategy is adopt:

$$if\ dc \in [d_i - \Delta d, d_i + \Delta d], p_i + r \quad (2)$$

where  $dc$  is the delay of a piece of data, and  $\Delta d$  is the degree of link fluctuation, which takes 0.3 ms here.  $R$  represents the probability update value. Then, the probability of forwarding information under the name prefix is normalized.

The probability update value is a problem worth exploring [13]. If the value is too large, the link is unstable. If the value is too small, the speed of link aggregation to the optimal link is too slow. Therefore, this paper adopts the following method to determine the value of  $R$ :

$$\exists r \in [0, 1], \min \left\{ f(r) = \frac{1}{n} \sum_{j=1}^n d_j \right\} \quad (3)$$

That is to find an optimal  $r$  value to minimize the average delay of  $n$  packets transmitted on the network. According to the experiment, when  $r$  is 0.015, the average delay is minimum. After a period of data transmission, the link with high cache hit ratio and low latency is automatically selected. However, if all data is aggregated into one link, the link is highly likely to be congested.

For this reason, we can refer to the Additive Increase and Multiplicative Decrease strategies (AIMD) used in transport layer congestion control in IP networks. That is, if the delay of a link returning data is greater than the threshold or packet loss occurs, the link probability is halved and the probability is normalized. In addition, when several consecutive packets sent to an IP address are not responded to, the link to the IP address is determined to be faulty, and the time when the fault occurs is recorded. After two seconds, try to enable the link.

### 3.2.2 Link Cost Function Forwarding Policy

First, maintain a routing table as shown in [Tab. 1](#).

**Table 1:** Routing information table

| IP address      | R              | Priority       | Hop count      | CD                   |
|-----------------|----------------|----------------|----------------|----------------------|
| IP1             | 100            | 1              | 3              | 0.202                |
| IP2             | 50             | 10             | 1              | 0.375                |
| IP3             | 300            | 10             | 2              | 0.053                |
| ...             | ...            | ...            | ...            | ...                  |
| IP <sub>i</sub> | R <sub>i</sub> | P <sub>i</sub> | H <sub>i</sub> | CD <sub>i(t+1)</sub> |

Where,  $R_n$  indicates the rate limit of a link.  $P_n$  indicates the priority of a link, which is set to 1 for common links and 10 for links with NDN nodes.  $H_n$  stands for hops;  $CD_n$  (Congestion Degree based on RTT) represents the degree of path congestion. The specific calculation method is given in Section 3.2.3.

Then, the gateway calculates the link forwarding cost according to the routing information table. To forward data, select the link with the lowest forwarding cost. The cost function is as follows:

$$f(R_n, P_n, H_n) = w_1 / R_n + w_2 / P_n + w_3 * H_n + w_4 * CD_n \quad (4)$$

where,  $w_1, w_2, w_3, w_4$  are weight parameter. The weight value of each attribute is determined through Analytic Hierarchy Process (AHP). Firstly, according to the importance of different attributes, the hierarchical discriminant matrix is obtained as shown in [Tab. 2](#). Then, the weight of each attribute are can be obtained through consistency test. As shown in [Tab. 3](#).

### 3.2.3 Calculation Method of Path Congestion

Network resources do not always meet user requirements. When user requirements exceed the network processing capacity, congestion or even packet loss occurs. Literature [14] proposed a method to calculate congestion degree according to one-way delay. This method is used in this paper to calculate the path congestion degree by using Round Trip Time (RTT) in the fusion scenario of named data network and traditional network.

**Table 2:** The hierarchical discriminant matrix of forwarding cost function weight

|           | Bandwidth | Priority | Hop count | CD  |
|-----------|-----------|----------|-----------|-----|
| Bandwidth | 1         | 1/5      | 1/3       | 1/9 |
| Priority  | 5         | 1        | 5         | 1/9 |
| Hop Count | 3         | 1/5      | 1         | 1/9 |
| CD        | 9         | 9        | 9         | 1   |

**Table 3:** Weighted value

| wi | value    |
|----|----------|
| w1 | 0.044965 |
| w2 | 0.195837 |
| w3 | 0.083612 |
| w4 | 0.675586 |

Set RTTmax as the maximum round-trip time on path  $i$ , and RTT min as the minimum round-trip time, then the variation range of round-trip time on path  $i$  can be obtained as  $R_i$ :

$$R_i = RTT_{\max} - RTT_{\min} \quad (5)$$

In the process of data forwarding, the congestion degree of the path at the next moment needs to be estimated, so the round-trip time of data packet transmission at the next moment needs to be predicted. Jacobson [15] proposed an estimation method of  $RTT_{n+1}$ , namely:

$$\overrightarrow{RTT}_{n+1} = \alpha * \overrightarrow{RTT}_n + (1 - \alpha)M_n \quad (6)$$

where,  $M_n$  is the measured round-trip time of the current data packet. Let  $\overrightarrow{RTT}_2 = M_1$ , thus get the round trip time of the next data request,  $\alpha$  take the experience value 0.1.

Set the congestion level of path  $i$  at the next moment as  $CL_i$ . According to formulas (5) and (6), the following equation can be obtained:

$$CL_i = \frac{\overrightarrow{RRT}_{n+1} - RRT_{\min}}{R_i} \quad (7)$$

Obviously,  $CL_i \in [0, 1]$ ,  $CL_i$  directly reflects the round-trip time of the next packet, and indirectly reflects the congestion of path  $i$ .

In addition to considering the range of round-trip times, the trend is also need to be considered. Set the congestion trend  $CT_i$  of path  $i$  as the ratio of the round-trip time of adjacent data requests, i.e.,

$$CT_i = \frac{RTT_{n+1}}{RTT_n} \quad (8)$$

Obviously, the value of  $CT_i$  is greater than or equal to 1, which can also reflect the change trend of the congestion level of path  $i$ . Then, based on the change range and change rate of round-trip time,

$CDTi(t)$  of path  $i$  at time  $t$  can be calculated as follows:

$$CDTi(t) = \alpha * CLi(t) + (1 - \alpha) * [1 - e^{-CTi(t)}] \quad (9)$$

where,  $\alpha$  is an adjustable parameter, indicating the proportion of the congestion level and congestion trend in calculating the congestion level.  $CDTi \in [0, 1]$ ,  $\alpha$  get experience value of 0.8. In addition, referring to the TCP packet loss to congestion window adjustment scheme, when packet loss occurs on path  $i$ , the path congestion can be increased by two times. The path congestion degree (CD) is:

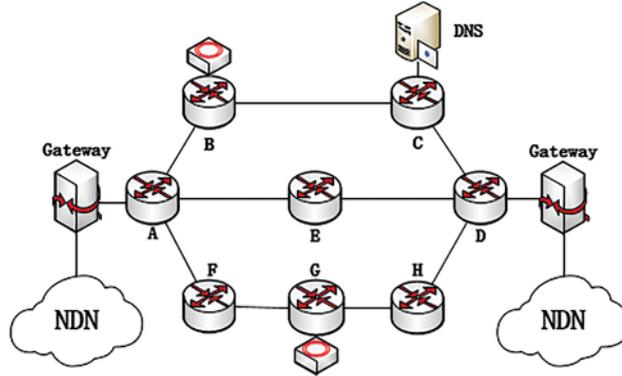
$$CDi(t+1) = \begin{cases} CDRi(t+1) & \text{no packet loss happend} \\ 2 * CDRi(t+1) & \text{packet loss happend} \end{cases} \quad (10)$$

In this way, the gateway maintains the routing information table and selects the link with the lowest forwarding cost by calculating the link forwarding cost. When the link to the target node is about to be congested, the path congestion increases, which leads to the increase of link forwarding cost, and then the transmission will be switched to other better link.

## 4 Experiment

### 4.1 Experimental Environment

All the experiments in this paper were carried out in NDN simulation experiment environment NDNSim [16], and the network topology is shown in Fig. 4.



**Figure 4:** Experimental network topology

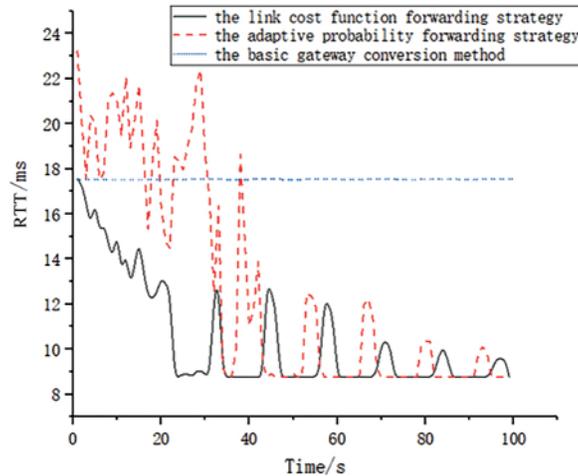
In Fig. 4, the main part of the network is the IP network, which is composed of 8 routing nodes, among which 2 nodes can provide NDN service [17]. Since ndnSim does not have the function of domain name resolution, the open source software bind is installed in the virtual machine to make the nodes in ndnSim communicate with Linux virtual machine, so as to achieve the function of domain name resolution. In addition, there are 2 local NDN networks connected to the main body of the IP network, and the request of the left NDN can be answered at the right NDN.

Set the link delay between the NDN and the gateway and between the gateway and the router to 1 ms, the remaining delay to 2 ms, and the network bandwidth to 5Mbps. Without considering the transmission delay, theoretically, the delay of the left local NDN from sending interested packets to receiving packets is 20 ms along the A-B-C-D link, and 8 ms when hitting the cache. It is 16 ms along the A-E-D link. If the cache is not hit along A-F-G-H-D, the value is 28 ms. If the cache is hit, the value is 12 ms.

## 4.2 Experimental Results and Analysis

### 4.2.1 Round-trip Time Comparison Experiment

Round-trip time is the most direct indicator to measure the quality of a forwarding strategy. On a traditional network, the round-trip time is the time between the time the sender sends data and the time the sender receives an acknowledgement from the receiver. In the fusion scenario of NDN and traditional network, round-trip time refers to the time from sending interest packet to receiving data packet. In the topology shown in Fig. 4, the cache size is 100, and the data of round-trip time of the three forwarding strategies with time changes are tested respectively. The experimental results are shown in Fig. 5.



**Figure 5:** Delay comparison of three strategies

As can be seen from Fig. 5, after joining NDN service nodes, both the adaptive probabilistic forwarding strategy and the link cost function forwarding strategy will be forwarded to the A-B-C-D link. Due to the existence of NDN service node cache, the large probability of interested packets will be satisfied and the requested data will be returned accordingly, so the delay is significantly reduced.

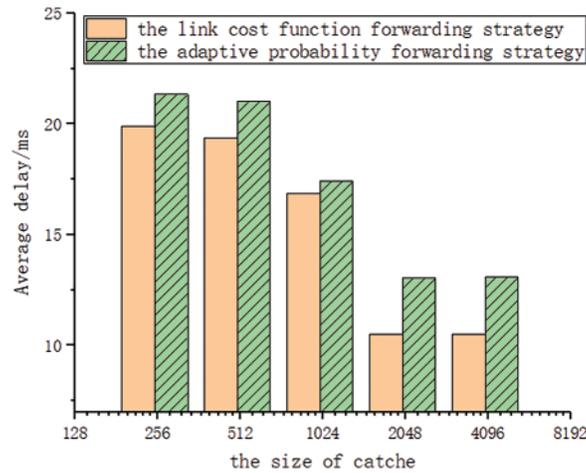
However, if the link with NDN service node is selected, and the data resource in the cache of NDN service node is not hit, the producer needs to request data from the corresponding NDN data source (producer). In this case, the delay time increases, which is reflected in the fluctuation of the delay time in Fig. 5. In addition, the delay fluctuation of the basic gateway transformation method was around 0.002 ms during the experiment, so the fluctuation was not obvious in Fig. 5.

Since the adaptive probabilistic forwarding strategy adjusts the probability dynamically, and the link cost function forwarding strategy can directly select the optimal link, the link cost function forwarding strategy is better in terms of delay performance.

### 4.2.2 Experimental Comparison of Delay Time Under Different Cache Capacities

We also tested round-trip times for different cache sizes. The experimental results are shown in Fig. 6.

Fig. 6 shows the delay comparison between adaptive probabilistic forwarding strategy and link cost function forwarding strategy under different cache sizes of NDN service nodes in the IP network.

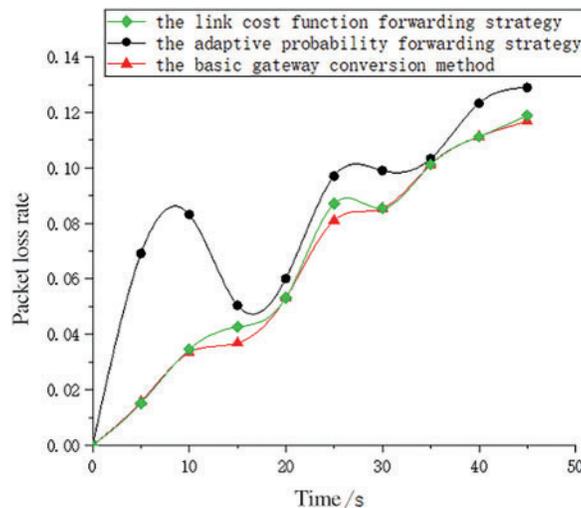


**Figure 6:** Latency comparison for different cache sizes

It can be seen that under different cache sizes, the average delay of data transmission in link cost function forwarding strategy is always lower than that in adaptive probabilistic forwarding strategy.

#### 4.2.3 Comparative Experiment of Packet Loss Rate

Packet loss rate is an important index to measure system stability. Therefore, the packet loss rate of the improved method and the basic method is tested in this paper, at different packet sending speeds. At the beginning of the experiment, the packet sending rate was zero, and then the packet sending rate increased by 40 per second. The experimental results are shown in Fig. 7.



**Figure 7:** Comparison graph of packet loss rate

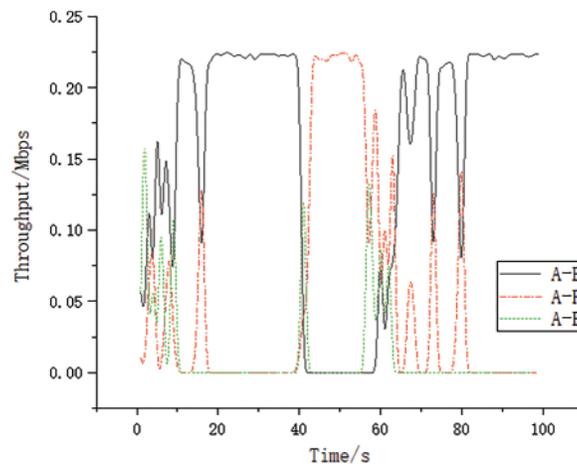
The adaptive probabilistic forwarding strategy and link cost function forwarding strategy tend to select links with NDN service nodes, which results in data re-encapsulation. Therefore, the packet loss rate of both strategies is higher than that of the basic gateway transformation method. In addition, the

adaptive probabilistic forwarding strategy selects links through probability adjustment, which leads to link instability. Therefore, the packet loss rate is higher than the link cost function forwarding policy.

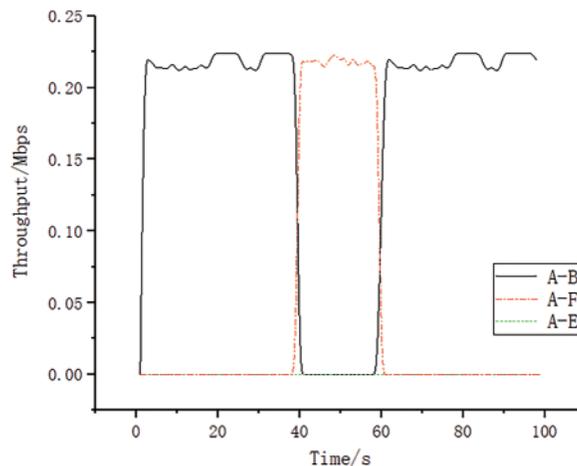
#### 4.2.4 Fault Tolerance Experiment

When a routing node fails and a link becomes unavailable, a good forwarding strategy can not only find the faulty link and forward data to other links, but also recover the link that passes through the node after the faulty node recovers. Therefore, this experiment tests the fault tolerance of the two forwarding strategies by setting node B to fail and recover again during the experiment and observing the throughput of each link.

Figs. 8 and 9 show the throughput changes of the three links measured under the adaptive probabilistic forwarding strategy and link cost function forwarding strategy respectively.



**Figure 8:** The three link throughput changes under link cost function forwarding strategy



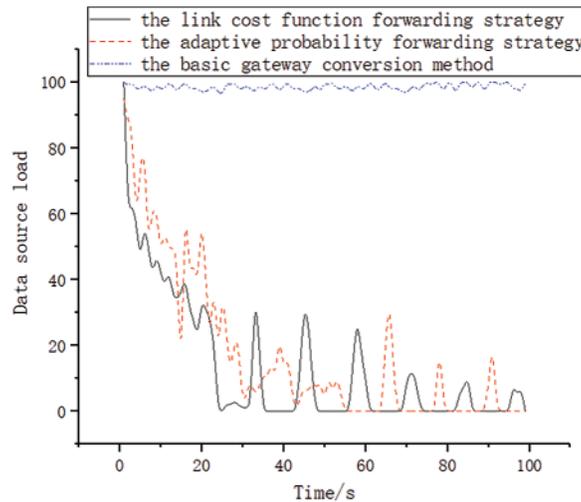
**Figure 9:** The throughput changes of three links under adaptive probabilistic forwarding strategy

It can be seen from Figs 8 and 9 that, in the 40 s, after node B fails, both forwarding strategies can avoid links with faulty nodes. And in 60 s, after node B recovers, it can be restored to the optimal link

A-B-C-D. It can be seen from the change of link throughput that, when the link changes, it takes a period of time for the adaptive probabilistic forwarding policy to find the optimal link, and the link stability is relatively poor.

#### 4.2.5 Comparative Experiment of Load Change of Data Source

In order to verify the effect of adding NDN nodes into the traditional IP network, the load changes of data sources before and after adding NDN service nodes in the network topology is tested in this paper as shown in Fig. 4. That is, the number of requests received by data sources per second. The experimental results are shown in Fig. 10.



**Figure 10:** Data source load variation

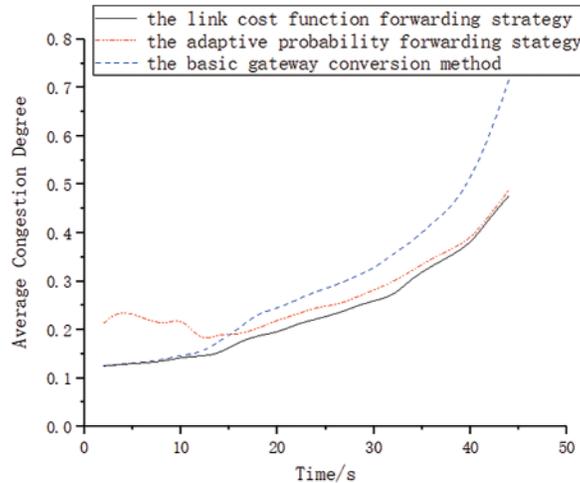
It can be seen that after NDN service nodes are added, the data source load of both strategies is greatly reduced, while the gateway transformation method basically has no significant change. The reasons are as follows: Due to the existence of NDN service node cache, popular content will be cached in NDN service node with the increase of data requests. Therefore, most data requests can be satisfied through NDN service nodes, and only a few rare data requests need to request data from data sources. As a result, the load on the data source is reduced to some extent. Meanwhile, it can also be seen from Fig. 10 that the data source load of the link cost function forwarding strategy decreases faster than that of the adaptive probabilistic forwarding strategy.

#### 4.2.6 Congestion Comparison Test

In order to measure the performance of the three forwarding policies in terms of congestion degree, the average congestion degree of the three forwarding policies is calculated using the method given in Section 3.2.3. The experimental results are shown in Fig. 11.

As can be seen from Fig. 11, at the beginning of the experiment, the average congestion of the adaptive probabilistic forwarding strategy is higher than that of the other two methods due to the instability of the link. As the rate of sending packet gradually increased, three strategies of congestion degree are increasing. However, because the link cost function forwarding strategy and adaptive probabilistic forwarding strategy make use of NDN service nodes in traditional IP network, most data

requests are satisfied at NDN service nodes. So, the average congestion degree of these two strategies is lower than that of basic gateway transformation method, and the growth rate is slower.



**Figure 11:** Congestion comparison

## 5 Conclusion

In order to promote the real implementation of named data network, a method of progressively deploying NDN on traditional IP network is proposed in this paper. That is, through the combination of tunnel technology and gateway, and the use of domain name resolution to achieve the location of NDN data resources, make different NDN transmitting across IP network become more possible. On this basis, the adaptive probabilistic forwarding strategy and link cost function forwarding strategy are proposed.

In order to prove the effectiveness of the proposed method, we set up an experimental environment and carried out a large number of comparative experiments. The experimental results show that, the link cost function forwarding strategy proposed in this paper performs better than the adaptive probabilistic forwarding strategy in terms of packet loss rate and data transmission delay, and can better achieve the desired goal. Next, bigger and complicated network topology will be used to test and the security of NDN will take into account [18,19].

**Funding Statement:** This research is supported by Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] L. Zhang, A. Afanasyev, J. Burke, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.
- [2] I. Moiseenko and D. Oran, "TCP/ICN: Carrying TCP over content centric and named data networks," in *Proc. of the 3rd ACM Conf. on Information-Centric Networking*, Kyoto, Japan, pp. 112–121, 2016.

- [3] T. Refaei, J. Ma and S. Ha, "Integrating IP and NDN Through an Extensible IP-NDN Gateway," in *Proc. of the 4th ACM Conf. on Information-Centric Networking*, Berlin, Germany, pp. 224–225, 2017.
- [4] S. Y. Kim, S. Robitzsch and D. Trossen, "Realizing IP-based services over an information-centric networking transport network," in *Proc. of the 3rd ACM Conf. on Information-Centric Networking*, Kyoto, Japan, pp. 215–216, 2016.
- [5] S. Wang, J. Bi, J. Wu, "On adapting HTTP protocol to content centric networking," in *Proc. of the 7th Int. Conf. on Future Internet Technologies 2*, Seoul, Korea, pp. 1–6, 2012.
- [6] Z. Li, J. Bi and S. Wang, "HTTP-CCN Gateway: Adapting HTTP protocol to content centric network," in *2013 21st IEEE Int. Conf. on Network Protocols (ICNP)*, Goettingen, Germany, pp. 1–2, 2013.
- [7] I. Moiseenko, M. Stapp and D. Oran, "Communication patterns for web interaction in named data networking," in *Proc. of the 1st ACM Conf. on Information-Centric Networking*, Paris, France, pp. 87–96, 2014.
- [8] X. Yang, "Design and Implementation on HTTP-NDN Gateway," *Ph.D. dissertation*, Tsinghua University, China, 2012.
- [9] P. Zuraniewski, N. Adrichem and D. Ravesteijn, "Facilitating ICN deployment with an extended openflow protocol," in *Proc. of the 4th ACM Conf. on Information-Centric Networking*, Berlin, Germany, pp. 123–133, 2017.
- [10] M. Vahlenkamp, F. Schneider and D. Kutscher, "Enabling ICN in IP networks using SDN," in *ICNP*, vol. 3, pp. 1–2, 2013.
- [11] Y. Shen, Y. Lu, Z. Wang, X. Xv, F. Qi *et al.*, "Dns service model based on permissioned blockchain," *Intelligent Automation & Soft Computing*, vol. 27, no. 1, pp. 259–268, 2021.
- [12] K. Lei, "Networking (ICN) and Named-Data-Networking (NDN)," Beijing, China: Beijing University Press, pp. 60–68, 2015.
- [13] A. Tariq, I. U. Din, R. A. Rehman and B. Kim, "An intelligent forwarding strategy in sdn-enabled named-data iov," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 2949–2966, 2021.
- [14] M. Liu, "Research and Design of the Flow Scheduling Algorithm in Multi-Path Transmission System," *Ph.D. dissertation*, Beijing University of Posts and Telecommunications, China, 2018.
- [15] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.
- [16] A. Afanasyev, I. Moiseenko and L. Zhang, "ndnSIM: NDN simulator for NS-3," Tech. Rep, 2012.4, [Online]. Available: <http://named-data.net/techreports.htm>.
- [17] S. S. Ullah, S. Hussain, A. Gumaei, M. S. Alhilal, B. F. Alkamees *et al.*, "A Cost-effective approach for ndn based internet of medical things deployment," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 233–249, 2022.
- [18] Y. Xu, P. Gu and X. Xu, "Research on detection method of interest flooding attack in named data networking," *Intelligent Automation & Soft Computing*, vol. 30, no. 1, pp. 113–127, 2021.
- [19] S. S. Ullah, S. Hussain, A. Gumaei and H. AlSalman, "A secure ndn framework for internet of things enabled healthcare," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 223–240, 2021.