

# Learning Forwarding Mechanism in Content-centric Networking

Hu Yao and Shigeki Goto

Department of Computer Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku, Tokyo 169-8555, Japan

E-mails: {huyao, goto}@goto.info.waseda.ac.jp

---

**Abstract:** In Content-centric Networking (CCN), data names are carried in packets without specifying IP addresses. This change leads to a new network forwarding model: CCN routers forward *Interest* (request) and Data packets using only their names. One major challenge is realizing intelligent forwarding of Interest packets over multiple available paths according to a Forwarding Information Base (FIB), which is currently configured manually with static routes. To address this issue, we propose a new FIB learning method by extending the current CCN prototype. With this new method, FIB entries can be automatically generated based on earlier successful data retrievals. We demonstrate that the proposed FIB learning forwarding mechanism is efficient and self-adaptive in adverse conditions such as link failure. This paper also illustrates an application example to demonstrate resilience when handling mobile nodes and effectiveness in changing network configurations.

**Keywords:** Content-centric networking; ICN; FIB; learning; multipath forwarding.

---

## 1. Introduction

According to a recent prediction [1], global IP traffic will reach 966 exabytes per year by 2015. Most of this traffic increase will come from peer-to-peer (P2P) communications and various forms of video traffic, such as TV and video on demand (VoD), which will account for approximately 90% of global consumer IP traffic. Global mobile data traffic is also expected to enormously increase in the same time frame.

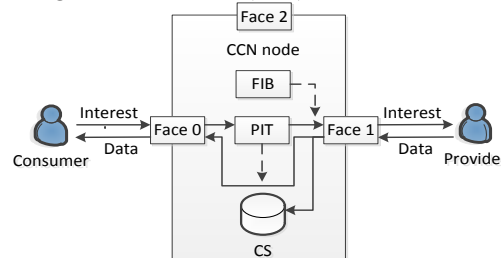
To deal with this growth in terms of data volume and devices, one solution is to deploy application layer overlays such as Content Delivery Networks (CDNs) [2] and P2P applications that cache content, provide location independent access to data, and optimize the delivery of

data. This solution should provide access to named data or objects, including replicated web resources, rather than the traditional host-to-host packet delivery model. However, these techniques would reside in overlay networks and it is difficult to realize the full potential of content-based distribution in today's IP-based platform.

Content-centric Networking (CCN) is a new networking approach that takes named data or objects as first-class network entity [3][4]. It leverages object naming and ubiquitous in-network caching to provide more efficient and robust network services. CCN will shift address space from one billion IPs to at least one trillion content names [5]. CCN can be perceived as a generalization of CDN technologies, and CCN deployment is feasible on a scale similar to CDN [6]. In addition, CCN is not limited to media distribution scenarios; it can include other scenarios such as data collection.

There are two message types in CCN: Interest and Data (Content), as shown in Figure 1. The Interest message is used to request data by name. Data (also called Content or Content Object) carries the requested data. An Interest message should identify a data chunk (piece) to retrieve it by specifying a full name or a name prefix with other restrictions that indicate acceptable data. A Data (Content Object) message contains a data payload, the identifying name of the data chunk, and the identification of the publisher.

Figure 1 illustrates a CCN node that contains the following data structures to provide ubiquitous caching and loop-free forwarding: Content Store (CS), Face, Forwarding Information Base (FIB), and Pending Interest Table (PIT).



**Figure 1. CCN Node**

The CS is a cache storage that maintains content or objects for future retrieval through lookup by names. It is implemented using popular cache replacement algorithms [7][8] that maximize the possibility of data reuse, such as First-in First-out (FIFO) or Least Recently Used (LRU). The Face is a generalization of the concept of interface. A face is a connection to a hardware communication link or to a software application process running on the same host as the CCN node. The FIB is a table that maintains outbound face information for Interests. These tables are used for longest-match prefix lookup by name. Each FIB prefix entry has a set of corresponding

faces rather than a single face. The PIT is a table for recording unsatisfied (pending) Interests. Each entry in the PIT has a list of source entries represented by incoming faces to the CCN node. The PIT has a timeout interval for unsuccessful Interest requests to avoid maintaining old Interest records indefinitely.

Information-centric network architecture has been developed in Europe. PSIRP/PURSUIT [9][10] preforms forwarding information in packets, while CCN stores routing information in CCN nodes. In CCN, Interest packets are forwarded following the FIB, while Data packets are forwarded according to the PIT, which guides Data packets back to consumers or requesting users. Thus, an excellent design of the FIB is essential for both Interest and Data packets transferred in CCN.

The remainder of this paper is organized as follows. Section 2 proposes our new method. Section 3 examines an experiment to evaluate the new method, and Section 4 concludes this paper.

## **2. Proposed Method**

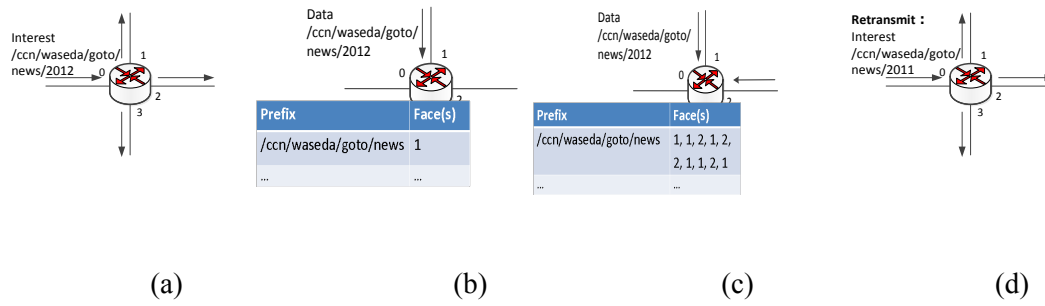
### *2.1. Forwarding Interest Base*

In CCN, generally, an FIB has multiple outbound faces or next-hop destinations. We need an intelligent forwarding strategy to select the best face among multiple faces according to some selection algorithm. Designing a scalable forwarding plan is essential for deploying CCN on a large scale. A popular CCN software prototype, CCNx [11], has been developed at PARC. The key component of CCNx is called the ccnd daemon, which implements a packet forwarding function. In the latest version of CCNx (0.7.1), the FIB still needs to be configured manually with static routes, which is appropriate for small experimental networks. However, manual configuration causes significant surplus traffic in larger networks. We require more sophisticated mechanisms to manage the FIB forwarding entries. Classical routing algorithms, such as OSPF [12], are possible solutions for this issue. However, OSPF, for example, does not make the best use of cached content (Data) records in the CS.

There are several factors to consider when designing or devising CCN forwarding strategies. Because CCN is not aware of host location or address, pure source routing, as is established in IP-based networking, is not feasible. CCN does not specify paths. It specifies only faces, therefore, it only determines the face to which an Interest packet is forwarded. Forwarding strategies should adhere to CCN forwarding principles and should maintain the balance of traffic because there is a one-to-one correspondence between Interest and Data packets.

### *2.2. Method for Forwarding Interest Packets*

We focus on two issues in our proposal. The first issue is how to construct FIB entries with a name prefix and face entries. The second is how to forward Interests to potential content providers or holders according to the FIB entries. The proposed FIB learning makes forwarding decisions based on FIB lookup results and the network environment such as response time. This FIB learning method is based on the statistics of incoming successful Data packets. However, it maintains a persistent ability to send Interests to a new face. Therefore, unbalanced traffic distribution is unlikely in the new FIB learning mechanism. It also responds quickly to changing network configurations in CCN. Figures 2 (a), (b), (c), and (d) show the steps in the proposed FIB learning mechanism.



**Figure 2.** Learning Forwarding Information Base.

#### [a] Arrival of New Interest

At the initial stage, the FIB on each CCN node is empty. An incoming Interest packet at Face 0 is propagated to all Faces (1, 2, and 3) except the incoming Face 0. This is an initial flooding.

#### [b] Arrival of the First Data Packet

When the first Data packet arrives, the corresponding FIB entry is created. The prefix of the Data name is stored in the Prefix field and the arrival face of the Data is recorded in the Face(s) field, which can maintain N maximum elements (faces).

#### [c] Arrival of Data Packet with the Same Prefix

When there is the second Data packet with the same prefix as an existing packet in the FIB, the arrival face will be added to the corresponding Face(s). If N faces are already stored in the set of Face(s), a FIFO operation is performed. That means that the first oldest arrival face will be deleted.

#### [d] Interest Forwarding Principle

When a new Interest packet arrives that matches Prefix entries in the FIB, Interest forwarding is performed by selecting one face among the elements in Face(s) according to the occurrence ratio of the faces. That is, the most successful face is selected. Since the FIB can maintain at

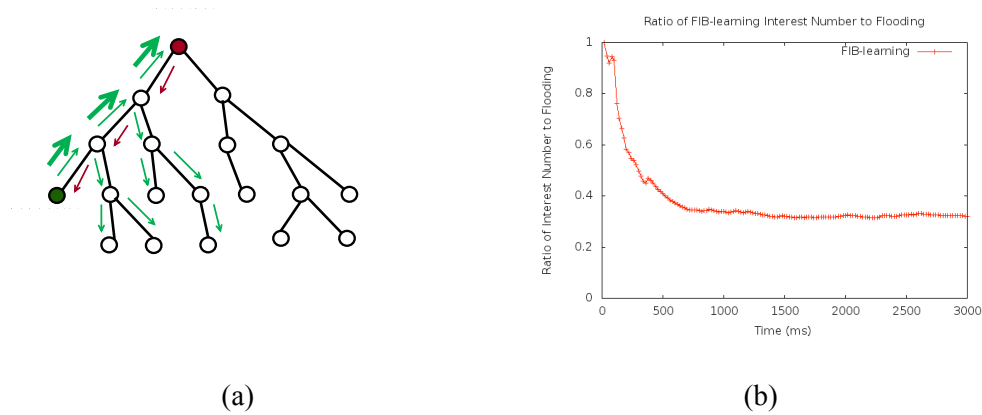
most  $N$  faces, the learning mechanism adjusts the face selection according to the recent data retrievals. In the case of link failure or packet loss, Data does not return in time; therefore, the data requester or consumer returns to step [a], flooding the Interest to all available connected faces to discover a working path quickly.

### 3. Evaluation of the Proposed Method

We designed three evaluation scenarios to examine the effectiveness and efficiency of the proposed FIB learning mechanism. The simulations evaluated how well the FIB learning mechanism works and whether it achieves robust packet delivery under adverse conditions and with changing network configurations. The experimental simulations were conducted in a NS-3-based environment, ndnSIM [13][14].

#### 3.1. Space and Time Tradeoff

We compared the FIB learning mechanism with a flooding method where the router simply forwards incoming Interest packets to each connected face. In the tree topology, shown in Figure 3 (a), the data Requester or consumer at the leftmost node requests data from the data Provider at the topmost root node by issuing 100 Interests per second. The data Provider returns 1024-byte data as a response to each data request. The experiment lasted for 3 s.



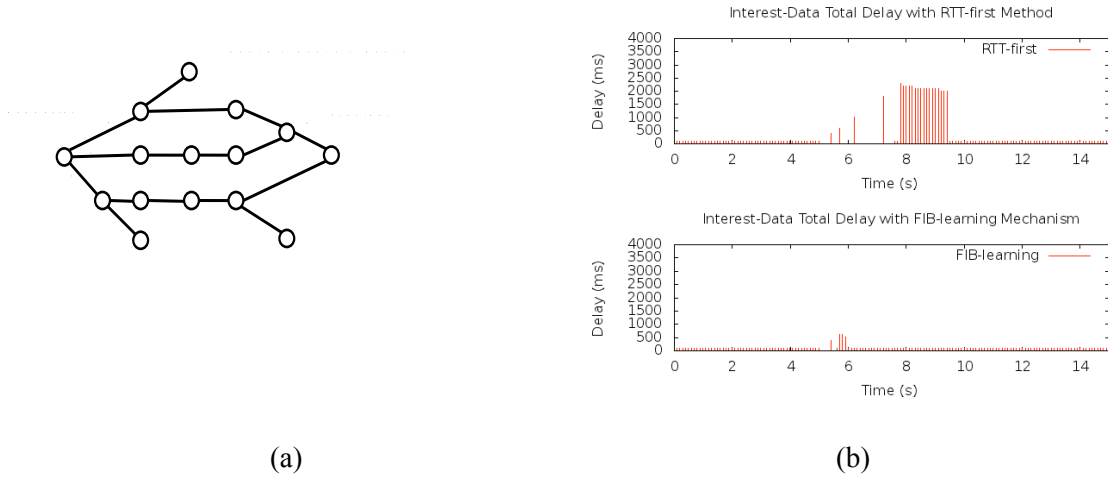
**Figure 3. (a) Topology. (b) Ratio of forwarded Interest packets to total flooded packets vs. Time.**

Figure 3 (b) shows the results. The vertical axis represents the ratio of the number of forwarded Interest packets to the total number of flooding Interest packets. The proposed FIB learning mechanism forwards a smaller number of Interest packets compared to the simple flooding method. The simple flooding method forwards an incoming Interest packet to all available connected faces (thin green arrows). The Provider returns the required Data packet following the reverse path (red arrows). The FIB learning mechanism gradually learns the most

efficient route (thick green arrows) from the initial flooding strategy and reaches a stable advantageous space occupancy in a short time, i.e., approximately 700 ms.

### 3.2. Link Failure

We compared the FIB learning mechanism with the RTT-first method [14] when a link failure occurred in the network topology during the data transfer as shown in Figure 4 (a). The RTT-first algorithm always selects the single face that shows the shortest Round Trip Time (RTT) among available outgoing faces. If a link failure occurs on the RTT-first path, the data consumer will select the face with the second shortest RTT value as the Interest forwarding face. The new FIB learning mechanism reverts to the initial flooding strategy [a] if the data is not received within the timeout period. Failure to receive data prior to timeout indicates a link failure. The FIB mechanism detects a link failure and attempts to learn a new working face among connected faces. In this experiment, the timeout period was set to 200 ms.

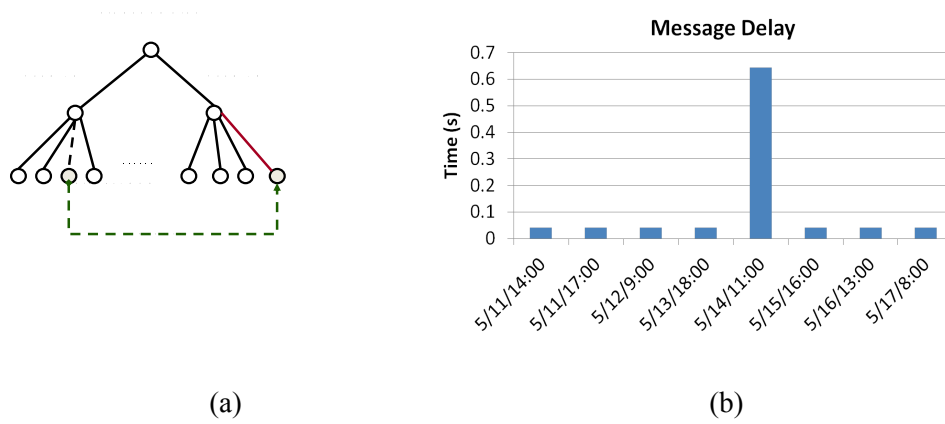


**Figure 4. (a)** Topology. **(b)** Delay between first Interest sent and Data received.

We assume that the data consumer continuously issues 10 Interests per second for 10 s. The data Provider returns the Data with a 1024-byte payload for each Interest. Suppose that a link failure, denoted by X in Figure 4 (a), occurs 5 s after Interests are initiated. Figure 4 (b) shows that the new FIB learning mechanism outperforms the RTT-first method; the FIB learning mechanism recovers more quickly from a link failure in such a scenario. It requires less time to learn an alternative working face for forwarding the succeeding Interests. Total delay refers to the time between the first Interest sent and Data received, including Interest retransmission time.

### 3.3. Application

A practical application of the FIB learning mechanism is exemplified. Student Unions (SUs) from two friendly universities hold an annual football game. Each player belongs to one team (Team\_A or Team\_B). Each player has a unique registered number, e.g., /Football\_Game/Player\_No. Each SU can retrieve a variety of statistical information for each player. For example, the number of goals by /Football\_Game/Player\_011/Goals, the number of assists by /Football\_Game/Player\_129/Assists, and fouls committed by /Football\_Game/Player\_219/Fouls. Assume that on May 14, Player\_019 transfers from Team\_A to Team\_B as illustrated in Figure 5 (a). We observe how the new FIB learning mechanism works to update FIB behavior from the current learned configuration status.



**Figure 5. (a) Topology. (b) Request-data Message Delay.**

Figure 5 (b) shows the simulation results. It is observed that the requesting SU fails to retrieve information about Player\_019 because FIB learning occurs before May 14, which is the data when Player\_019 transferred from Team\_A to Team\_B. This assumes an old topology. However the SU learns the transfer in a short time from the current configuration and can easily find the new working route to obtain data about Player\_019 within 0.7 s. There is a larger delay on May 14 because the SU goes back to the initial flooding for finding a new working path in order to retrieve data from Player\_019. The Interest requests after May 14 are successful within 0.1 s because they benefit from the newly learned FIB entries.

This simple application of the proposed FIB learning mechanism demonstrates that CCN architecture applies to data collection and that the learning mechanism works smoothly to adjust a forwarding strategy when a mobile node, such as Player\_019, is detected. This simulation also demonstrates the effectiveness of the FIB learning mechanism to deal with mobility, which is significant because mobile devices like smart phones and tablet PCs are becoming increasingly popular.

#### 4. Discussion

Forwarding strategy is essential to CCN architecture. This paper proposes a learning forwarding mechanism that works reasonably well in handling link failure and node mobility. However, many possibilities exist in forwarding strategy designs to suite different network environment and problems. For example, concerns may arise about the overhead at the initial flooding stage. There are alternatives: selecting a single face or flooding an Interest to all faces. There is a tradeoff between the overhead and delay time to retrieve data. Flooding explores all possible paths at once to discover a working path quickly. This overhead produced by the initial flooding is acceptable compared with the overall data traffic, especially in small or medium scale networks. For a large scale network, we should investigate the possibility of combining simple flooding mechanism and an appropriate routing algorithm. We have a plan to evaluate the performance together with overhead of the combinations.

#### 5. Conclusion

Unlike traditional IP-based networks, CCN's communication model, which retrieves data by name, leads to a different forwarding plan that requires intelligent use of a FIB at each router. This paper proposed a new FIB learning mechanism to utilize forwarding states to learn the successful outgoing faces for future Interest forwarding. With this learning mechanism, the FIB table entries are learned automatically from successful Data retrieval. The Interest forwarding is optimized owing to self-adaptive path discovery for the required Data or Content.

We evaluated the data delivery performance of the FIB learning mechanism under adverse conditions such as link failure. The simulation results show that our FIB learning mechanism works efficiently to reduce redundant traffic and provides good performance when handling link failures. Finally, in light of today's enormously increasing number of mobile end-user devices, such as smart phones and tablet PCs, mobility becomes a major concern. Thus, an application example was provided to demonstrate the FIB learning mechanism's resilience when handling mobile nodes in CCN.

#### References

1. Cisco Systems, Inc., Entering the zettabyte era, white paper, June 2011.
2. G. Pallis; A. Vakali. Insight and Perspectives for Content Delivery Networks. *Communications of the ACM*, 49: 101-106, January 2006.
3. V. Jacobson; D. K. Smetters; J. D. Thornton; M. F. Plass, N. H. Briggs; R. L. Braynard. Networking named content. *CoNEXT'09*, pages 1-12, New York, NY, USA, 2009. ACM.
4. L. Zhang; D. Estrin; V. Jacobson; B. Zhang. "Named data networking project" in Technical Report, NDN-0001, 2010.



5. The size of content naming space. <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html/>.
6. D. Perino; M. Varvello. A reality check for content centric networking, *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, 2011, pp. 44-49.
7. I. Psaras et al. Modeling and Evaluation of CCN-Caching Trees. *Networking 2011*, Lecture Notes in Computer Science, vol. 6640, Springer Berlin/Heidelberg, 2011, pp. 78-91.
8. G. Carofiglio et al. Modeling Data Transfer in Content Centric Networking, *Proc. 23rd, Int'l. Teletraffic Congress (ITC)*, 2011.
9. B. Ahlgren et al. A survey of information-centric networking. *Communications Magazine*, IEEE, vol. 50, Issue 7, July 2012, pp. 26-36.
10. PURSUIT Publish-Subscribe Internet Technology, <http://www.fp7-pursuit.eu/PursuitWeb/>.
11. Project ccnx: <http://www.ccnx.org/>.
12. OSPF Version 2, <http://tools.ietf.org/html/rfc2328/>.
13. NS-3 Network Simulator, <http://www.nsnam.org/>.
14. A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM: NDN simulator for NS-3, Tech. Rep. NDN-0005, NDN Project (July 2012).

© 2013 by the authors; licensee Asia Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).