Cooperative Routing Protocol for Content-Centric Networking

Saran Tarnoi†, Kalika Suksomboon‡, Wuttipong Kumwilaisak§, Yusheng Ji†‡
† Department of Informatics, The Graduate University For Advanced Studies, Tokyo, Japan
‡ National Institute of Informatics, Tokyo, Japan
§ King Mongkut’s University of Technology Thonburi, Bangkok, Thailand
Email: {saran, kalika, kei}@nii.ac.jp, wuttipong.kum@kmutt.ac.th

Abstract—A typical Forwarding Information Based (FIB) construction in the Content Centric Networking (CCN) architecture relies on the name prefix dissemination following the shortest path manner. However, routing based on the shortest path may not fully exploit the benefits of forwarding and data planes of the CCN architecture since different content requester routers may use disjoint paths to forward their interest packets, even though these packets aim at the same content. To exploit this opportunity, we propose a cooperative routing protocol for CCN, which focuses on a FIB reconstruction based on the content retrieval statistics to improve the in-network caching utilization. A binary linear optimization problem is formulated for calculating the optimal path for the cooperative routing. The simulation results show an improvement in the server load and round-trip time provided by the cooperative routing scheme compared with that of the conventional shortest path routing scheme.

Index Terms—Content-centric networking, future Internet architecture, cooperative routing protocol, content popularity.

I. INTRODUCTION

The way we use the Internet has been changing since the date it was built. The majority of traffic on the Internet comes from content retrieval, instead of end-host communications. Various designs of future Internet architectures have been proposed. One of the promising architectures is the Content-Centric Networking (CCN) architecture [1].

CCN uses prefix names to identify the contents and to route packets. The prefix name can be hierarchically constructed based on the URI Representation [1], [2], [3]. There are three tables in a CCN router, i.e., Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). The FIB of a CCN router partly resembles the routing table of IP routers whose destination field is changed into the prefix of the content names. The routing in CCN is receiver-driven, which is controlled by the forwarding of interest packets. An interest packet contains a prefix name of the content. When an interest packet arrives at a CCN router, it searches for the desired content that may be stored in the CS. If there is a matching content, a data packet is created and sent back along the reverse path of the relevant interest packet. Otherwise, the CCN router checks its PIT whether any interest packet asking for the same content has been sent. If so, the arrival face of the interest packet is added to the existing PIT entry. Otherwise, a new PIT entry is created and the FIB is consulted to determine where to forward the interest packet. When the forwarded interest packet meets its desired content, the relevant data packet follows the reverse path to the requester router based on the recoded arrival faces in the PITs. The content in a data packet is stored in the CS of each CCN router it traverses depending on the caching decision and cache replacement policies. The detailed description of CCN architecture can be found in [1].

The FIB is typically built for forwarding interest packets in the shortest path manner [2], [3]. However, using the shortest paths may not fully exploit the benefits of the caching ability. Different CCN routers may independently use disjoint paths to forward the interest packets, even though they are asking for the same content. Each CCN router accordingly benefits from the cached content only by its generated traffic. In contrast, the different CCN routers can cooperatively use a common path, so the cached content is not only useful for a given CCN router but all the participants. Moreover, the CCN routers may frequently generate interest packets that ask for some content originated from the same originator. In practice, this scenario can be illustrated by considering a group of people who access the same website. Once the popular content is viewed by someone, it is often viewed again by other people who access the website rather than the people who have never known its existence. This scenario occurs to YouTube [4], one of the largest content distributors on the Internet.

Various routing architectures for CCN have been proposed [5], [6], [7], but none of them considered a popularity-based routing. A popularity-based routing for CCN was recently proposed in [8]. The scheme conducts load balancing based on the popularity of the content retrieval and implements different cache replacement policies for different content objects. Even though the approach improves the cache utilization, it required a massive system resource to observe the popularity of all content objects. Another work that used the popularity of content retrieval in the routing is proposed by Dai et al. [3]. The scheme firstly lets each router flood the interest packets to all available faces. After the content popularity has been learned, the provider router then floods the prefix announcement messages to construct the FIBs. This method can effectively solve the FIB explosion problem. However, the scheme did not consider the caching functionality in the CCN architecture.

We realize this challenge and set a goal to improve the cache utilization by introducing a cooperative routing to CCN. The
major contributions of this work are:

- We propose a cooperative routing protocol for CCN that focuses on a FIB reconstruction that is based on the content retrieval statistics.
- We formulate a binary linear optimization problem used for calculating the optimal path for the cooperative routing.
- We investigate the performance of the cooperative routing for CCN from the aspects of the server load and round-trip time, when it is used with various implementations of the cache replacement for CSs.

The rest of this paper is organized as follows. Section II describes our design of the cooperative routing protocol for CCN. Section III introduces the optimization problem used for calculating the optimal path for the cooperative routing. We evaluate the proposed scheme in Section IV, and our concluding remarks are in Section V.

II. COOPERATIVE ROUTING PROTOCOL FOR CCN

The objective of the cooperative routing protocol is to selectively aggregate the multiple flows of interest packets onto the same path to improve the cache utilization. Two logical types of the CCN routers are defined as follows.

Definition 1. A requester router is a CCN router that generates or receives interest packets from the other CCN routers. The requester router forwards these interest packets to the next-hop CCN routers if the relevant content is not found in its CS.

Definition 2. A provider router is a CCN router that originates some name prefixes. The provider router is connected to the server that generates the relevant content.

A CCN router can collect the prefix citation statistics in its FIB. When the interest packets asking for the similar content\(^1\) are frequently sent out from a requester router, the requester router sends a notification message to the relevant provider router to subscribe to a prefix group. A prefix group consists of a content provider router, which originates the prefix, and a number of subscribed requester routers. A subscribed router frequently forwards the interest packets containing the associated prefixes. The provider router examines the request message, and then calculates a compromising path that partly resembles a shortest-path tree. The objective of doing this is to aggregate the multiple flows of the interest packets that require the similar content. Using the compromising path should provide the following advantages.

1) Cache-Hit Rate: The requester routers send the interest packets asking for the content objects from the same originator using the same path. The possibility that a content object in the CSs on that path consumes an interest packet increases, and thus enhances the cache-hit rate. In other words, the cache utilization is improved.

2) Total Network Traffic: According to the forwarding process of CCN, the state of an unsatisfied interest packet is inserted in the PIT for the routing of the relevant data packet. If the PIT has already an entry for the prefix, the ingress face used by the interest packet is added to the existing entry. Therefore, one interest packet sent out from the CCN router can be responsible for many incoming interest packets. Putting similar interest flows on the same path increases the number of mentioned events, and thus saves the bandwidth used by the routers for repeatedly sending the same data.

3) Server Load: The number of data packets served by the provider router can be reduced by improving the CS and PIT utilizations.

4) Content Retrieval Time: An interest packet can meet its desired content in a closer CS more frequently as a result of the improved cache utilization.

Aggregating traffic onto the same path may lead to undesired network utilization in IP networks. However, the CCN design removes some of packet transmission redundancy by storing data in the CS while supporting multicast transmission using the redesigning forwarding process [1], [9]. We are aware of this interesting property and formulate a binary linear optimization problem for calculating the compromising path, which will be discussed in Section III.

The FIB entries of the CCN routers that are involved in the compromising path are reconstructed by adopting the update messages orderly sent from the provider router. A CCN router can simultaneously be both the requester and provider routers of different prefix groups.

III. COMPROMISING PATH CALCULATION

There are one provider router and a number of requester routers in each prefix group. The provider router calculates the compromising path for its prefix group. The compromising path calculation of a prefix group is independent of other prefix groups in the network. We formulate a binary linear optimization problem for this purpose. The formulated problem is intended to consume a lightweight computation and to provide a resilient design for future modification.

A. Network Model

A network is modeled as a directed graph \(G(N,E)\), where \(N\) and \(E\) are sets of CCN routers and links in the network, respectively. Let \(s\) and \(D\) be a content provider router and a set of requester routers, respectively. For link \(l \in E\), let \(t(l)\) and \(r(l)\) be the transmitter and receiver routers of link \(l\), respectively. For router \(n \in N\), let \(T_0(n) = \{l \in E|n = t(l)\}\) and \(T_1(n) = \{l \in E|n = r(l)\}\) be the sets of outgoing and incoming links of router \(n\), respectively. Each link \(l\) has capacity \(c_l\).

B. Optimal Compromising Path Calculation

A binary linear optimization problem is formulated for the compromising path calculation. The objective is to minimize

\(^1\)A similar content is defined by the different content objects that are originated from the same originator, therefore their prefixes are similar and share some components. For instance, “com/youtube/sports/au5165a”, “com/youtube/news/dc46a534”, and “com/youtube/sports/sh4165nd” are the prefixes of the similar content that share “com/youtube/”. 


the cost of the compromising path while guaranteeing accesses to the contents of all subscribed requester routers in the prefix group. The objective function and associated constraints are described as follows.

1) Cost Function: Our objective is to minimize the cost used by the compromising path. If \( t_l = 1 \), link \( l \) is used in the compromising path. Otherwise, \( t_l = 0 \). The cost of using link \( l \) is denoted by \( p_l \), which can be a function of link capacity \( c_l \). Based on the defined variables, the cost function is defined as

\[
\sum_{l \in E} p_l t_l. \tag{1}
\]

Minimizing Eq.\,(1) results in minimalizing the cost of the selected compromising path. Note that the cost of using the compromising path is less than or equal to the sum of the cost used by each shortest path connecting the provider router to each requester router.

2) Access to Content Constraint: Define \( x_d \) as a variable, where \( x_d = 1 \) indicates that requester router \( d \) has access to the provider router. Otherwise, \( x_d = 0 \). The number of the requester routers in the prefix group is equal to \(|D|\). The access to content constraint is formulated to guarantee that all requester routers have accesses to the provider router, which is defined by

\[
\sum_{d \in D} x_d = |D|. \tag{2}
\]

3) Path Share Constraint: The different requester routers in the prefix group are encouraged to share the same path. If \( f^d_l = 1 \), link \( l \) is used in the path connecting the provider router to requester router \( d \). Otherwise, \( f^d_l = 0 \). The constraint can be written as

\[
f^d_l \leq t_l, \forall d \in D. \tag{3}
\]

4) Path Length Constraint: We formulate this constraint to ensure that the path length from each requester router to the provider router is not longer than that when using the shortest path. It is written as

\[
\sum_{l \in E} f^d_l \leq h^d_s, \forall d \in D, \tag{4}
\]

where \( h^d_s \) denotes the shortest path length measured from provider router \( s \) to requester router \( d \).

The constraint narrows down the feasible solutions of the compromising path. To alleviate the problem and allow more routes to become the candidates of a compromising path, Eq.\,(4) can be rewritten in a more general form as

\[
\sum_{l \in E} f^d_l \leq h^d_s + k, \forall d \in D, \tag{5}
\]

where \( k \in \mathbb{N}^+ \), where \( \mathbb{N}^+ \) is a set of non-negative integer numbers.

5) Problem Formulation: To compute the optimal compromising path, a binary linear optimization is established as follows.

Minimize

\[
\sum_{l \in E} p_l t_l \tag{6}
\]

Subject to

\[
\sum_{l \in T_{T_i(n)}} f^d_l - \sum_{l \in T_{T_i(n)}} f^d_l = \begin{cases} x_d, & n = s \\
-x_d, & n = d \\
0, & \text{otherwise} \end{cases}, \tag{7}
\]

\[
\forall d \in D, \forall n \in N, \sum_{d \in D} x_d = |D|, \tag{8}
\]

\[
f^d_l \leq t_l, \forall d \in D, \forall l \in E, \tag{9}
\]

\[
\sum_{l \in E} f^d_l \leq h^d_s + k, \forall d \in D, \tag{10}
\]

\[
f^d_l t_l, x_d \in \{0, 1\}, \forall d \in D, \forall l \in E. \tag{11}
\]

Constraint (7) is the flow conservation constraint that provides the connectivity of the path between the provider router to all requester routers in the prefix group. Constraint (8) guarantees that each registered requester router has access to the provider router. Constraint (9) encourages the different requester routers in the prefix group to share the same path. Constraint (10) is a path length constraint. However, using the more relaxed path length constraint may result in a longer delay than that of the legacy shortest path, even though some interest packets benefit from the cached contents. We, therefore, let \( k = 0 \) and leave this study for our future work. Constraint (11) controls the feasible values of \( f^d_l, t_l, \) and \( x_d \).

We use the GNU Linear Programming Kit (GLPK) package [10], which is a free software and offers various linear solver methods, to solve the proposed binary linear optimization problem. An optimal solution is a set of \( t_l = 1 \), for \( l \in E \), which represents the optimal compromising path for the prefix group.

IV. Simulation Results

Evaluation of the cooperative routing protocol for CCN is conducted through simulation to evaluate two aspects of the content retrieval of CCN: 1) the server load which is also called the server-hit rate in a network and 2) the round-trip time delay (RTT). We conduct the simulation on ndnSIM (NDN simulator for NS-3) [11].

Two routing schemes are conducted in the simulation, i.e., the cooperative routing protocol (CP) and the legacy shortest path routing protocol (SP) [2]. The data packets that traverse each CCN router are always stored in the CS of the router. To gain a better understanding of the behavior of the cooperative routing scheme compared to the shortest path routing of CCN, we perform three cache replacement configurations for the CSs, i.e., least frequently used (LFU), least recently used (LRU), and random replacement (RR). Therefore, we have six scenarios for our simulation.
A. Network and Traffic Models Used in the Experiments

We study the performance of the cooperative routing protocol when it is used in the network with mesh topology. Differently from other works which assumed a homogeneous content retrieval throughout the network, we model the interest traffic based on the geographic popularity [4].

The mesh topology used in our simulations is generated by using the igraph package [12]. Eight geographic random topologies containing 100 nodes are used in the simulation, where the average degree of each node is seven. Each node represents a CCN router whose CS can store 1,000 content objects. The size of all content objects are identical. We randomly select a number of nodes to represent the set of provider routers disseminating 30 prefixes. Each provider router generates 10,000 to 100,000 unique contents, so that there are 300,000 to 3,000,000 unique content objects in total. We randomly select a set of nodes to be the requester routers that subscribe to 30 prefix groups. Each prefix group contains one provider router and five to ten requester routers, which are randomly distributed in the mesh network. Each requester router generates random interest packets asking for the similar content corresponding to its subscribed prefix group(s) at 100 to 200 Hz per prefix group. Therefore, the total traffic of the interest packets is in the range of 15,000 to 60,000 Hz.

The bandwidth of each link is identical. The propagation delay of the transmission on each link is set to 10 milliseconds. Each link is assumed to be lossless. As a result, the cost of using each link is the same, so the optimal path selection depends on a number of hops. The Zipf’s distribution is used to model the content retrieval traffic at each requester router. The Zipf’s exponent parameter (α) is varied from 0.4 to 1.0, as used in [8]. The simulation time is 1000 seconds with a 400 second warm up period. The average results are reported with 95% confidence interval.

B. Experimental Results

The results for the mesh network are shown in Fig. 1. We measure the server load and RTT as functions of the cache replacement policies of CSs and α. The server load and RTT when using our cooperative routing are smaller than those when using the shortest path routing scheme. The gains can be obtained for all cache replacement policies of CSs and α values. These improvements are more obvious when α value increases. Although the simplest cache replacement configuration in the simulation, i.e., RR is deployed in CSs, our cooperative routing consistently achieves the server load reduction and RTT gains. Interestingly, the server load and RTT when using RR are better than those when using LRU for all α values. However, LUFU provides the best results compared with the other cache replacement policies. In addition, our cooperative routing scheme reduces the total network traffic but we cannot show the results here due to the limited space.

V. CONCLUSION

We propose a cooperative routing protocol for CCN, which focuses on an FIB reconstruction based on the content retrieval statistics to improve the cache utilization of CCN. A binary linear optimization is formulated to calculate the optimal path of the cooperative routing. We compare the performance of our proposed scheme to the shortest path routing scheme when they are used with different CS cache replacement configurations and different values of Zipf’s exponent parameter. By using the proposed cooperative routing scheme, the simulation results show an improvement in the server load and round-trip time compared with those of the shortest path routing scheme.

REFERENCES