Multi-Armed Bandit-based Routing Method for In-network Caching

Gen Tabei*, Yusuke Ito†, Tomotaka Kimura‡, and Kouji Hirata§
* Graduate School of Science and Engineering, Kansai University, Osaka, Japan
E-mail: k190688@kansai-u.ac.jp
† Faculty of Engineering, Tokyo University of Science, Tokyo, Japan
E-mail: y-ito@ee.kagu.tus.ac.jp
‡ Faculty of Science and Engineering, Doshisha University, Kyoto, Japan
E-mail: tomkimur@mail.doshisha.ac.jp
§ Faculty of Engineering Science, Kansai University, Osaka, Japan
E-mail: hirata@kansai-u.ac.jp

***Abstract*—This paper proposes a routing method based on the multi-armed bandit algorithm for in-network caching. In-network caching is a technology that enables intermediate routers to store contents in their cache. Thus, users can download the contents from the routers in addition to original content servers. This feature can save network resources and reduce response time. The proposed method applies multi-armed bandit algorithm, which is one of reinforcement learning, to route selection of the routers, where we consider the routers as players in a multi-armed bandit problem. In the proposed method, each router forwards content requests to appropriate output ports based on rewards calculated by the multi-armed bandit algorithm. Through simulation experiments, we show that the cache hit rate is improved by about 10% while suppressing the increase in the average hop counts, compared with a shortest path algorithm.**

I. Introduction

The Internet has been used as a communication infrastructure that enables end-to-end information transmission such as e-mail and file transfer. In recent years, however, various kinds of applications such as high-quality video distribution, Internet of Things, and social networking services rapidly developed, and their traffic has become dominant of the Internet traffic. In this way, services and environments required by people are constantly changing. The number of devices connected to networks and data traffic are expected to increase further in the future.

To accommodate the increasing traffic, in-network caching has been considered [1]. In-network caching enables routers in a network to store contents passing through them in their cache. Users can download the contents from the routers in addition to original content servers when requesting the contents. As a result, high responsivity and adaptability can be achieved.

In general, when a user requests a content, the shortest path from the user to the corresponding original content server is used as a routing path. In this case, cached contents are not fully utilized because the probability that requested contents are cached on the path is low. In order to utilize the caching mechanism, some routing methods have been proposed in the past [2], [3]. In this paper, we propose a multi-armed bandit-based routing method. The multi-armed bandit problem [4] is one of problems in reinforcement learning. It aims to maximize the reward by repeating exploration and exploitation for multiple target objects. The proposed method applies multi-armed bandit algorithms to routing for in-network caching. In the proposed method, we consider each router as a player in the multi-armed bandit problem and the router learns the optimal routing path to each content. When receiving a content request, each router forwards the request to an appropriate output port that is likely to be connected to a router having the requested contents, based on the reward of each output port. By doing so, the proposed method is expected to enhance the cache hit rate and reduce the number of hops to download contents. In this paper, we show the performance of the proposed method through simulation experiments.

II. Multi-armed bandit problem

As shown in Fig. 1, the multi-armed bandit problem models the behavior of a person playing multiple slot machines [5]. The player chooses one slot machine per trial, and then gets a reward by pulling the arm of the slot machine. For each slot machine, the reward follows a certain probability distribution, which is not known by the player. Therefore, the player needs to estimate the probability distribution from trial results to maximize the reward obtained from the multiple candidates. Therefore, in the multi-armed bandit algorithm, the player performs exploration and exploitation [6]. Exploration is the operation that selects an arm in some way and evaluates its value based on the resulting reward. Exploitation is the operation that selects the arm with the highest evaluation value. For example, in Fig. 1, the player selects slot machine 1 when performing exploitation. The probability distributions of the slot machines are estimated by iteratively updating their evaluation values based on the rewards obtained from exploration and exploitation. In the following, we describe algorithms for exploration and exploitation that are commonly used in multi-armed bandit problems.

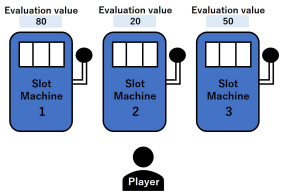

Fig. 1. Multi-armed bandit problem.

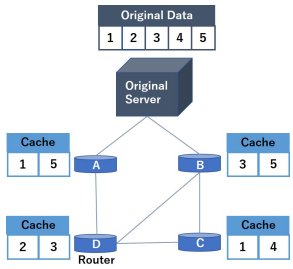

Fig. 2. System model.

- ϵ-greedy algorithm: The most straightforward multi-armed bandit algorithm is the ϵ-greedy algorithm [7]. In this algorithm, the player performs exploration with probability ϵ and exploitation with probability $1-\epsilon$ ($0 < \epsilon < 1$). In the case of exploitation, the arm with the highest evaluation value is chosen. In the case of exploration, one of the other arms is randomly chosen. In this paper, we set $\epsilon = 0.1$ because 0.1 is generally used for the multi-armed bandit problem.
- UCB1 algorithm: The UCB1 algorithm [8] defines a UCB score, which is an evaluation criterion combining the expected reward and its uncertainty of each arm. The player selects the arm with the highest UBC score. The UCB score for each arm i is given by

$$U_i = \hat{\mu}_i + \sqrt{\frac{2\log s}{n_i}}, \tag{1}$$

where $\hat{\mu}_i$ denotes the expected reward for arm i, s denotes the total number of choices for all arms, and n_i denotes the number of choices for arm i so far. In (1), the second term is a correction based on the confidence level of the expected reward for each arm as determined by the number of times that each arm is selected.

III. Proposed Method

The proposed method aims to enhance the cache hit rate by applying the multi-armed bandit problem to the routing problem for in-network caching-enabled networks. Specifically, the proposed method deals with each router in a network as a player in the multi-armed bandit problem. Each router considers its output ports as the arms and uses a multi-armed bandit algorithm such as the ϵ-greedy algorithm and the UCB1 algorithm to select appropriate output ports for incoming content requests.

A. System model

Fig. 2 represents the system model assumed in this paper. Let F denote a set of contents. There exists an original server that has all the contents in the network. For simplicity, we assume that the sizes of all the contents are the same. Let $\mathcal{R}$ denote a set of routers, each of which can store at most C copies of the contents in its cache. Let $\mathcal{N}_i$ denote a set of neighboring routers of router $i \in \mathcal{R}$ (i.e., a set of output ports). Each router has a table that holds the expected reward for each combination of contents and neighboring routers.

When requesting a content, a user sends a content request to the original server through the routers and basically obtains it from the original server. In the case where the requested content is stored in the cache of a router along the routing path, it can be retrieved from there and then is sent to the user. When the content is downloaded, each router along the downloading path to the user can replace it with their cached contents according to a cache replacement policy such as Least Frequently Used (LFU).

B. Routing method based on multi-armed bandit algorithms

In the proposed method, a user sends a content request to the nearest router when requesting a content as shown in Fig. 3. The router receiving the content request checks whether it has the corresponding copy of the content in its cache. If the router has the content copy (i.e., cache hit), it sends back the requested content to the users. If the router does not have the content copy (i.e., cache miss), it forwards the content request to one of neighboring nodes based on the multi-armed bandit algorithm discussed later. By repeating this procedure, the user can retrieve the requested contents. Note that in the case where there are no content copies in routers, this procedure increases the hop count of the content request. To overcome this problem, the proposed method forwards the content request to the original content server when the hop count becomes over a certain value T.

We explain this procedure with an example shown in Fig 3 where the user requests content A. Let us assume that T is set to 2 and router 4 has the copy of content A. First, the user sends a content request for content A to the nearest router (i.e., router 1). Router 1 receives the content request and checks its cache. However, router 1 does not have content A. Thus router 1 selects one of the neighboring routers based on the multi-armed bandit algorithm, and then sends the content request to the selected router. We here assume that router 3 is selected. When router 3 receives the content request, it checks its cache. Because the router 3 does not have content A, it forwards the content request to the next router based on the multi-armed bandit algorithm. We here consider two cases. In the case where router 3 forwards the content request to router 4, which has a copy of content A, router 4 sends back the copy to the user. On the other hand, in the case where router 3 forwards the content request to router 5, which does not have a copy of content A, router 5 sends the content request to the original server. This is because in this case, the hop count of the content request becomes 3 ($> T = 2$). The

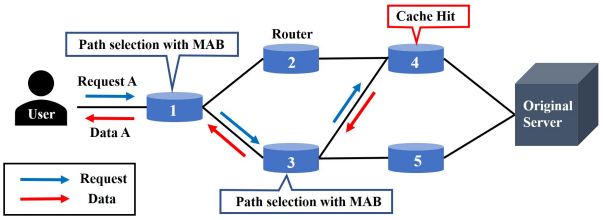

Fig. 3. Example of the proposed method.

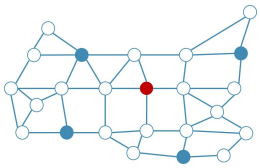

Fig. 4. Network topology.

original server sends the content to the user when receiving the content request.

C. Applying multi-armed bandit algorithms to the proposed method

In this paper, we apply the ϵ-greedy algorithm and the UBC1 algorithm to the proposed method. In the proposed method, when a router receives a content request, it selects an output port (a neighboring router) to forward the content request based on the expected reward of each output port, using the multi-armed bandit algorithms. Each router $i \in \mathcal{R}$ has the expected reward $V_i^{c,j}(t)$ for the combination of each content $c \in \mathcal{F}$ and each neighboring router $j \in \mathcal{N}_i$, where t denotes the number of times that router i has forwarded the request of content c to neighboring router j.

The cache hit rate and the number of hops required to retrieve requested contents are important factors of routing for in-network caching. Thus, in this paper, when a requested content is downloaded, each router i along the routing path updates the expected reward $V_i^{c,j}(t)$ for the combination of the content c and the selected neighboring router j as follows:

$$V_i^{c,j}(t+1) = \frac{N_i^{c,j}-1}{N_i^{c,j}} V_i^{c,j}(t) + \frac{1}{N_i^{c,j}} R_i^{c,j}(t+1), \quad (2)$$

where $N_i^{c,j}$ denotes the number of times that router i has forwarded requests for content c to neighboring router j so far. $R_i^{c,j}(t)$ denotes the reward that router i gets when the router selects neighboring router j to download content c. $R_i^{c,j}(t)$ is given by

$$R_i^{c,j}(t) = \begin{cases} 0.5 + 0.5/h_i, & \text{in the case of cache hit} \\ 0.5/h_i, & \text{otherwise,} \end{cases} \quad (3)$$

where h_i denotes the number of hops that it took from router i to find the requested content.

In the proposed method, each router selects a neighboring router based on the multi-armed bandit algorithm when receiving a content request. Specifically, if we use the ϵ-greedy algorithm, one router is randomly selected from non-optimal neighboring routers in the exploration phase. On the other hand, in the exploitation phase, the neighboring router with the highest expected reward is selected. If we use the UCB1 algorithm, the router selects the neighboring router with the highest UCB score $U_i^{c,j}$, which is given by

$$U_i^{c,j} = V_i^{c,j}(t) + \sqrt{\frac{\log N_{\text{total}}^{i,c}}{N_i^{c,j}}}, \quad (4)$$

where $N_{\text{total}}^{i,c}$ denotes the number of times that router i has forwarded requests for content c so far.

IV. Simulation Experiments

A. Model

In this paper, we conduct simulation experiments to evaluate the performance of the proposed method. We use the network shown in Fig. 4, which consists of 24 nodes (routers) and 43 bi-directional links. We assume that one original server (red circle in the figure) and four users (blue circle in the figure) are connected to different routers, which are randomly selected. The content requests are generated from the users according to Zipf's law [9]. Specifically, the request frequency of content c ($c = 1, 2, \ldots, |\mathcal{F}|$) is proportional to $f(c; \alpha, |\mathcal{F}|) = \frac{1}{c^\alpha} / \sum_{k=1}^{|\mathcal{F}|} \frac{1}{k^\alpha}$ where α is a bias parameter, which is set to 0.8 in this paper. The number $|\mathcal{F}|$ of contents is 1,000 and the sizes of all the contents are the same. The cache size of each router is 50 contents. When a content is downloaded, each router along the routing path replaces the downloaded content with one of cached contents. As the cache replacement policy, we use LFU. We collect 2,000 independent samples from the experiments.

B. Results

Fig. 5 shows the cache hit rate of the proposed method using the ϵ-greedy algorithm as a function of elapsed time, where $\epsilon = 0.1$. We assume that one content request is generated from a randomly selected user per one time slot. The cache hit rate at a time slot is defined by the number of samples where the cache hit has occurred at the time slot over the total number of samples. For the sake of comparison, we plot the result of a shortest path method where each router selects shortest paths to the original server in terms of the number of hops. As we can see from Fig. 5, the proposed method improves the cache hit rate by about 10% compared with the shortest path method. This result indicates that each router becomes able to select appropriate routes by repeating the trial.

Fig. 6 shows the average hop count of the proposed method using the ϵ-greedy algorithm as a function of elapsed time, where $\epsilon = 0.1$. The average hop count is defined by the sum of hop counts of the content requests in all the samples at the time slot over the total number of samples. As shown in Fig. 6, the average hop counts of the proposed method becomes small as the time elapses. It is slightly higher than that of the shortest path method.

Next, we examine the performance of the proposed method when using the UCB1 algorithm. Fig. 7 shows the cache hit

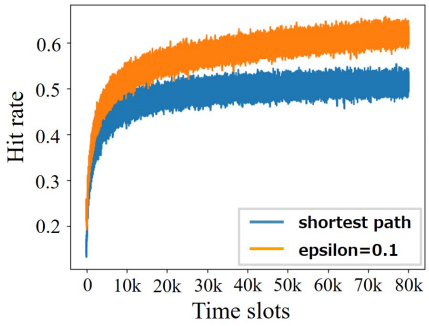

Fig. 5. Cache hit rate (ϵ-greedy).

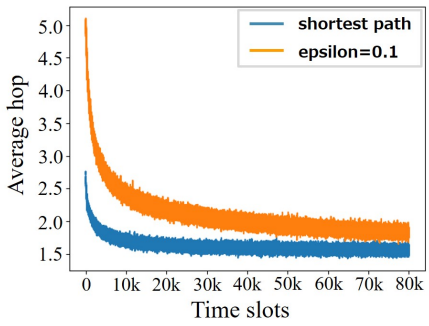

Fig. 6. Average hop counts (ϵ-greedy).

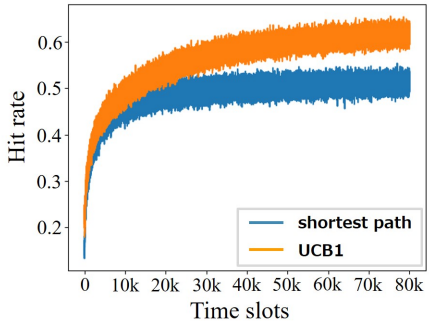

Fig. 7. Cache hit rate (UCB1).

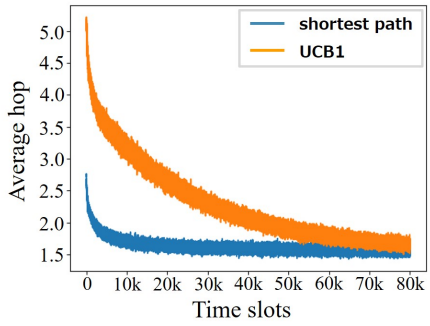

Fig. 8. Average hop counts (UCB1).

rate of the proposed method using the UCB1 algorithm as a function of elapsed time. From this figure, we observe that the cache hit rate of the proposed method is higher than that of the shortest path method, similar to the result in Fig. 5. Also, Fig. 8 shows the average hop counts of the proposed method using the UCB1 algorithm as a function of elapsed time. As we can see from this figure, the average hop counts of the proposed method becomes nearly equal to that of the shortest path method with time elapses.

V. Conclusions

In this paper, we proposed a multi-armed bandit-based routing method for in-network caching. In the proposed method, each router uses a multi-armed bandit algorithm to choose routing path. Through simulation experiments, we showed the performance of the proposed method using the ϵ-greedy algorithm and the UCB1 algorithm. As future work, we will examine cache replacement methods to enhance our routing method.

Acknowledgement This research was partially supported by Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant No. 18K11282.

References

[1] Y. An and X. Luo, "An In-Network Caching Scheme Based on Energy Efficiency for Content-Centric Networks," *IEEE Access*, vol. 6, pp. 20184–20194, 2018.

[2] L. Saino, I. Psaras and G Pavlou. "Hash-routing schemes for information centric networking," in *Proc. The 3rd ACM SIGCOMM workshop on Information-centric networking (ICN '13)*, New York, USA, Aug. 2013, pp. 2732.

[3] Y Xin, Y Li, W Wang, W Li and X Chen, "Content aware multi-path forwarding strategy in Information Centric Networking," in *Proc. The 21th IEEE Symposium on Computers and Communications (ISCC)*, Messina, Italy, Jun. 2016, pp. 816-823.

[4] A. Dubois, J. Dehos and F. Teytaud, "Improving Multi-modal Optimization Restart Strategy Through Multi-armed Bandit," in *Proc. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, Florida, USA, Dec. 2018, pp. 338–343.

[5] S. Habib, A. Beemer and J. Kliewer, "Learned Scheduling of LDPC Decoders Based on Multi-armed Bandits," in *Proc. 2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, California, USA, Jun. 2020, pp. 2789-2794.

[6] N. Gutowski, O. Camp, T. Amghar and F. Chhel, "Using Individual Accuracy to Create Context for Non-Contextual Multi-Armed Bandit Problems," in *Proc. 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, Danang, Vietnam, Mar. 2019, pp. 1-6.

[7] S. D. Maqbool, T. P. Imthias Ahamed and N. H. Malik, "Analysis of adaptability of Reinforcement Learning approach," in *Proc. 2011 IEEE 14th International Multitopic Conference*, Karachi, Pakistan, Dec. 2011, pp. 45–49.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, 2002.

[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM'99*, New York, USA, Mar. 1999, pp. 126–134.