

*Article*

## Adaptive Prioritized Probabilistic Caching Algorithm for Content Centric Networks

Warit Sirichotedumrong<sup>1</sup>, Wuttipong Kumwilaisak<sup>1\*</sup>, Saran Tarnoi<sup>2</sup>,  
and Nattanun Thatphitthukkul<sup>3</sup>

<sup>1</sup> King Mongkut's University of Technology Thonburi, Bangkok, Thailand

<sup>2</sup> National Institute of Informatics, Tokyo, Japan

<sup>3</sup> National Electronics and Computer Technology Center, Pathumthani, Thailand

\*E-mail: wuttipong.kum@kmutt.ac.th (Corresponding author)

**Abstract.** This paper presents an adaptive prioritized probabilistic caching algorithm (APP) for content centric networks (CCN). The objective of the new caching algorithm is to satisfy content requesters with both improving received data quality and maintaining overall network performance. APP allows CCN routers to cache data packets based on the caching probability which is prioritized and unequally handles incoming data packets according to data priorities. APP adjusts the caching probability based on cache events occurred at the CCN router, and the current caching probability is calculated from the previous caching probability. We evaluate APP performance via computer simulations and compare the performance of our caching algorithm with previous caching schemes. The performance evaluation metrics compose of the received data quality, cache-hit percentage, server load, and traffic load. The computer simulation results show that APP yields the better data quality to content requesters and nearly performs in-network caching as well as the previous probabilistic caching scheme.

**Keywords:** Caching algorithm, content centric networks (CCN), data priority, probabilistic caching.

ENGINEERING JOURNAL Volume 21 Issue 6

Received 28 August 2017

Accepted 16 October 2017

Published 31 October 2017

Online at <http://www.engj.org/>

DOI:10.4186/ej.2017.21.6.11

## 1. Introduction

Content-centric networking (CCN) was proposed to be a distinctive Future Internet Architecture (FIA) [1, 2], which is transformed from location-based communication to content-based communication. CCN transports data in a network by looking at “name” instead of content location. By this means, CCN allows routers, which are called “CCN router”, to have their own storage and forwarding contents based on “name” of the contents. There are two types of packets in CCN: 1.) interest packet; and 2.) data packet. The content requester sends interest packet to the nearby CCN routers to find the required content, and the data packets will be sent from the CCN routers or servers to satisfy the requester with the corresponding data packets. CCN routers consist of three main tables: 1.) Forwarding Information Base (FIB); 2.) Pending Interest Table (PIT); and 3.) Content Store (CS). CS performs the most important role in CCN to deliver the corresponding data to the requester with the cached contents. In reality, the CS capacity is limited and cannot store every data in the network. Hence, caching decision algorithm plays an important role for improving network performance and satisfying the requester with the best data quality.

Several caching schemes were proposed to be used in CCN [3-12]. Originally, CCN employs universal caching scheme (Always), which allows CS to cache every incoming data packet not existing in the CS. The popularity-based caching strategies were presented in [3] and [4], respectively. The probabilistic caching schemes (Prob) were proposed in [5-11] to randomly cache data packets based on probability. Adaptive probabilistic caching scheme (A-prob) [10] was introduced to adjust caching probability according to recent cache-hit and cache-miss event. Cache Replacement Strategies for Scalable Video Coding (SVC) [11] presented the replacement policy dealing with different layers of scalable video. By nature of data contents mainly multimedia, different portions of data possess different priorities from the compression process. However, the proposed algorithms in [3-11] did not consider various priorities and unequal importance of data packets while the prioritized probabilistic caching algorithm [12] cannot maintain the overall network performance yielded by Prob.

Therefore, we propose the adaptive prioritized probabilistic caching algorithm based on caching probability to solve the limitations of previous caching algorithms. Our proposed method focuses on both popularities of contents and various priorities of data packets. The proposed caching algorithm dynamically adjusts caching probability like A-prob. However, the priorities of data layers are taken into account for calculating caching probability. The more importance layer will be cached more than the less important ones. We evaluate the performance of our proposed caching algorithm by normalized information value reflecting the reproduced data quality, cache-hit ratio, server load, and traffic load.

The rest of paper is organized as follows. Section 2 describes the system model. Section 3 presents our proposed caching algorithm and shows how to calculate caching probability of adaptive prioritized probabilistic caching algorithm. Section 4 evaluates and discusses the performance of our algorithm by the computer simulations in both data quality and network performance. Finally, conclusion remarks are in Section 5.

## 2. System Model

### 2.1. Data Priorities

In multimedia transmission, data packets are normally unequally important. Some data packets are more important than the others. An obvious example is scalable video coding [13]. We define “layer” to represent data priorities of data packets. Since the data priorities have to be known by all network components, the  $i^{th}$  layer tag is attached to the data packets. For simplicity, we assume four layers in this paper: 1.) base layer; 2.) the first enhancement layer; 3.) the second enhancement layer; and 4.) the third enhancement layer. Base layer is the most important data packet and has the highest data priority value. To reconstruct the data, base layer must be ensured that it was sent successfully to the requester. Without base layer, all enhancement layers corresponding to the considering base layer are useless for reconstructing the whole data. The enhancement layers serve as the quality improvement data packets which will be reconstructed with the base layer. The levels of the importance of the data layer  $\gamma_i$  are quantified as an example in Table 1. The more number of priority value, the more important of data. This reflects the effect on reconstructed data quality at the requester. In this paper, we will stick on this table for computer simulations as a toy example.

However, the priority values can be varied from types of multimedia data as well as the compression algorithms. Therefore, the priority values can be adjusted to serve different media data files.

Table 1. The priority values of data layers.

<i>Layers</i>	<i>Base Layer</i> <i>(i=0)</i>	<i>First Enhancement Layer</i> <i>(i=1)</i>	<i>Second Enhancement Layer</i> <i>(i=2)</i>	<i>Third Enhancement Layer</i> <i>(i=3)</i>
$\gamma_i$	1.0	0.3	0.2	0.1

## 2.2. Content Request Model

We assume that there are  $N$  groups of content. Each group of content consists of  $K$  contents which are requested based on the request statistic of requesters called “*Zipf-Mandelbrot Distribution*” [14, 15]. Each content object is divided to  $m$  chunks, and each chunk consists of  $n$  layers. When a requester requests content, it will send the interest packets for every chunk and every layer of the content. For instance, requester A demands the content named “/video1”, “/video1” is divided into two chunks, and each chunk consists of two layers. Requester A will send the interest packets to ask for base layer and the first enhancement layer of both chunks to the connected CCN router as shown in Fig. 1. If the requested content is not available at the CCN router, it needs to request the content from the content server. It can be seen that the more CCN routers contacting with the content server, the more server load. Hence, CCN routers and caching algorithm play a very important role in reducing the server load.

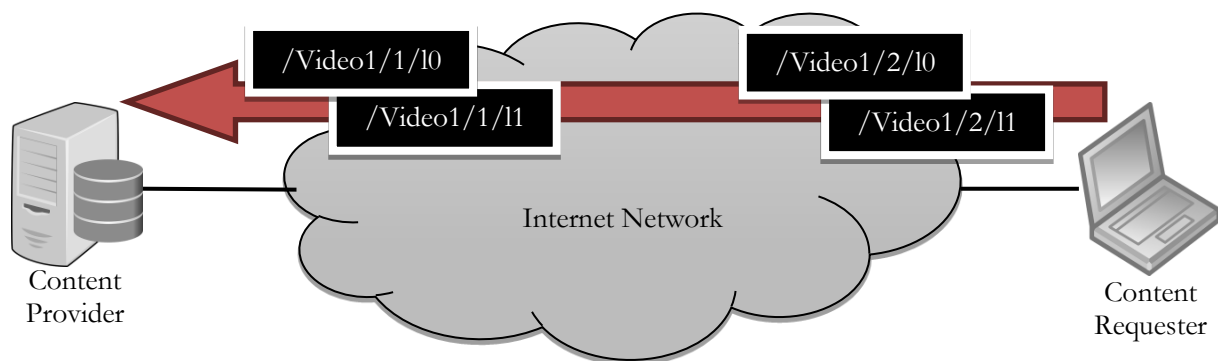


Fig. 1. Content requesting process.

## 2.3. Caching System Model

At CCN router, the caching system is utilized for controlling Content Store (CS). The caching system consists of a caching decision algorithm and a cache replacement policy. A caching decision algorithm decides which data packets have to be cached. When CS is full and would like to cache the new incoming data packets, CS will evict the previous cached contents based on the cache replacement policy.

We propose a new caching algorithm which caches data packets based on caching probability called “Adaptive Prioritized Probabilistic Caching Algorithm”. The proposed caching algorithm is used for calculating the caching probability according to the occurred events in the networks. Furthermore, we use Least-Recently Used (LRU) as a cache replacement policy in the caching system because LRU is the best cache replacement policy when it is deployed with probabilistic caching scheme [16]. When there is no room for the new incoming data packet, CS will evict the least-recently used data packet.

## 2.4. Network Model

There are two network models deployed in this paper. The first model is cascading network topology which is used for evaluating the overall performance of the caching algorithm without any effect of transmission schemes such as routing protocols. Cascading network topology consists of an original server, a requester,

and five CCN routers. As shown in Fig. 2, the CCN routers are connected in the middle of the original server and requester

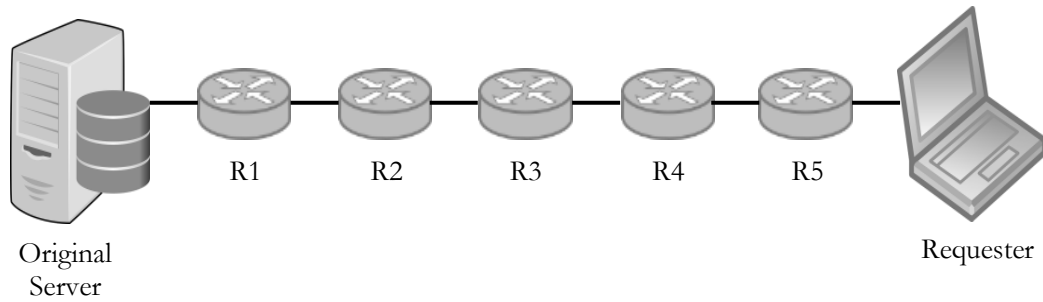


Fig. 2. Cascading network topology.

The other are random topologies which are generated using the *igraph* network analysis package [17]. We utilize random topologies to study the results and effects of the transmission schemes on the caching algorithms. The example of the random topology is shown in Fig. 3.

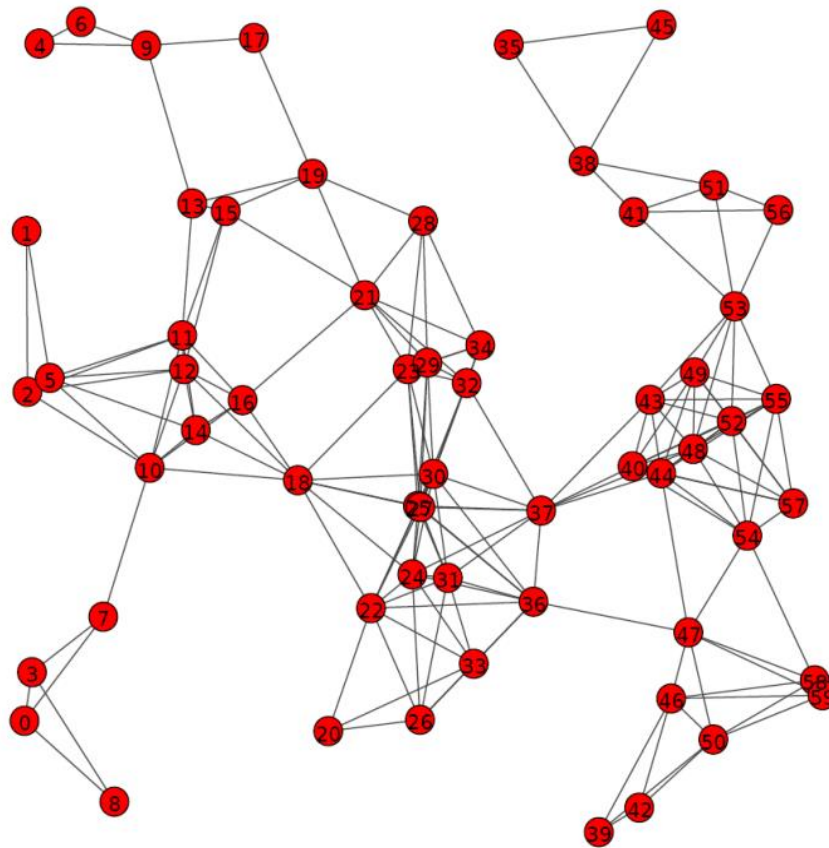


Fig. 3. Random topology.

### 3. Adaptive Prioritized Probabilistic Caching Algorithm

According to the Adaptive Probabilistic Caching Scheme (A-Prob) [11], the caching probability is calculated and adapted along cache-hit and cache-miss events. A-Prob did not pay an attention to the inequality of packet priorities. Built on top of A-Prob, Adaptive Prioritized Probabilistic Caching Algorithm (APP) brings packet priorities to the consideration of a caching scheme.

### 3.1. Design Objective

The performance of CCN including network performance and the reproduced data quality at the requesters heavily depends on network equipment specifications and resource management. The main objective of this paper is to improve CCN performance by modifying caching algorithm in CCN routers without upgrading network equipment.

Due to the limitation of CS capacity, the CS cannot store every data packet in the network, so caching decision algorithm performs the most important role in the CS. To fully utilize and deliver the best quality to requesters, CS should unequally handle incoming packets and cache more important data packets than the others. Not only improving the reproduced data quality, but network performance could be enhanced. Because of the uncertainty of content requesting behavior of requesters, we design the new adaptive prioritized probabilistic caching algorithm based on probabilistic caching scheme [5-10] and adaptive probabilistic caching scheme [11] which dynamically adapts itself to the network circumstances.

### 3.2. Caching Probability

In our proposed algorithm, when a CCN router receives an incoming data, it randomizes the real number in the range of [0,1] and compares the random number with the caching probability. If the random number is less than or equal to the caching probability, the data packet will be cached to the CS. We define *factors* and *events* as the types of calculation component. The *factors* consist of caching probability ( $P$ ), additional factor ( $c_a$ ), multiplication factor ( $c_m$ ), and priority values ( $\gamma$ ). In the caching probability calculation process, a CCN router adjusts the caching probability based on the *events* including cache-hit and cache-miss events. Cache-hit represents a rate that the cached data packets satisfying the incoming interest packets. CCN router will reduce the caching probability of the next incoming data packet, when a current cache hit is high. In contrast, if the cached data packets cannot satisfy the incoming interest packets, cache-miss event occurs. CCN router will cache the next incoming content with a high probability because the current contents in the CS tend not to be the ones that requesters need. As the data is prioritized into many layers, each layer will possess its own caching probability. If the data is divided into  $n$  layers, CCN router will calculate and store  $n$  caching probabilities for all layers. Equation (1) computes  $P_i$  at time  $t + \delta$  of the  $i^{th}$  layer of data packet, and  $\gamma_i$  is a factor for weighing the caching probability depended on a priority of the  $i^{th}$  layers.

$$P_i(t + \delta) = \begin{cases} 1 & , P_i(t) > 1 \\ P_i(t) \times (c_m \times \gamma_i) & , \text{If there is a hit at time } t, \\ P_i(t) + (c_a \times \gamma_i) & , \text{otherwise} \end{cases} \quad (1)$$

where  $c_m, c_a \in (0,1]$ .

As we can see from Eq. (1), the caching probability at time  $t + \delta$  is calculated from the previous state caching probability at time  $t$ . When a CCN router receives an interest packet, it will find the corresponding data packet in its CS and update the caching probability after the cache event occurs. When the cache-miss event occurs, the caching probability will be added by the product between the additional factor and the priority value of data layer. In contrast, when the cache-hit occurs, the caching probability is decreased by multiplying with the product of multiplication factor and priority value.

In conclusion, CS in every CCN router caches data packets based on the priority of the incoming data packet and decides whether it will be cached or not according to the caching probability of the incoming data packet layer. For example, if the incoming data packet at time  $t + \delta$  is base layer, the random real number will be compared and cached relied on caching probability  $P_0(t + \delta)$ .

## 4. Evaluation Metrics

### 4.1. Normalized Information Value

To evaluate the quality of reproduced data at the requesters, we define Normalized Information Value ( $V$ ).  $V$  is a mathematical expression for roughly estimating the quality of received data at a requester. It is defined as the proportion between amount of data packets received within transmission deadline ( $\tau$ ) and the total received data packets at the requesters. Generally speaking, base layer is the most important data packet and data cannot be reproduced at a requester without base layer. The enhancement layers are the data packets using for improving the reconstructed data quality at a requester. To successfully utilize the enhancement layer, it has to be reconstructed with the corresponding base layer and lower layer content chunks. Otherwise, it will be discarded. Four components using in normalized information value calculation are shown in Eq. (2).

$$N_m(t) = \begin{bmatrix} n_{0m}(t) \\ n_{1m}(t) \\ n_{2m}(t) \\ n_{3m}(t) \end{bmatrix}, Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \gamma = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}, \text{ and } K_m = \begin{bmatrix} k_{0m} \\ k_{1m} \\ k_{2m} \\ k_{3m} \end{bmatrix} \quad (2)$$

$N_m(t)$  is the column vector which refers to the accumulated number of useful data packets at time  $t$ .  $n_{im}(t)$  is the accumulated number of useful  $i^{\text{th}}$  layer received data packets.  $Q$  is a binary column vector for counting which data layer is useful. Every element of  $Q$  will be equal to 1 if every data layer is received within  $\tau$  seconds. If the round trip time of base layer data packet is longer than  $\tau$  seconds, every element of  $Q$  will be equal to 0. In case of only receiving base layer within  $\tau$  seconds,  $q_0$  is set to 1 while other elements are set to be equal to 0.  $\gamma$  is the priority column vector which is assigned the value according to Table. 1.  $K_m$  represents the total number of received data packets of  $m^{\text{th}}$  chunk data packets.  $k_{im}$  is the number of  $i^{\text{th}}$  layer received data packets.

$$N_m(t + \delta) = N_m(t) + Q. \quad (3)$$

The accumulated number of successfully reproduced data packets at time  $t + \delta$  is calculated from Eq. (3). As this paper assumes that the data packet is divided into four layers,  $l$  is equal to four.

After calculating the number of successfully reproduced data packets, it will be normalized by the total received data packets which are weighed by the priority matrix. The normalized information value can be written as Eq. (4).

$$V_q = \sum_{m=1}^D \left( \frac{\gamma^T \cdot N_m(t_s)}{\gamma^T \cdot K_m} \right). \quad (4)$$

From Eq. (4),  $V_q$  is the normalized information value of the  $q^{\text{th}}$  prefix group, and  $D$  represents the number of content chunks per prefix. Note that prefix group refers to the group of different data packets that have the associated prefixes [18, 19]. According to the end-to-end delay for real-time communication [20],  $\tau$  is set to 250 millisecond to ensure that the requester will perceive content with acceptable quality on time. The matrices of successfully reproduced data packets and total data packets are prioritized by multiplying with  $\gamma^T$ .  $t_s$  is the simulation time, so  $N_m(t_s)$  is the total number of useful data packets in the experiment.

## 4.2. Cache-hit Percentage

Cache-hit percentage is a caching performance evaluation metric which is equal to the ratio between the number of cache-hit events and the total interest packets arriving at the CCN router. High cache-hit percentage implies that the CCN router can satisfy a large portion of interest requests with the cached data packet in the CS instead of forwarding them to upstream nodes.

## 4.3. Server Load

Server load is a network performance evaluation metric which is equal to total requests arriving at the content provider in one second. If the caching system is efficient, the server load will be reduced because the content requester will be satisfied by retrieving contents from cached contents in CCN routers. High server load means that the server has to process a lot of requests, and its hardware would need to be upgraded to have a better performance to process many requests in one second.

## 4.4. Traffic Load

Traffic load is the total data packets transmitted in one second. High traffic load means that the links contain a lot of data packets, and if the traffic load exceeds the link bandwidth, the packet loss event will be occurred. Although our evaluation model assumes low packet loss event, this evaluation metric is used to determine the caching system performance. If CCN routers can satisfy the requester with the cached contents, CCN router will not forward the interest packet to other CCN routers, so the traffic load will be reduced.

## 5. Performance Evaluation

We evaluate the proposed caching algorithm using the computer simulation tool called ndnSIM [15, 16]. ndnSIM is an open source NS-3 based NDN simulator and composes of CCN architecture and CCN basic components. We conduct the simulation in two models: cascading network topology and random topologies. The bandwidth and propagation delay of each link are equal to ten Gbps and 25 ms, respectively. The cache capacity of each CCN router is equal to one percent and ten percent of data packet population. In the cascading topology, there is one prefix group named as “/prefix”. The cascading topology contains one prefix group, and there is 100 different contents which are sliced to five chunks, and each chunk consists of two layers.

Other than evaluating our proposed caching algorithm by the simple topology, this paper employs 30 random topologies to evaluate the proposed caching algorithm with various transmission schemes. Each topology consists of 60 nodes, and each topology contains three prefix groups. We randomly assign three nodes to be provider nodes and ten nodes to be requester nodes per each prefix group. In this simulation set-up, the transmission schemes used for the evaluation are as shown as follows:

- Shortest path routing protocol (SP) is the default transmission scheme of CCN [1, 21].
- Cooperative routing protocol (CP) is the routing protocol that focuses on an FIB reconstruction and selectively aggregates the multiple interest packets into the same path [18].
- Optimal cooperative routing protocol (OCRP) is selectively multiple flows of interests into the same path and calculates the optimal path using binary linear optimization [19].

The experimental notation description of different caching algorithms is described in Table 2.

Table 2. The experimental notation description.

Notation	Experiment Description
Prob	Probabilistic Caching Scheme ( $P = 0.001$ ) + LRU
A-prob	Adaptive Probabilistic Caching Scheme (0.001, 0.5) + LRU
APP <sub>1</sub>	Adaptive Prioritized Probabilistic Caching Algorithm ( $c_a = 0.001$ and $c_m = 0.5$ )+LRU
APP <sub>2</sub>	Adaptive Prioritized Probabilistic Caching Algorithm ( $c_a = 0.0001$ and $c_m = 0.5$ )+LRU
APP <sub>3</sub>	Adaptive Prioritized Probabilistic Caching Algorithm ( $c_a = 0.0001$ and $c_m = 0.7$ )+LRU
APP <sub>4</sub>	Adaptive Prioritized Probabilistic Caching Algorithm ( $c_a = 0.0001$ and $c_m = 0.8$ )+LRU
APP <sub>5</sub>	Adaptive Prioritized Probabilistic Caching Algorithm ( $c_a = 0.0001$ and $c_m = 0.9$ )+LRU

## 6. Results and Discussions

### 6.1. Cascading Topology

Table 3 shows the improvement of network performance evaluation metrics compared to Always when the cache size of each CCN router is equal to one percent and ten percent of data packet population. Figures 4 and 5 show the average cache-hit percentage of various caching algorithms at each CCN router when the CS capacity is the same as Table 3.

When cache size is equal to one percent of packet population, the Normalized Information Value improvement of APP5 is equal to 53.20 percent, while the Normalized Information Value gain of Prob is equal to 42.54 percent. APP2, APP3, and APP4 yield the received data better than Prob at the requester. For the caching performance metric, the cache-hit gain of Prob is equal to 5.44 percent, while the cache-hit gain of APP is in the range of 3.50 percent to 5.40 percent. Similarly, the server load and traffic load of APP2, APP3, APP4, and APP5 are not much different from Prob, so we can conclude that the load reduction rate between Prob and APP is not significantly different. When we consider the CCN router individual performance as shown in Fig. 4, Prob satisfies the requester better than the others for every router. Although the result indicates that the individual caching performance of APP is inferior to Prob, the margin is insignificant, showing that APP performs almost as well as Prob.

Table 3. The improvement gain for cascading topology (%).

Metric	Cache Size	Prob	A-prob	APP <sub>1</sub>	APP <sub>2</sub>	APP <sub>3</sub>	APP <sub>4</sub>	APP <sub>5</sub>
V	1%	42.54	40.97	43.41	49.35	50.90	52.40	<b>53.20</b>
	10%	55.56	55.99	62.80	63.19	64.80	66.90	<b>68.90</b>
Cache-hit percentage	1%	<b>5.44</b>	5.00	3.50	5.40	5.25	5.05	4.65
	10%	13.18	13.41	13.15	13.55	<b>13.70</b>	13.62	13.47
Server load	1%	<b>-15.21</b>	-13.71	-8.36	-15.13	-14.46	-13.75	-12.10
	10%	-60.92	-61.60	-60.87	-61.57	<b>-61.91</b>	-61.70	-61.52
Traffic load	1%	<b>-2.78</b>	-2.31	-1.13	-2.69	-2.52	-2.38	-2.05
	10%	-25.91	<b>-26.32</b>	-25.82	-26.09	-26.12	-26.06	-26.16



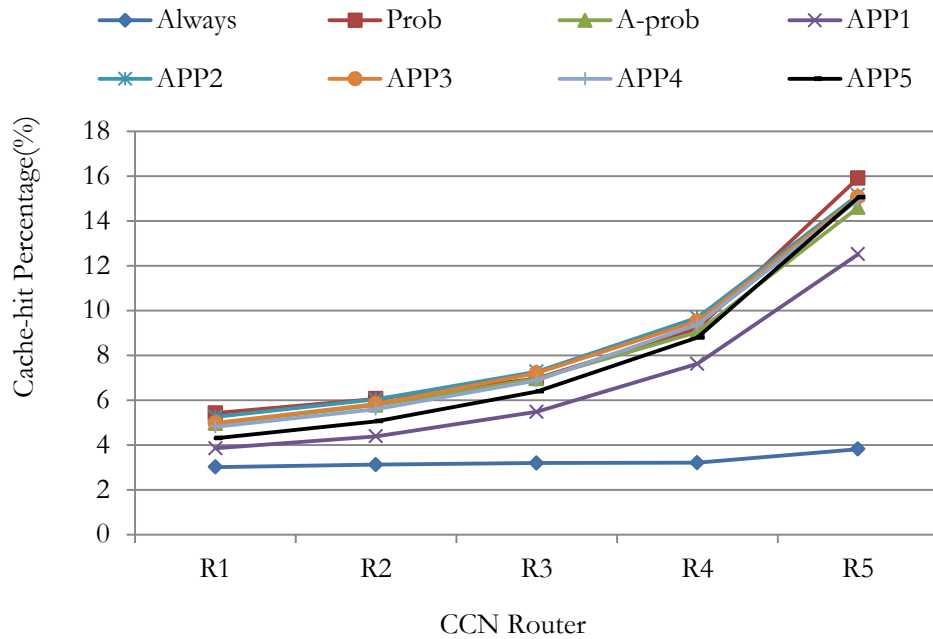


Fig. 4. Cache-hit percentage of each CCN router. (Cache size = 1% of content population).

When the cache size is increased to ten percent of packet population, APP significantly delivers a better Normalized Information Value than Prob for approximately 13 percent and improves the cache-hit percentage compared to Prob. Furthermore, the server load reduction of APP3 is equal to -61.91 percent, while APP2, APP3, APP4, and APP5 also reduce the traffic load compared to Prob. Although the improvement in terms of cache-hit percentage, server load, and network load is not very significant, our proposed caching algorithm offers better data quality to the requester compared to the other caching algorithms. When we consider the individual caching performance of each router, the proposed caching algorithm offers better cache-hit percentage. The caching probability of the proposed caching algorithm is gradually decreased and increased to give a better Normalized Information Value. The optimal value of additional and multiplication factor giving the best performance are equal to 0.0001 and 0.9, respectively.

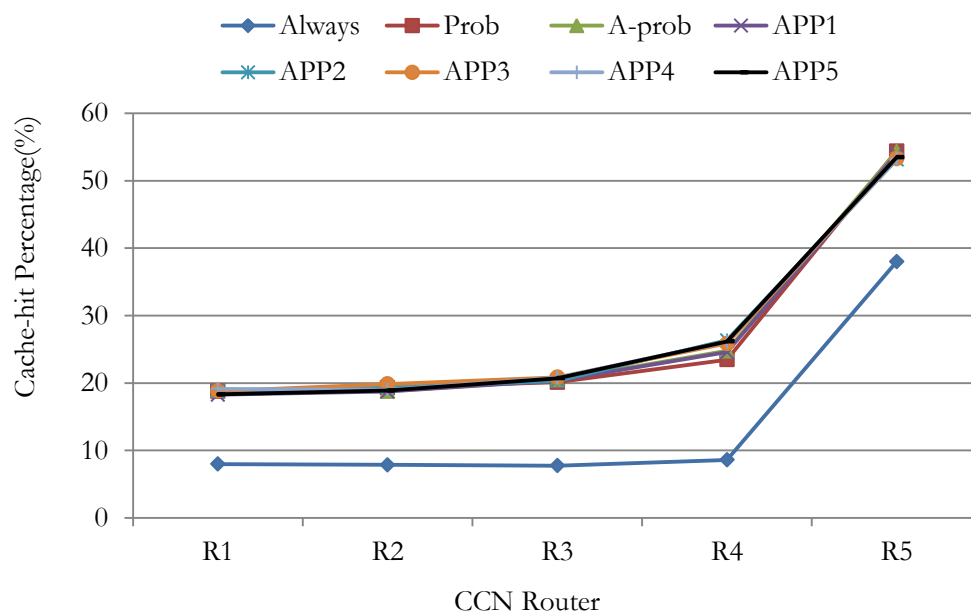


Fig. 5. Cache-hit percentage of each CCN router. (Cache size = 10% of content population).

## 6.2. Random topology

The improvement of random topology in various transmission schemes is shown in Table 4. In every transmission scheme, APP satisfies the requesters with a better reconstructed data quality than Prob, and APP5 offers the best reconstructed data quality to the requesters while Prob performs better caching performance and network performance.

When we operate the network with SP, the Normalized Information Value of APP5 and Prob are respectively equal to 18.84 and 13.86. Moreover, APP2, APP3, and APP4 offer better data reconstructed quality to the requesters. If we compare the results of APP5 in various transmission schemes, the results show that it yields the highest Normalized Information Value when it operates with OCRP because OCRP selectively aggregate the same prefix group interest packets in to the same path, so the data packets of the same prefix group are cached into the CCN router on the specific path. Furthermore, the possibility of successful data reconstruction of APP is better than other algorithms because APP tends to cache base layer more than enhancement layers.

Next, cache-hit percentage in Table 4 shows that Prob gives the best cache-hit percentage and instantly satisfies the requests with higher hit-rate than other caching algorithms. Cache-hit percentage gain of APP with OCRP varies in the range between 6.52 and 6.77, and the OCRP cache-hit gain is in the range of 6.64 and 6.80. Although the cache-hit of APP with CP are higher than OCRP, the margin is not significant. These results imply that the proposed algorithm is not affected by the transmission schemes because the proposed algorithm is individually operated based on the data priorities. The results in random topology environment gives the same tendency as in cascading topology environment.

In terms of server load and traffic load, Prob with SP reduces the loads more than APP, and the server load reduction of Prob varies between -17.80 and -16.23. The traffic load reduction percentage of Prob with SP, CP, and OCRP are equal to -10.39, -7.00, and -7.01 respectively. The result of network load reduction has the same trend with cache-hit percentage. In every transmission scheme, Prob reduces the server and traffic loads more than APP.

Table 4. The improvement gain of random topology in various transmission schemes (%).

Metric	Routing Protocol	Prob	APP <sub>2</sub>	APP <sub>3</sub>	APP <sub>4</sub>	APP <sub>5</sub>
V	SP	13.86	15.45	16.83	17.67	<b>18.84</b>
	CP	16.21	17.92	19.26	20.04	<b>21.47</b>
	OCRP	18.53	19.65	21.09	22.18	<b>23.57</b>
Cache-hit percentage	SP	<b>9.60</b>	6.62	6.65	6.65	6.45
	CP	<b>9.63</b>	6.79	6.80	6.84	6.64
	OCRP	<b>9.27</b>	6.77	6.70	6.69	6.52
Server load	SP	<b>-17.80</b>	-11.94	-11.78	-11.57	-10.79
	CP	<b>-16.23</b>	-11.26	-11.17	-10.92	-10.21
	OCRP	<b>-16.36</b>	-11.73	-11.51	-11.19	-10.47
Traffic load	SP	<b>-10.39</b>	-5.41	-5.54	-5.55	-5.47
	CP	<b>-7.00</b>	-3.02	-3.13	-3.22	-3.18
	OCRP	<b>-7.01</b>	-2.99	-3.06	-3.10	-3.07

## 7. Conclusion

We presented an adaptive prioritized probabilistic caching algorithm to satisfy the requester with the better reproduced data quality and maintain a good network performance to CCN. APP allows the CCN router to cache data packets based on the caching probability which unequally handles the incoming data packet according to its importance. APP adjusts the caching probability based on the *events* occurred at the CCN router, and the current caching probability is calculated from the previous caching probability. We evaluate APP performance via computer simulations and compare the performance of our caching algorithm with previous caching schemes. The performance evaluation metrics compose of the reproduced data quality, cache-hit percentage, server load, and traffic load. The computer simulation results showed that APP delivers higher reconstructed data quality at the requester. Moreover, the proposed scheme maintains the network performance comparing to the previous schemes.

## Acknowledgements

This research is supported in part by Thailand Graduate Institute of Science and Technology (TGIST) National Science and Technology Development Agency (NSTDA) with contract number TGIST 01-57-022 and Research Strengthening Project of the Faculty of Engineering, King Mongkut's University of Technology Thonburi.

## References

- [1] V. Jacobson D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *International Conference on emerging Networking Experiments and Technologies*, Rome, Italy, 2009, pp. 1-12.
- [2] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024-1049, Jul. 2013.
- [3] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *IEEE International Conference on Communications*, Budapest, Hungary, 2013, pp. 3619-3623.
- [4] T. Kawano, M. Shimamura, and H. Koga, "A selective caching scheme that adapts to content popularity changes in content-centric networking," in *11th International Conference on the Design of Reliable Communication Networks*, Kansas City, USA, 2015, pp. 83-84.
- [5] N. Laoutaris, H. Che, and I. Stavrakakis, "The lcd interconnection of lru caches and its analysis," *Perform. Eval.*, vol. 63, no. 7, pp. 609-634, Jul. 2006.
- [6] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache less for more in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758-770, Apr. 2013.
- [7] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition ACM SIGCOMM Workshop on Information-Centric Networking*, Helsinki, Finland, 2012, pp. 55-60.
- [8] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920-2931, Dec. 2013.
- [9] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi., and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *2012 IEEE Conference on Computer Communications Workshops*, Orlando, Florida, 2012, pp. 316-321.
- [10] S. Tarnoi, V. Suppakitpaisarn, and Y. Ji, "Adaptive probabilistic caching scheme for information-centric networking," IEICE Technical Report, 2015.
- [11] J. Lee, K. Lim, and C. Yoo, "Cache replacement strategies for scalable video streaming in CCN," in *19th Asia-Pacific Conference on Communications*, Bali, Indonesia, 2013, pp. 184-189.
- [12] W. Sirichotedumrong, W. Kumwilaisak, S. Tarnoi, and Nattanun Thatphithakkul, "Prioritized probabilistic caching algorithm in content centric networks," in *Proceedings of the 12th International Conference on Computing and Information Technology*, Khon Kaen, Thailand, 2016, pp. 255-265.
- [13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions in Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, Sept, 2007.
- [14] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, 2012
- [15] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," Technical Report on NDN, NDN-0028, 2015.
- [16] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for content-centric networks," in *39th Annual IEEE Conference on Local Computer Networks*, Edmonton, Canada, 2014, pp. 99-106.
- [17] *Igraph Library*. [Online]. Available: <http://igraph.org/> [Accessed: 1 August 2014].

- [18] S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Cooperative routing protocol for content-centric networking," in *38<sup>th</sup> Annual IEEE Conference on Local Computer Networks*, Sydney, Australia, 2013, pp. 716-719.
- [19] S. Tarnoi, W. Kumwilaisak, and Y. Ji, "Optimal cooperative routing protocol based on prefix popularity for content centric networking," in *39<sup>th</sup> Annual IEEE Conference on Local Computer Networks*, Edmonton, Canada, 2014, pp. 414-417.
- [20] R. M. Perea, "The media plane," in *Internet Multimedia Communications Using SIP*, 1st ed. San Francisco: Morgan-Kaufmann, 2008, ch. 10, sec. 2, pp. 203.
- [21] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF based routing protocol for named data networking," University of Memphis and University of Arizona, Tech. Rep., 2012.