International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **1**, February **2021**

ICIC International ©2021 ISSN 1349-4198
pp. **193–203**

OPTIMAL HETEROGENEOUS CACHE ALLOCATION MECHANISM IN INFORMATION-CENTRIC NETWORKING

Duo Jin[1] and Jianhui Lv[2]

[1]Information Center
Jilin Institute of Chemical Technology
No. 45, Chengde Street, Longtan District, Jilin 132022, P. R. China
jinduo2020_31@tom.com

[2]International Graduate School at Shenzhen
Tsinghua University
University Town of Shenzhen, Nanshan District, Shenzhen 518055, P. R. China
lvjianhui2012@163.com

Received March 2020; revised August 2020

Abstract. *We in this paper investigate the cache allocation problem of Information-Centric Networking (ICN), i.e., distribute the cache capacity across content routers under a constrained and fixed total cache budget, in which both topology information and traffic characteristics are considered as two factors to determine the importance of content router. Meanwhile, the evaluation of network topology depends on degree centrality, betweenness centrality and closeness centrality, while that of traffic characteristics depends on node load and interest preference. In particular, t-distributed stochastic neighbor embedding is used to reduce the dimensions of data. The simulation is driven by the real dataset over a real network topology, and the experimental results demonstrate that the proposed cache allocation mechanism is efficient by comparing with two baselines in terms of cache hit ratio, cache utilization rate and routing delay.*

Keywords: ICN, Heterogeneous deployment, Dimensionality reduction, Cache allocation

1. **Introduction.** At present, the inherent caching strategies in Information-Centric Networking (ICN) have some limitations, such as redundant content copies, low cache utilization rate, and unbalanced node load. Given this, a number of optimization schemes have been proposed [1], including for homogeneous scenario (i.e., all Content Routers (CRs) have the same cache size) and heterogeneous scenario (i.e., different CRs are likely to be allocated different cache sizes while the total cache capacity is fixed). Although some schemes devised for the homogeneous scenario can be also applied to the heterogeneous scenario, the obtained effect is usually indistinctive compared to those schemes specially devised for the heterogeneous scenario. In fact, the assumption that each node is deployed with the same cache capacity is unscientific in the real practical applications. On the one hand, not all CRs for a network topology have the same significance because their corresponding affairs are different; for example, the core CR that handles traffic is several times larger than the ordinary CR. On the other hand, it is very expensive and costly; for example, a CR with 10TB cache costs 300000 dollars, consuming 500W at the full work [2]. Therefore, it is required to pay more attention to the cache allocation under the heterogeneous scenario.

Certainly, the heterogeneous cache allocation has the considerable research value, especially when the (time or/and space) distribution of user requests changes greatly. To

DOI: 10.24507/ijicic.17.01.193

address this issue, the fundamental thought is summarized as that more important CR is allocated larger cache size in order to cache more items. However, it is very difficult to determine which nodes are important and which nodes are not that important because the term "important" is an abstract word. Regarding this, there are three different types of solutions, i.e., central nodes [3,4], edge nodes [5] and selective nodes according to the real user requests and network topology [6]. In particular, the first two solutions are the opportunistic caching and usually have no good global performance over the arbitrary network topology. Regarding this, there are some related researches. For example, in [7], an oblivious request routing scheme based on cache division was devised. It divided a cache into a number of slices. In [8], authors presented a greedy caching scheme by caching the relatively popular contents at the edge CRs. In [9], an adaptive caching strategy was proposed to enable the caching system to adapt to the dynamic network environment. Furthermore, [10] raised the heterogeneous cache allocation problem in ICN, and its experiments had demonstrated that topological properties and request patterns could affect cache performance. In [11], authors explored the caching capacity of the path by sharing the contents. In [12], the economics and game theory were exploited to address the heterogeneous cache allocation. In [13], the top centrality based cache allocation was proposed to reduce the computing complexity. Different from them, this paper adopts the last approach, i.e., using both network topology and traffic characteristics of user requests to determine the importance of CR and then does the heterogeneous cache allocation.

This paper proposes an Optimal Heterogeneous Cache Allocation (OHCA) mechanism in ICN to distribute the cache capacity across CRs under a constrained cache budget, and the major contributions are concluded as follows. Network topology and traffic characteristics are exploited to determine the importance of CR, where the former considers degree centrality, betweenness centrality and closeness centrality while the latter considers node load and interest preference. Especially for the high-dimensional data with multiple aspects and multiple parameters, t-distributed Stochastic Neighbor Embedding (t-SNE) is used for dimensionality reduction (see Figure 1).

Section 2 presents the method of network modelling. In Section 3, the information including network topology analysis, traffic characteristics analysis and information integration is extracted. The heterogeneous cache allocation method is proposed in Section 4. The experimental results are reported in Section 5 and finally Section 6 concludes this paper. In particular, Section 2 gives the model of traffic distribution which supports the information extraction in Section 3. Especially for the high-dimensional data integrated by network topology and traffic characteristics in Section 3, Section 4 provides a method for dimensionality reduction and on this basis does cache allocation.

2. **Network Modelling.** ICN topology is modelled as $G = (V, E, \Gamma)$, where V is the set of CRs, E is the set of edges and Γ is the request model of interests. Here, V and E are defined as

$$V = \{CR_i | 1 \leq i \leq n \wedge i \in \mathbb{N}_+\}, \tag{1}$$

$$E = \{e_{ij} | CR_i \in V \wedge CR_j \in V, \quad i \neq j\}, \tag{2}$$

where n is the number of CRs.

For Γ, it usually follows the independent reference model and the arrival rate of interest requests follows the Poisson distribution [14]. However, the interest requests are dynamic, showing temporal locality and spatial locality, which indicates that the independent reference model cannot be acceptable to describe these dynamic interest requests. Instead, in this paper, we exploit the shot noise model to describe the distribution of interests.

For any content item, denoted by c, it is expressed as a four tuples, i.e., $< \tau_c, N_c, \lambda_c(t), class_c >$. Among them, τ_c is the initial time when c is requested, N_c is the average number

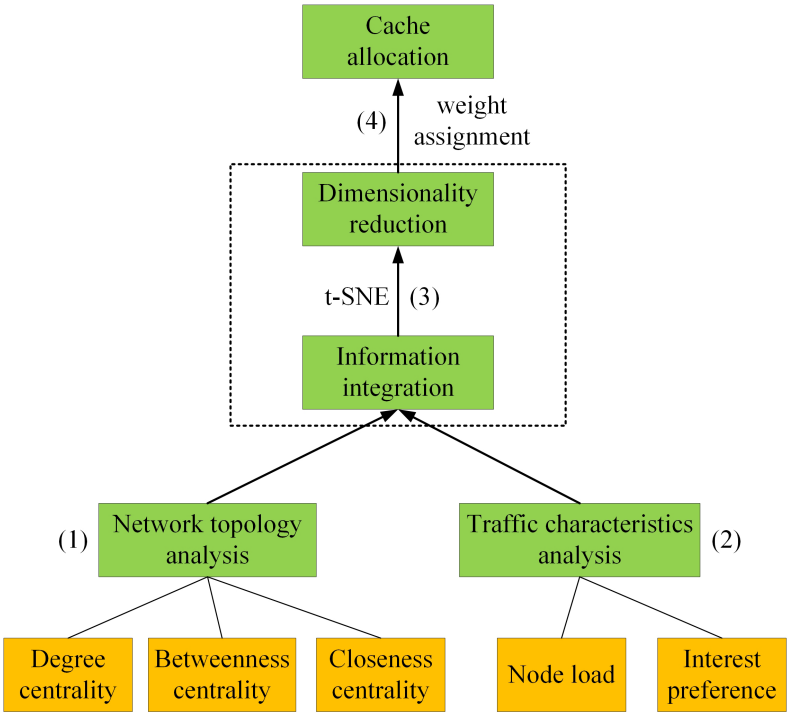

FIGURE 1. The system framework of OHCA

of requests requesting c, $\lambda_c(t)$ is the independent mathematical expectation with respect to c, and $class_c$ is the application type of c. In particular, $\lambda_c(t)$ meets the conditions:

$$\begin{cases} \lambda_c(t) \geq 1 \\ \int_0^{+\infty} \lambda_c(t)dt = 1 \end{cases}. \tag{3}$$

Let N_{class_c}, T_{class_c}, Nt_{class_c} and $\lambda_{class_c}(t)$ denote the average number of requests requesting $class_c$, the average survival time of $class_c$, the total number of content items included by $class_c$ and the integral mathematical expectation with respect to $class_c$ respectively, and we have

$$\lambda_{class_c}(t) = \frac{N_{class_c} Nt_{class_c}}{T_{class_c}}. \tag{4}$$

3. Information Extraction.

3.1. **Network topology analysis.** The CR that locates the hub usually has higher probability to handle more interest requests. As a result, this paper selects three metrics on centrality to determine the importance of CR, i.e., degree centrality, betweenness centrality and closeness centrality, denoted by C_d, C_b and C_c respectively.

At first, C_d is used to reflect the direct influence of CR, and C_d of CR_i is defined as

$$C_{d_i} = \sum_{k=1,k\neq i}^{n} a(CR_i, CR_k), \tag{5}$$

$$a(CR_i, CR_k) = \begin{cases} 1, & CR_i \text{ is adjacent to } CR_k \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where $a(\cdot)$ is the connection function.

Then, C_b of CR_i is defined as

$$C_{b_i} = \sum_{CR_s \neq CR_i \neq CR_t \in V} \frac{\gamma_{st}(i)}{\gamma_{st}}, \tag{7}$$

where γ_{st} is the number of the shortest paths between CR_s and CR_t while $\gamma_{st}(i)$ is that via CR_i. This equation implies the proportion of the shortest paths and it reflects the indirect influence of CR.

Finally, C_c is used to reflect the relative location of CR, and C_c of CR_i is defined as

$$C_{c_i} = \left(\sum_{k=1, k \neq i}^{n} sd(CR_i, CR_k) \right)^{-1}, \tag{8}$$

where $sd(CR_i, CR_k)$ is the distance of the shortest path between CR_i and CR_k.

3.2. **Traffic characteristics analysis.** The traffic characteristics information shows the distribution of user requests clearly, where node load and interest preference are regarded as its two attributes to distinguish the importance of CR. Furthermore, for the evaluation of node load, it depends on the received number of interest requests, the responded number of interest requests and the number of content replacements, denoted by $RecI$, $ResI$ and $RepC$ respectively, which can be obtained during the process of routing. For the evaluation of interest preference, it depends on the aggregation rate of interests and the influence degree of interests, denoted by Agg and Inf respectively.

At first, Agg of CR_i is defined as

$$Agg_i = \frac{Nface_i}{RecI_i}, \tag{9}$$

where $Nface_i$ is the additional interfaces to Pending Interest Table (PIT) of CR_i and $RecI_i$ is $RecI$ of CR_i. In particular, the higher aggregation rate means that the requested content has higher popularity and the CR has more serious congestion.

Then, Inf of CR_i is defined as

$$Inf_i = \sum_{q=1}^{TTL} Hhop_q * \frac{1}{q+1}, \tag{10}$$

where TTL is the abbreviation of time to live and it is the maximal tolerance number of hops, q is the traversed number of hops from interest requester to CR_i, and $Hhop_q$ is the total number of interest requests requesting q. In particular, if a CR can respond to interest requests as many as possible, it means that the cached contents are popular and important.

3.3. **Information integration.** According to the above, we know that the importance of CR is determined by two classes of parameters, i.e., the first class with C_d, C_b and C_c, and the second class with $RecI$, $ResI$, $RepC$, Agg and Inf. Let $Traf_i(t)$ denote the traffic characteristics information collected at CR_i within t time frame, and we have

$$Traf_i(t) = \{RecI_i(t), ResI_i(t), RepC_i(t), Agg_i(t), Inf_i(t)\}, \tag{11}$$

where $Traf_i(t)$ only expresses the traffic condition for CR_i with a five-dimensional vector and it cannot comprehensively reflect the change situation of network traffic. Thus, it requires to select T segments of sampling time, denoted by $t_1, t_2, \ldots, t_T$ respectively. Based on this, combining C_d, C_b and C_c, we have

$$X_i = \{Traf_i(t_1), \ldots, Traf_i(t_T), C_d, C_b, C_c\}, \tag{12}$$

where X_i is the integrated information within T sampling periods and it is a $(5T+3)$-dimensional vector. Consider that there are n CRs, the integrated information including network topology and traffic characteristics is defined as

$$X = \{X_1, X_2, \ldots, X_i, \ldots, X_n\}, \tag{13}$$

which indicates that there are $n(5T+3)$ data points in G.

4. **Heterogeneous Cache Allocation.**

4.1. **Dimensionality reduction.**

The integrated information cannot be used for cache allocation directly due to the high-dimensional feature (i.e., multiple aspects and multiple parameters). Thus, the dimensionality reduction is performed in advance. In this paper, t-SNE [15], a machine learning algorithm is improved and used for dimensionality reduction, as follows. At first, the similarity between data points according to the conditional probability is computed. Then, the gradient training is performed, where a part of the collected data is used as the training set and the other as the verification set.

Let dpx_i and dpx_j denote any two data points in the high-dimensional space, and their mapped data points in the low-dimensional space are denoted by dpy_i and dpy_j. For the high-dimensional space, the law of normal distribution is exploited to compute the similarity, and we have

$$hp_{j|i} = \frac{\exp\left(-dx_{ij}^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-dx_{ik}^2/2\sigma_i^2\right)}, \tag{14}$$

$$dx_{ij} = \|dpx_i - dpx_j\|, \tag{15}$$

where dx_{ij} is the Euclidean distance between dpx_i and dpx_j. For $n(5T+3)$ data points, let Np denote the number of calculations on similarity, and we have

$$Np = \frac{n(5T+3)(n(5T+3)-1)}{2}. \tag{16}$$

It is obvious that the computation overhead is considerably expensive, thus Np should be optimized. Indeed for X, the similarity between data points that are far apart is very small, which has no significant averse impact on dimensionality reduction. Given this, we can only compute the similarity between two adjacent data points. Let Neb_i denote the nearest neighbors set of dpx_i, and (14) is modified as

$$hp_{j|i} = \frac{\exp\left(-dx_{ij}^2/2\sigma_i^2\right)}{\sum_{k\in Neb_i}\exp\left(-dx_{ik}^2/2\sigma_i^2\right)}. \tag{17}$$

In particular, Neb_i is obtained by constructing K-nearest neighbor based on Vantage Point Tree (VPT), and the process consists of four steps as follows. (i) A data point is randomly selected as the highland. (ii) The distance between the highland and the other any data point is computed, and a distance set is obtained. (iii) The mid-value of distance set is computed and these data points are divided into two parts according to the mid-value: if some distance is smaller than the mid-value, its corresponding data point belongs to the left-subtree; otherwise, that for the right-subtree. (iv) The left-subtree and right-subtree are completed by the recursive manner.

Furthermore, the Euclidean distance cannot be employed directly in the high-dimensional space due to the curse of dimensionality [16]; instead, the shortest distance between data points is used to modify (15). Let sdx_{ij} denote the shortest distance between dpx_i and dpx_j, and we have

$$sdx_{ij} = \min\{dx_{ij}, dx_{ik} + dx_{kj}\}. \tag{18}$$

With the above optimization consideration, the similarity between dpx_i and dpx_j is defined as

$$hp_{j|i} = \frac{\exp\left(-sdx_{ij}^2/2\sigma_i^2\right)}{\sum_{k\in Neb_i}\exp\left(-sdx_{ik}^2/2\sigma_i^2\right)}. \tag{19}$$

According to (19), for the high-dimensional space, the joint probability distribution function on dpx_i and dpx_j is defined as

$$hp_{ij} = \frac{hp_{j|i} + hp_{i|j}}{2n(5T+3)}. \tag{20}$$

Consider that the law of t-distribution is exploited in the low-dimensional space, here the degree of freedom is 1, and the joint probability distribution function on dpy_i and dpy_j is defined as

$$lp_{ij} = \frac{\left(1+dy_{ij}^2\right)^{-1}}{\sum_{k\neq i}\left(1+dy_{ik}^2\right)^{-1}}. \tag{21}$$

To the best of our knowledge, the ultimate objective of t-SNE is to minimize the difference between hp_{ij} and lp_{ij}, denoted by $Cost$, and we have

$$Cost = \sum_{\forall i}\sum_{\forall j} hp_{ij}\log\frac{hp_{ij}}{lp_{ij}}. \tag{22}$$

Let Y denote the result of dimensionality reduction from X. The process from X to Y is completed by the stepwise iterations, involving the gradient training:

$$\frac{\delta Cost}{\delta dpy_i} = 4\sum_{j\neq i} dy_{ij}(hp_{ij} - lp_{ij})\left(1+dy_{ij}^2\right)^{-1}. \tag{23}$$

In order to guarantee convergence speed and avoid running into the local optimum, three variables are introduced, i.e., the number of iterations, learning rate and momentum factor for each iteration, denoted by I, η and $\alpha(I)$ respectively. Therefore, the iteration equation is defined as

$$Y(I) = Y(I-1) + \eta\frac{Cost}{Y} + \alpha(I)\left(Y(I-1) - Y(I-2)\right). \tag{24}$$

In particular, the first $\frac{1}{3}T$ data is regarded as the training set while the last $\frac{2}{3}T$ data is regarded as the verification set during the process of gradient training. If the finally acquired two accuracy values are close, t-SNE for dimensionality reduction is acceptable. Mathematically, we have

$$|actr - acve| = \varepsilon \to 0, \tag{25}$$

where $actr$ and $acve$ are the finally acquired accuracy values by the training set and the verification set respectively. Suppose that there are κ analogies generated by the training set and each one has an accuracy $actr_i$ $(1 \leq i \leq \kappa)$ for the training result, and we have

$$actr = \min_{i=1}^{\kappa} actr_i, \tag{26}$$

where $actr_i$ is defined as the ratio of the correct data points and the total data points.

4.2. **Allocation method.** The previous sections have introduced information integration and dimensionality reduction, and this section will make cache allocation based on their outputs (see Figure 1). In this paper, we use weight assignment method to allocate cache capacity for different CRs, where the importance of CR is allocated large cache size and has large weight. According to Y, n CRs are divided into Q classes, where num_j denotes the number of CRs that belong to class j ($1 \leq j \leq Q$ and $j \in \mathbb{N}_+$). Let W denote the weight vector to which Q classes correspond, and we have

$$W = \{w_1, w_2, \ldots, w_i, \ldots, w_Q\}, \tag{27}$$

$$\begin{cases} \sum_{j=1}^{Q} w_j = 1 \\ 0 < w_j < w_{j+1} < 1 \end{cases}. \tag{28}$$

Suppose that the total cache budget is C_{total} and that num_j CRs are allocated C_j capacity, and we have

$$C_j = w_j C_{total}. \tag{29}$$

For these CRs that belong to the same class, their cache capacities are equally allocated. For one of num_j CRs, let C_{ji} denote its cache capacity, and we have

$$C_{ji} = \frac{C_j}{num_j}. \tag{30}$$

5. Performance Evaluation.

5.1. **Simulation setup.** The proposed OHCA is implemented over Network Simulator 3 (NS3) based on C++ programming language, running on a personal computer with Intel(R) core(TM)i5-6200u, CPU2.92 GHz, 4GB RAM. The simulation is driven based on the real YouTube dataset, of which the collection of trace comes from a campus network measurement [17]. In particular, the YouTube dataset contains 18751 user requests for 13764 short videos across 2377 hosts, which has some inherent distribution laws. In spite of this, the dataset does not present the designated network topology. To address this, Global Technology Service for Continent Europe (GTS-CE) with 130 nodes and 168 edges [18] is used as simulation topology. According to those distribution laws of the original YouTube dataset, we distribute 18751 user requests and 13764 short videos across 130 nodes rather than 2377 hosts in the same proportion.

We compare the proposed OHCA with two state-of-the-art mechanisms, shorted for BCN [12] and BToC [13] respectively. Certainly, there are many heterogeneous cache allocation mechanisms. However, most of them are not the latest studies or do not show the concrete schemes. Different from those mechanisms, [12] and [13] present the relatively systematic designs and thus they are selected as the baselines. In addition, Average Hit Ratio (AHR), Average Utilization Rate (AUR) and Average Routing Delay (ARD) are considered as three evaluation metrics. Furthermore, we divide these 18751 interest requests into five intervals in chronological order. For each interval, we extract 400 interest requests, i.e., $[1, 400]$, $[3751, 4150]$, $[7501, 7900]$, $[11251, 11650]$ and $[15001, 15400]$ and report the corresponding experimental results. For these parameters, we make simulations under different settings to find the proper one. As shown in Table 1, we give the settings for the involved parameters.

TABLE 1. Parameters

Parameter	Setting	Ownership
Collection duration	24h	YouTube
The size of dataset	166GB	YouTube
The period of sampling	10mins	OHCA
T	$24 \times 60/10 = 144$	OHCA
C_{total}	20GB	OHCA
η	0.6	OHCA
ε	0.18	OHCA
The number of simulations	30	OHCA
The network bandwidth	10Gb/s	GTS-CE

5.2. **t-SNE optimization analysis.** The conventional t-SNE has the considerably high computation overhead; thus this paper optimizes Np by constructing VPT-based K-nearest neighbor. For different interest requests, we report the corresponding Np before and after optimization, as shown in Table 2. We observe that the number of required calculations after optimization only approximatively accounts for 8.1% of that before optimization, which indicates that the optimization effect is very significant.

TABLE 2. t-SNE optimization test

Interests	400	600	1000	1500	2500
Np before optimization	4417013055	4417013055	4417013055	4417013055	4417013055
Np after optimization	366612081	366574145	366503926	366641157	366630394

5.3. **Comparison analysis.**

5.3.1. *Cache hit ratio.* The cache hit ratio is defined as $Num_{hit}/Num_{success}$, where Num_{hit} is the number of interest requests which are satisfied by CS and $Num_{success}$ is the successful number of interest requests. AHRs for OHCA, BCN, BToC under different interest requests are reported in Figure 2. We observe that OHCA has the highest AHR, followed by BToC and BCN. In fact, only BCN does not consider the dynamic change of interest requests, that is to say, the corresponding CRs cannot satisfy interest requests as many as possible, which causes that lots of interest requests have to retrieve the contents from the origin server. Thus, BCN has the lowest AHR. Although both OHCA and BToC are devised for the dynamic interest requests, BToC only considers the inherent

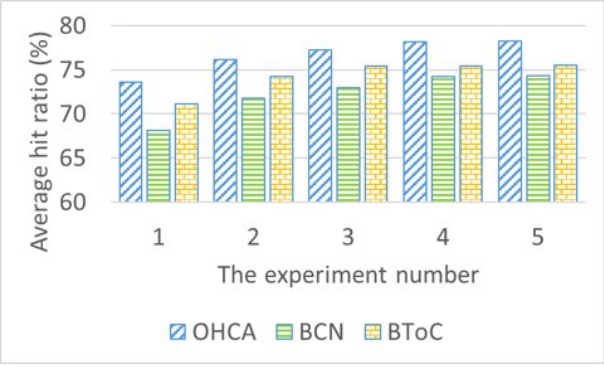

FIGURE 2. Average hit ratios for OHCA, BCN and BToC

network topology; instead, OHCA comprehensively considers network topology and traffic characteristics. Furthermore, for the same mechanism, we observe that the subsequent experiments have higher AHR and gradually tend to become stable, this is because the initial experiment has no optimal caching collaboration and the latter experiments continuously adjust the cached contents until the system is stable in terms of the YouTube dataset.

5.3.2. *Cache utilization rate.* The cache utilization rate is used to measure the usage of cache and defined as $\sum_{i=1}^{n} c_{ui}/C_{total}$, where c_{ui} is the used cache for CR_i. In particular, the large cache utilization rate means good cache allocation. AURs for OHCA, BCN and BToC under different interest requests are reported in Figure 3. We observe that OHCA has the highest AUR, followed by BCN, BToC, which suggests that the high cache utilization rate benefits from the cache allocation mechanism. In fact, it is very hard to clearly and accurately explain the inherent reason on which mechanism has the best cache allocation performance. However, by reviewing the external factors, OHCA considers network topology and traffic characteristics, i.e., analyzing the most comprehensive factors (i.e., degree centrality, betweenness centrality, closeness centrality, node load and interest preference). Especially the introduction of node load and interest preference can make CR cache more valuable videos, which cause that OHCA has higher AUR than BCN and BToC. For BCN and BToC, although the information on network topology is analyzed, BToC only considers degree centrality irrespective of betweenness centrality and closeness centrality, which results in that it cannot obtain the relatively optimal cache size according to the extracted topology information. From the other perspective, BCN leverages the game theory and tries to find the optimal solution, while BToC only obtains the sub-optimal cache location in SPT. Given the two aspects, BCN has higher AUR than BToC.

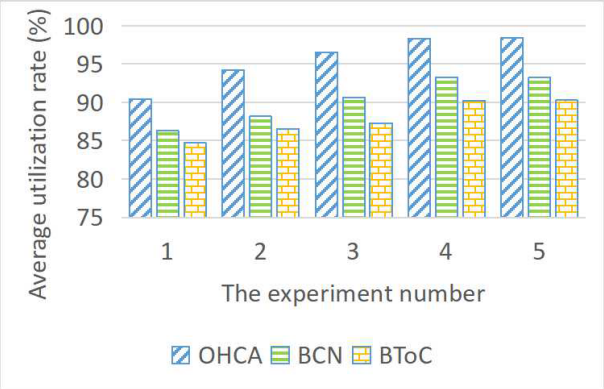

FIGURE 3. Average utilization rates for OHCA, BCN and BToC

5.3.3. *Routing delay.* The routing delay is defined as the difference between the time-point when interest request is sent and that when the corresponding content is obtained by user. ARDs for OHCA, BCN and BToC under different interest requests are reported in Figure 4. We observe that OHCA has the smallest ARD, followed by BToC and BCN, this is because OHCA has the highest AHR and AUR and it can respond to more interest requests as quickly as possible.

5.4. **Discussion.** As can be seen from Section 5.3, we know that the proposed OHCA has the best performance in terms of average hit ratio, average utilization rate and average routing delay, which suggests that it is optimal from the perspective of experiments. In

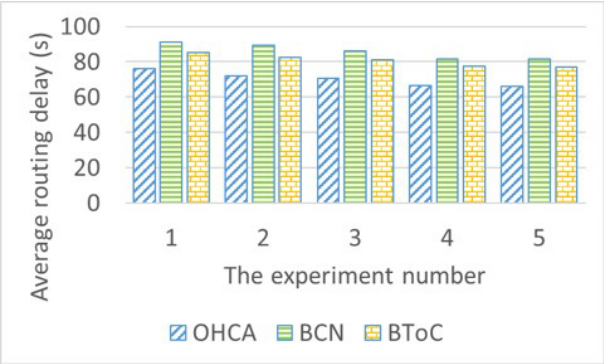

FIGURE 4. Average routing delays for OHCA, BCN and BToC

particular, we can observe that the average utilization rate of OHCA is always larger than 90%, which indicates that the proposed cache allocation is considerably acceptable.

6. **Conclusions.** This paper proposes an optimal heterogeneous cache allocation mechanism, called OHCA, which consists of two parts, i.e., cache allocation which distributes the cache capacity across CRs under a constrained and fixed total cache budget. Regarding the cache collocation, it is based on weight assignment by determining the importance of CR which depends on network topology and traffic characteristics. For the integrated information, t-SNE is used to do dimensionality reduction. The proposed OHCA is simulated based on the real YouTube dataset over GTS-CE topology, and the comparison experiments reveal that OHCA outperforms two state-of-the-art cache allocation mechanisms.

However, as a novel heterogeneous cache allocation mechanism, the proposed OHCA has two distinguished shortages. On the one hand, it lacks the theoretical analysis to support the optimal conclusion. On the other hand, more comprehensive factors should be considered during the process of information extraction. In future, we improve and enhance OHCA around above two issues. Specifically, we plan to prove OHCA in mathematics under some feasible assumptions. In addition, we also plan to consider and further the distribution of content location in order to do more complex and more comprehensive data analysis.

REFERENCES

[1] M. Zhang, H. Luo and H. Zhang, A survey of caching mechanisms in information-centric networking, *IEEE Communications Surveys & Tutorials*, vol.17, no.3, pp.1473-1499, 2015.

[2] D. Perino and M. Varvello, A reality check for content centric networks, *Proc. of ACM SIGCOMM on Information-Centric Networking*, pp.44-49, 2011.

[3] D. Rossi and G. Rossini, On sizing CCN content stores by exploiting topological information, *Proc. of IEEE INFOCOM*, pp.280-285, 2012.

[4] W. K. Chai, D. He, I. Psaras et al., Cache less for more in information-centric networks (extended version), *Computer Communications*, vol.36, no.7, pp.758-770, 2013.

[5] A. Kalla and S. K. Sharma, Exploring off-path caching with edge caching in information centric networking, *Proc. of International Conference on Computational Techniques in Information and Communciation Technolgies*, pp.1-6, 2016.

[6] A. Kalla and S. K. Sharma, A constructive review of in-network caching: A core functionality of ICN, *Proc. of IEEE International Conference on Computing, Communication and Automation*, pp.567-574, 2017.

[7] W. Chu, M. Dehghan, J. C. S. Lui et al., Joint cache resource allocation and request routing for in-network caching services, *Computer Networks*, vol.131, pp.1-14, 2018.

[8] B. Banerjee, A. Kulkarni and A. Seetharam, Greedy caching: An optimized content placement strategy for information-centric networks, *Computer Networks*, vol.140, pp.78-91, 2018.
[9] S. Tarnoi, W. Kumwilaisak, V. Suppakitpaisarn et al., Adaptive probabilistic caching technique for caching networks with dynamic content popularity, *Computer Communicatons*, vol.139, pp.1-15, 2019.
[10] Y. Wang, Z. Li and G. Tyson, Optimal cache allocation for content-centric networking, *Proc. of IEEE International Conference on Network Protocols*, pp.1-10, 2013.
[11] I. Psaras, W. K. Chai and G. Pavlo, In-network cache management and resource allocation for information-centric networks, *IEEE Trans. Parallel and Distributed Systems*, vol.25, no.11, pp.2920-2930, 2014.
[12] S. Hoteit, M. El-Chamie, D. Saucez et al., On fair network cache allocation to content providers, *Computer Networks*, vol.103, pp.129-142, 2016.
[13] Y. Wang, Z. Li, G. Tyson et al., Design and evaluation of the optimal cache allocation for content-centric networking, *IEEE Trans. Computers*, vol.65, no.1, pp.95-107, 2016.
[14] H. Yu, D. Zheng, B. Zhao et al., Understanding user behavior in large-scale video-on-demand systems, *ACM SIGOPS Operating Systems Review*, vol.40, no.4, pp.333-344, 2006.
[15] L. Maaten and G. Hinton, Visualizing data using t-SNE, *Journal of Machine Learning Research*, vol.9, pp.2579-2605, 2008.
[16] C. C. Aggarwal, A. Hinneburg and D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, in *Database Theory – ICDT 2001. Lecture Notes in Computer Science*, J. Van den Bussche and V. Vianu (eds.), Berlin, Heidelberg, Springer, 2001.
[17] *YouTube Traces*, http://traces.cs.umass.edu/index.php/Network/Network, Accessed on Jun 18, 2020.
[18] *GTS-CE*, http://www.topology-zoo.org, Accessed on Jun 18, 2020.