Modeling of Content Dissemination Networks on Multiplexed Caching Hierarchies

Satoshi Imai
Fujitsu Laboratories Ltd.
4–1–1 Kamikodanaka, Kawasaki,
Kanagawa, Japan
Email: imai.satoshi@jp.fujitsu.com

Kenji Leibnitz
CiNet, NICT and Osaka University
1–4 Yamadaoka, Suita,
Osaka, Japan
Email: leibnitz@nict.go.jp

Masayuki Murata
Osaka University
1–5 Yamadaoka, Suita,
Osaka, Japan
Email: murata@ist.osaka-u.ac.jp

***Abstract*—In-network caching technologies like *Content-Centric Networking* (CCN) are expected to reduce the network traffic and improve the service quality, such as communication latency, by storing content data on routers near to users. Meanwhile, the adaptive cache management using *Time-To-Live* (TTL) of content can realize efficient memory management per content. However, for a distributed cache system such as CCN, it is difficult to evaluate cache performance and network resources required in the cache mechanism using the TTL value. Therefore, we propose a theoretical model, which can analyze the impact of TTL-based caching on network resources and cache performance, and evaluate some scenarios using the proposed model. We finally introduce a cache mechanism using energy efficient TTLs and show its effectiveness by the model-based analysis.**

***Keywords*-In-network caching; distributed cache system; Content Centric Networking; Time-To-Live**

I. INTRODUCTION

The currently increasing network traffic is caused by the growing number of content dissemination services in the network and *Content Delivery Networks* (CDN) are well known as efficient content delivery mechanisms. Since the CDN service can provide content delivery at the edge of the networks by allocating content replicas in cache servers, which are in geographical proximity to users, it is expected to reduce the network traffic. Moreover, the caching services can improve communication quality, such as latency and throughput, in the delivery of content. Recently, a new communication paradigm, namely *Content Centric Networking* (CCN) [1], has been proposed. The CCN-enabled routers have autonomous caching functionality for content data. In the content dissemination mechanism of CCN, content publishers advertise newly released content from the origin site of the content along predefined routes. A content request (*Interest*) is forwarded on each content router (*CR*) based on the *Forwarding Information Base* (FIB) until the requested content is found. An *Interest* forwarded on a *CR* is added to the *Pending Interest Table* (PIT) in order to remember the interface on which to send back the replies (*Data*). When the requested content is found on a *CR*, *Data* of the content are transmitted based on the PIT. Furthermore, *Data* are cached on all *CR*s along the transmission route based on the specific replacement policy such as Least Recently Used (LRU) or Least Frequently Used (LFU). Therefore, CCN can automate the placement and delivery of *Data* by *CR*s on networks and it is highly expected to reduce network traffic and improve the communication quality.

However, the network traffic and communication quality are influenced by the cache locations because content is generated by many publishers at various locations in CCN. Moreover, the caching performance depends on the memory size in each *CR* on multiplexed delivery trees rooted at each origin site of content. Therefore, it is a major issue to analyze the impact of the distributed cache mechanism on network resources, such as memory or network devices, and cache performance, such as cache hit ratio and hop length, while delivering content.

Meanwhile, the persistent storage of content in each *CR* is inefficient because the content, such as video streaming, generally has a limited lifetime. Therefore, dynamic caching mechanisms often use a limited period of time, called *Time-To-Live* (TTL) for content. Moreover, TTL-based caching can improve the scalability of cache management compared with LRU or LFU replacement with cache coordination which sorts content by popularity and selects which content should be discarded. In this paper, we first propose an analytical model to evaluate the impact of TTL on cache performance and network resources of content in a distributed cache mechanism like CCN. Furthermore, we demonstrate evaluation results for some network scenarios using the proposed model. In addition, we introduce a cache mechanism using energy efficient TTLs as one possible application of TTL-based caching and show the effectiveness of the proposed cache mechanism using our model.

The remainder of this paper is organized as follows. Section II discusses the TTL-based cache mechanism followed by Section III which summarizes related work. We propose our analytical model in Section IV and demonstrate evaluation results using the proposed model in Section V. Furthermore, in Section VI we introduce a cache mechanism to improve energy efficiency using TTL and we evaluate the effectiveness of the proposed mechanism. Finally, we conclude the paper in Section VII.

II. TTL-BASED CACHE MECHANISM AND ISSUES

In this paper, we assume that each *CR* executes data caching using TTL of content which can, for instance, be signaled in the data header of the content or set at each *CR* in advance.

In TTL-based caching, each *CR* resets the time counter to

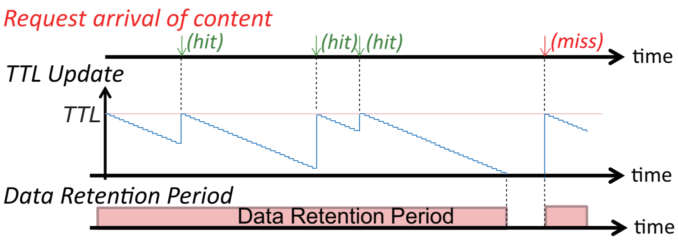


Figure 1: Traditional TTL-based caching

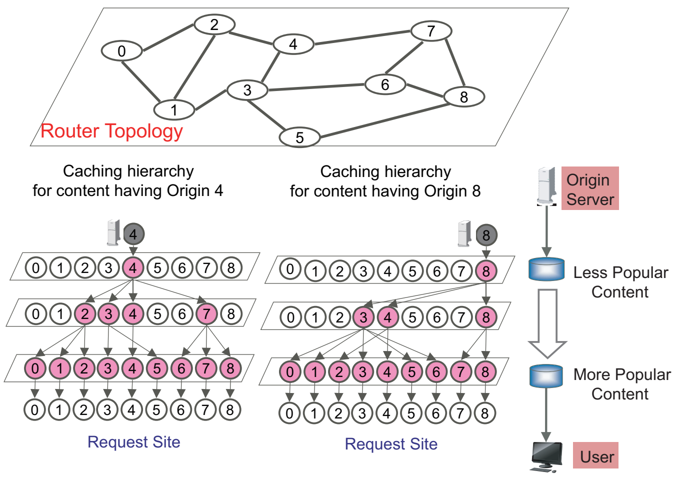


Figure 2: An example of caching hierarchies for origin sites 4 and 8

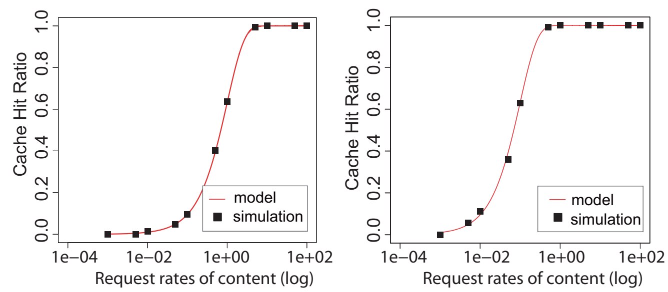
the TTL of content every time a new request for this content arrives and decreases the counter by 1 every time unit (cf. Figure 1). In this mechanism, each *CR* caches data of content delivered by another *CR* or an origin server when the content counter is above 0 and discards the data of content when the counter becomes 0.

Meanwhile in CCN, each *CR* autonomously constructs some caching hierarchies rooted at each origin site of content (cf. Figure 2). The caching hierarchy is constructed by routes between the origin site, caching routers, and users, such that less popular content is cached on *CR*s near to the origin site and more popular content is cached on *CR*s near to users. Therefore, it is difficult to evaluate the impact of TTL-based caching on network resources and performance because the characteristics in the distributed cache mechanism depend on the caching hierarchies connected by distributed cache nodes.

In this paper, we first propose an analytical model using matrix equations to evaluate the cache characteristics on multiplexed caching hierarchies of content and evaluate the validity of the proposed model and the impact of the TTL value.

III. Related Work

The modeling of efficient memory management is a major issue in content caching systems. Traditionally, there are content placement algorithms [2], [3] as a solution for *File Allocation Problems* [4], which minimize the cost imposed for content storage and queries, or maximize performance such as



(a) $TTL = 1$ (b) $TTL = 10$

Figure 3: Cache hit ratio when the requests per content with the rate λ input to a *CR* at an independent and identically distributed exponential interval and "total request rates of a content item [requests/sec]" and "TTL [sec]" to various values

distance to content. Baev *et al.* [2] propose a linear programming model which minimizes content placement cost and an approximation solution using a linear relaxation. Furthermore, Qui *et al.* [3] develop a method and compare it with some replica placement algorithms to solve a *K*-median problem for CDNs.

In contrast to the above-mentioned content placement problems, Borst *et al.* [5] formulate a liner programming model based on a hierarchical structure for content locations to minimize bandwidth costs through a distributed solution. In view of energy efficiency for content delivery networks, Guan *et al.* [6] build energy models of traffic transmission power and caching power for content delivery architectures such as "Conventional and decentralized server-based CDN", "Centralized server-based CDN using dynamic optical bypass", and CCN.

Furthermore, Carofiglio *et al.* [7] explore the impact of storage management on the cache performance per application in CCN and evaluate the effectiveness of static storage partitioning and dynamic management by priority-based weighted fair schemes combined with TTL-based caching. Moreover, they study the possibility of improving cache scalability in TTL-based caching without cache coordination.

However, these proposals don't discuss the cache characteristics using TTL of content and the impact of TTL on network resources and cache performance on the multiplexed caching hierarchies. Therefore in this paper, we construct a model to analyze the cache characteristics using TTL and evaluate the cache performance for the TTL value.

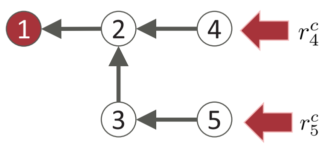
IV. Analytical Model

We first propose the evaluation model to analyze the cache performance using TTL of a content in the distributed cache system having multiplexed caching hierarchies.

In TTL-based caching, the cache probability of content c having request rates λ^c to a *CR* can be expressed by the following function [8].

$$\boldsymbol{f}(\lambda^c, TTL^c) = 1 - e^{-\lambda^c TTL^c}$$

As shown in Figure 3, we demonstrate that this statistical function can provide a good approximation of the cache hit



$$\mathbf{\Lambda}^c[s] = \begin{bmatrix} \lambda^c_{(1,4)}[s] & \lambda^c_{(1,5)}[s] \\ \lambda^c_{(2,4)}[s] & \lambda^c_{(2,5)}[s] \\ \lambda^c_{(3,4)}[s] & \lambda^c_{(3,5)}[s] \\ \lambda^c_{(4,4)}[s] & \lambda^c_{(4,5)}[s] \\ \lambda^c_{(5,4)}[s] & \lambda^c_{(5,5)}[s] \end{bmatrix} \quad \mathrm{R}^c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ r_4^c & 0 \\ 0 & r_5^c \end{bmatrix}$$

$$\mathrm{D}^c[s] = \begin{bmatrix} 0 & e^{-\sum_{j=4}^5 \lambda^c_{(2,j)} TTL^c} & 0 & 0 & 0 \\ 0 & 0 & e^{-\sum_{j=4}^5 \lambda^c_{(3,j)} TTL^c} & e^{-\sum_{j=4}^5 \lambda^c_{(4,j)} TTL^c} & 0 \\ 0 & 0 & 0 & 0 & e^{-\sum_{j=4}^5 \lambda^c_{(4,j)} TTL^c} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4: An example of matrices $\mathbf{\Lambda}^c$, $\mathbf{R}^c$, and D^c for the request propagation on the delivery tree having origin 1

ratio of content at a *CR* under the assumption that content requests arrive as a Homogeneous Poisson Process. We next show the matrix model of request propagation of content in TTL-based caching on caching hierarchy of content. The propagation of each request (*Interest*) of content c on its caching hierarchy is expressed by the following model.

$$\mathbf{\Lambda}^c[s+1] = \mathrm{D}^c[s] \cdot \mathbf{\Lambda}^c[s] + \mathbf{R}^c, \forall c \quad (1)$$

Under the condition that M, N, and s are the number of *CR*s, the number of sites having requesting users, and the number of steps that each request propagates to the next *CR*, respectively, we define $\mathbf{\Lambda}^c$ as the $M \times N$ matrix consisting of the request rates $\lambda^c_{(i,j)}$ of content c from the requesting user in site j to CR_i and $\mathbf{R^c}$ as the $M \times N$ matrix of which elements are the request rates r_i^c of content c from users in site i.

$$\mathbf{\Lambda}^c[s] := [\lambda^c_{(i,j)}]_{M\times N}$$

$$[\mathbf{R}^c]_{i,j} := \begin{cases} r_i^c, & \text{when } CR_i \text{ is located on } j\text{-th} \\ & \text{site having requesting users} \\ 0 & \text{otherwise} \end{cases}$$

D^c is the $M \times M$ matrix of request propagation for content c as follows.

$$[\mathrm{D}^c]_{m,n} := \begin{cases} 1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(n,k)}[s], TTL^c), \\ \quad m = parent\_node(n) \\ 0 \quad \text{otherwise} \end{cases} \quad (2)$$

Here we defined the condition that CR_m is a parent node of CR_n as "$m = parent\_node(n)$". In Figure 4, we show an example of these matrices for the delivery tree having origin 1.

In the iterative matrix equation, we can consider the request propagation process and data caching at each *CR* for content requested from each site. Moreover, the steady state of network resources and cache performance per content can be derived by iteratively calculating the equation s_{max}-times, which is the maximum number of hops from each site having requesting users to its origin site.

Using the proposed solution, we can model the system state and the cache performance for content c such as

- memory usage per content in each *CR*,
- the total amount of transmission data in the network,
- power consumption which is the sum of "cache allocation power" and "traffic transmission power",
- cache hit ratio per content which is the probability that the content is cached in the network, and
- average hop length per content.

A. Memory Usage

The memory usage of content c at CR_i is derived using the data size θ_c of content c as follows.

$$\mathrm{U}_i^c := \theta_c \boldsymbol{f}(\sum_k^N \lambda^c_{(i,k)}, TTL^c). \quad (3)$$

B. Transmission Data

The total amount of data delivery of content c through all *CR*s is derived as

$$\mathrm{Dt}^c := \theta_c \sum_j^N Tr_j^c \quad (4)$$

using the following vector consisting of the cumulative number of traffic flows Tr_j^c through each *CR* on the delivery route for content c having origin site o requested by users in site j.

$$\begin{aligned} \mathbf{Tr}^c &:= [Tr_1^c \cdots Tr_j^c \cdots Tr_N^c]^{\mathrm{T}} \\ &= (\mathbf{H} * \mathbf{\Lambda}^c)^{\mathrm{T}} \begin{bmatrix} \boldsymbol{f}(\sum_k^N \lambda^c_{(1,k)}, TTL^c) \\ \vdots \\ \boldsymbol{f}(\sum_k^N \lambda^c_{(M,k)}, TTL^c) \end{bmatrix} \\ &\quad + (\mathbf{H}[o,] * \mathbf{\Lambda}^c[o,])^{\mathrm{T}} (1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)) \end{aligned} \quad (5)$$

Here, we define "$*$" as the element-wise product of a matrix or vector and $\mathbf{H} = [h_{(i,j)}]_{M\times N}$ as the matrix consisting of shortest hop length $h_{(i,j)}$ from CR_i to CR_j.

Moreover, the second term in (5) presents the amount of transmission data which aren't cached on the network.

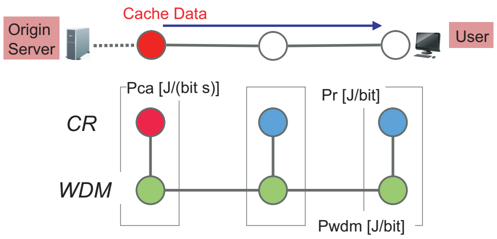


Figure 5: Network model

TABLE I: VARIABLES IN THE PROPOSED MODEL

Variable	Definition
M	The number of *CR*s
N	The number of sites having requesting users
θ_c	Data size of content c
$\lambda^c_{(i,j)}$	Request rates to CR_i for content c requested by users in site j
r^c_i	Request rate of content c requested by users in site i
TTL^c	TTL of content c at CR_i
U^c_i	Memory usage of content c at CR_i
Dt^c	Total amount of data delivery of content c through all *CR*s
Tr^c_j	Cumulative number of traffic flows through each *CR* on the delivery route for content c requested by users in site j
CHR^c	Cache hit ratio of content c in the network
AHL^c	Average hop length of content c
Hp^o	Hop length from origin server to the content router in origin site o
CP^c	Total power consumption [J] for data storage of content c in 1 sec
TP^c	Total power consumption [J] delivering content c on the delivery routes
P_{ca}	Power density for storage [J/(bit·s)]
P_r	Power density of a router [J/bit]
P_{wdm}	Power density of a WDM node [J/bit]

C. Power Consumption

We consider total power consumption based on *Energy Proportional Networks* [9], [10] in which power consumption of each device is proportional to its usage for a network composed of *CR*s and Wavelength Division Multiplexing (WDM) nodes in Figure 5. In this paper, we assume 1 sec as time unit.

Cache allocation power: CP^c [J] for storing data of content c in 1 sec, i.e., the total power consumed by storing content c on each *CR* in the network, is defined as

$$\mathrm{CP}^c := \theta_c P_{ca} \sum_i^M \boldsymbol{f}(\sum_k^N \lambda^c_{(i,k)}, TTL^c), \tag{6}$$

where P_{ca} is the memory power density [J/(bit· s)].

Traffic transmission power: TP^c [J] i.e., the total power consumed by network devices when data of content c are delivered on the shortest routes, is derived as

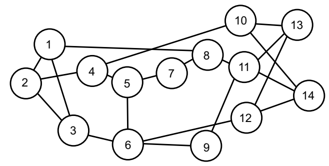
$$\mathrm{TP}^c := (P_r + P_{wdm})\mathrm{Dt}^c, \tag{7}$$

where P_r and P_{wdm} are the power densities [J/bit] of a router and of a WDM node along the delivery routes, respectively.

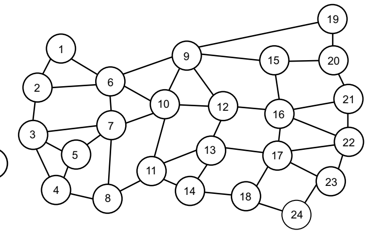
D. Cache Hit Ratio

The cache hit ratio of content c having origin o in the network is derived as

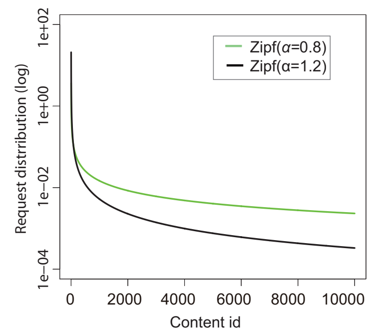
$$\mathrm{CHR}^c := 1 - \frac{\sum_j^N \lambda^c_{(o,j)} \left(1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)\right)}{\sum_j^N r^c_j}. \tag{8}$$



(a) Test topology A



(b) Test topology B



(c) Request distribution r^c_i

Figure 6: Evaluation conditions

E. Average Hop Length

The average hop length of content c having origin o is derived as

$$\mathrm{AHL}^c := \frac{\sum_j^N \left(Tr^c_j + \mathrm{Hp}^o \lambda^c_{(o,j)} (1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)\right)}{\sum_j^N r^c_j}. \tag{9}$$

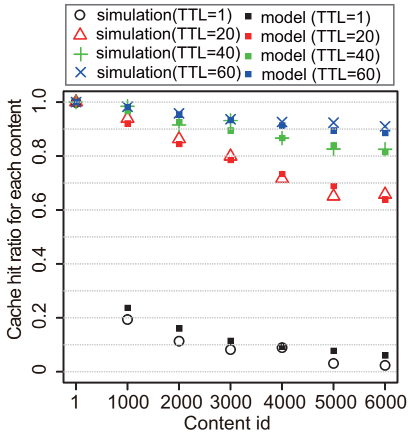
The second term of the numerator is a penalty for the hop length of content c, which isn't cached on any *CR*s in the network and for which a request reaches its origin server and Hp^o is the hop-length from the origin server to the content router in origin site o and is used as penalty if the content is not found in the network. All variables in the proposed model are summarized in Table I.

V. Evaluation using the Proposed Model

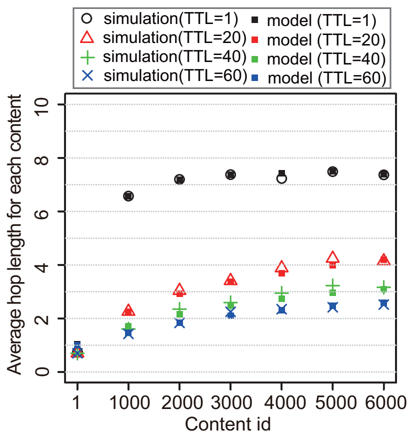
We evaluate the cache characteristics in TTL-based caching when changing the TTL value of content. The evaluation conditions are set to the following.

- **Test networks:** NSF topology with 14 *CR*s (Topology A), cf. Figure 6(a) / US-backbone topology with 24 *CR*s (Topology B), cf. Figure 6(b). The maximum number of hops (s_{max}) is 5 in Topology A and 7 in Topology B. Furthermore, we assume that the memory size of each *CR* is infinite and each site has requesting users for all content items, which means M is equal to N. For the evaluation, we set Hp^o to 5 as the penalty of hop length.

 ISBN: 978-1-61208-318-6

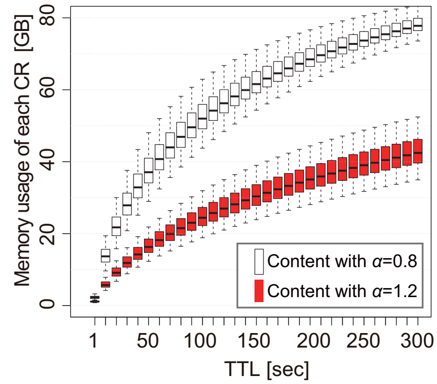


(a) Cache hit ratio

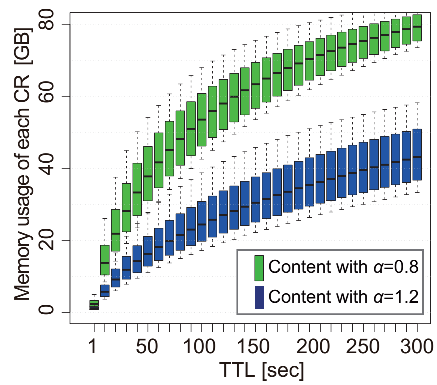


(b) Average hop length

Figure 7: Cache performance estimated by the proposed model and calculated by simulations for different content ids



(a) Topology A



(b) Topology B

Figure 8: Box plot of memory usage at each *CR* when the TTL value is changed

- **Content information:** Zipf-distributed requests from each site i for $K = 10000$ content items are defined as $r_i^c = \gamma k^{-\alpha}/c$, $c = \sum_{k=1}^{K} k^{-\alpha}$, cf. Figure 6(c). We set α to 0.8 for User Generated Content (UGC) and 1.2 for VoD [12] and γ to 100 [requests/sec]. Furthermore, the origin site t of content ID k is set randomly based on a uniform distribution. The content size is geometrically distributed with mean 10 MB [13].

A. Verification of the Proposed Model

To verify the proposed model, we compare the cache performance using the model with that measured by simulations for 7 content items with $\alpha = 0.8$ in Topology A. In the evaluations, we set the TTL value as $\{1, 20, 40, 60\}$ [sec].

Figure 7 shows that the cache hit ratio and average hop length for each content provide suitable approximations of the simulation results. As a result, we see that the proposed model can express the statistical characteristics for TTL-based caching.

B. Impact of TTL

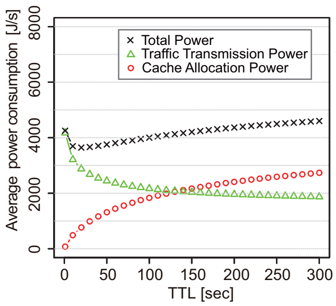
For the next evaluation, we define TTL as the same value for each content which is changed from 1 [sec] to 300 [sec] on the assumption that the TTL value is signaled in the data header of content.

Figure 8 shows the memory usage at each *CR* when the TTL value is changed. In these results, the memory usage of each *CR* becomes larger as the TTL value becomes larger. Moreover, the memory usage for content with $\alpha = 0.8$ is larger than that for content with $\alpha = 1.2$ because less popular content with $\alpha = 0.8$ has higher request rates and is easier to be cached than that with $\alpha = 1.2$.

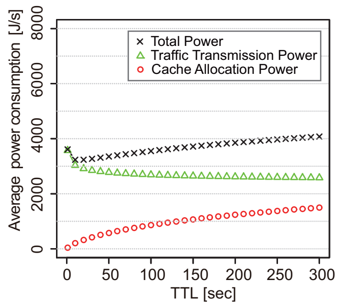
Furthermore, Figure 9 shows the power consumption of the target network according to the change of the TTL value using the power densities of network devices shown in Table II. In addition, Figure 10 presents cache hit ratio for all content items calculated by (8) and average hop length for all content items calculated by (9).

TABLE II: POWER DENSITY PARAMETERS

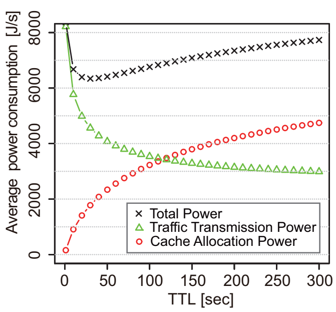
Device (Product)	Power / Spec	Power Density
DRAM	10 W / 4 GB	$P_{ca} = 3.125 \times 10^{-10}$ J/(bit · s)
Content Router (CRS-1)	4185 W / 320 Gbps	$P_r = 1.3 \times 10^{-8}$ J/bit
WDM (FLASHWAVE9500)	800 W / 480 Gbps	$P_{wdm} = 1.67 \times 10^{-9}$ J/bit



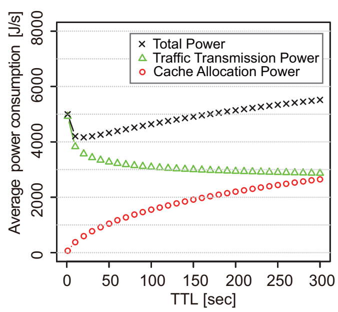
(a) Content with $\alpha = 0.8$ in Topology A



(b) Content with $\alpha = 1.2$ in Topology A

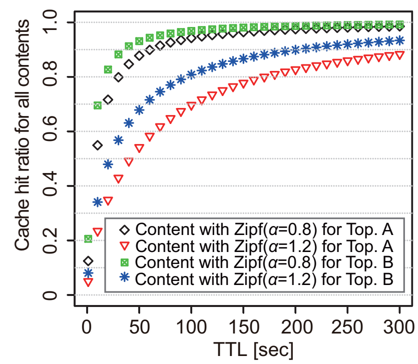


(c) Content with $\alpha = 0.8$ in Topology B

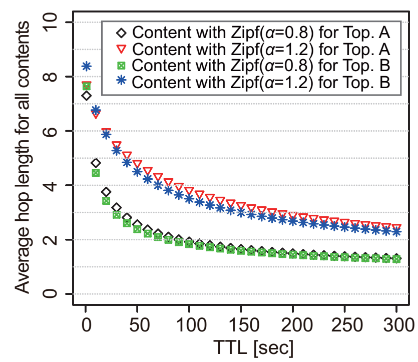


(d) Content with $\alpha = 1.2$ in Topology B

Figure 9: Power consumption of the network when the TTL value is changed



(a) Cache hit ratio for all content items



(b) Average hop length for all content items

Figure 10: Cache performance when the TTL value is changed

Figure 9 demonstrates the tradeoff between *cache allocation power* and *traffic transmission power* for the change of the TTL value. Figures. 9(a) and (c) show that there is a point reversing the relation of *cache allocation power* and *traffic transmission power* for the TTL value. Therefore, the energy impact of TTL is also different depending on the network conditions and the proposed model can search for the energy efficient TTL in consideration of the tradeoff of power consumption.

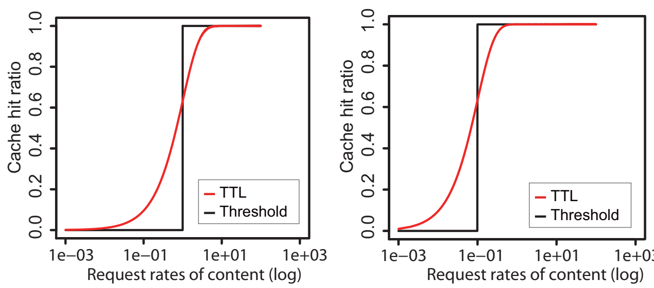
Meanwhile in Figure 10(a), the cache hit ratio is also low in the region of the TTL values leading to lower power consumption. Therefore, we should consider the relation between cache hit ratio and power consumption to search for the energy efficient TTL. Furthermore, Figure 10(b) shows that the average hop length of all content items becomes smaller as the TTL becomes larger. In these results, approaching the average hop length of 1 hop means that all content items are cached in all *CR*s. The cache hit ratio approaches to around 100 % as the average hop length is approaching to 1 and the memory usage becomes larger. Therefore, using the proposed model, we can analyze the cache characteristics in the distributed cache system and provide a design guideline for TTL of content in view of energy efficiency or efficient memory usage in each *CR*. Next, we introduce an energy efficient cache mechanism using TTL as application and demonstrate the effectiveness of the energy efficient TTL.

VI. Application to a Cache Mechanism using Energy Efficient TTLs

In consideration of energy efficiency in content dissemination networks, we previously proposed an ILP model to design the most energy efficient cache locations taking the multiplexed caching hierarchies into account [14].

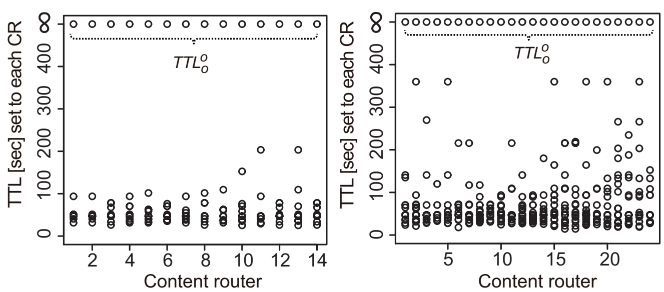
In [15], we proposed the threshold-based cache mechanism to locally search for locations which are near to the most energy efficient locations. In threshold-based caching, every *CR* automatically pre-designs a threshold of request rates of content using local information on each caching hierarchy before cache operation and the content data are cached when the request rate of the content is above a pre-designed threshold or isn't cached when the request rate is below that threshold.

 ISBN: 978-1-61208-318-6



(a) $TTL = 1$, $Th = 1$ (b) $TTL = 10$, $Th = 0.1$

Figure 11: Comparison of cache hit ratio at a *CR* with threshold-based caching and TTL-based caching



(a) Topology A (b) Topology B

Figure 12: Energy efficient TTL for each topology

In threshold-based caching, *CR*$_i$ has the threshold Th_i^o [requests/sec] of request rates for origin site o of content, which is uniquely determined for each delivery tree in the target network. Furthermore, we can express the request propagation matrix D_{th}^{c} of threshold-based caching in the proposed model as

$$[\mathrm{D}_{th}^{c \in \mathrm{C}_o}]_{m,n} := \begin{cases} 1 & \forall m = parent\_node(n) \\ & \wedge \sum_k^N \lambda_{(n,k)}^c[s] < Th_n^o \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

C_o is the set of content items having origin o.

In this paper, we propose an approximation method using TTL of threshold-based caching because TTL-based caching can realize a more simple cache management by just updating the TTL counter of content without having to measure the request rates of content like in threshold-based caching.

We derive the approximation method using TTL of threshold-based caching as follows.

$$TTL_i^o = \frac{1}{Th_i^o}, \forall i, o \tag{11}$$

Here, TTL_i^o is set to *CR*$_i$ and defined as a different value for each origin site o of content. In Figure 11, we evaluate the cache hit ratio of content at a *CR* for threshold-based caching and the energy efficient TTL-based caching. As a result, we see

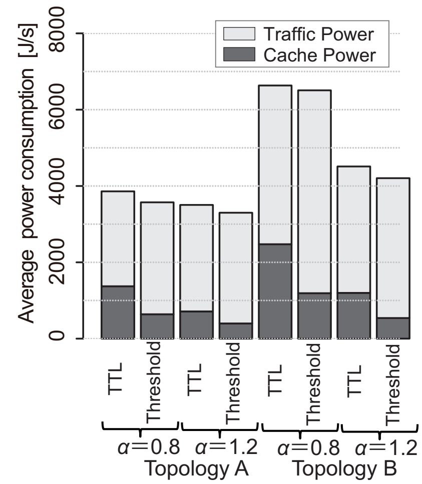


Figure 13: Power consumption for energy efficient TTL-based caching and threshold-based caching

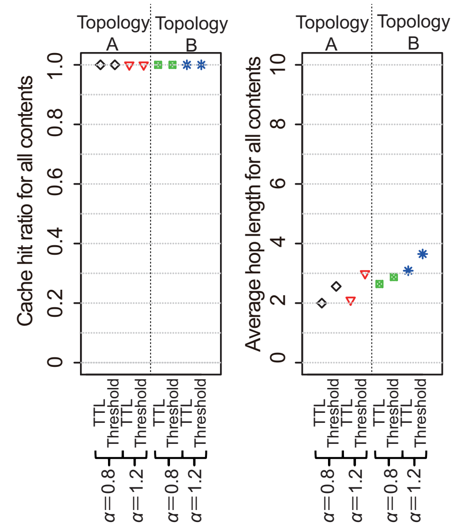


Figure 14: Cache performance for energy efficient TTL-based caching and threshold-based caching

that the cache hit ratio in the energy efficient TTL can provide similar characteristics to that in threshold-based caching.

Using (1), we can derive the request propagation matrix $\mathrm{D}_{ttl}^{c \in \mathrm{C}_o}$ using the energy efficient TTL as

$$[\mathrm{D}_{ttl}^{c \in \mathrm{C}_o}]_{m,n} := \begin{cases} 1 - \boldsymbol{f}(\sum_k^N \lambda_{(n,k)}^c[s], \frac{1}{Th_n^o}), \\ \qquad \forall m = parent\_node(n) \\ 0 \quad \text{otherwise} \end{cases}. \tag{12}$$

In these cache mechanisms, the threshold Th_o^o and the TTL $\frac{1}{Th_o^o}$ for content having origin o at *CR*$_o$ are defined as 0 and ∞, respectively. Therefore, all content items are always cached in the network unless memory overflow occurs in each *CR*.

Here, we demonstrate the effectiveness of the energy efficient TTL based on the same conditions as in Section V. Figure 12 shows the TTL values derived by (11). In these

 ISBN: 978-1-61208-318-6

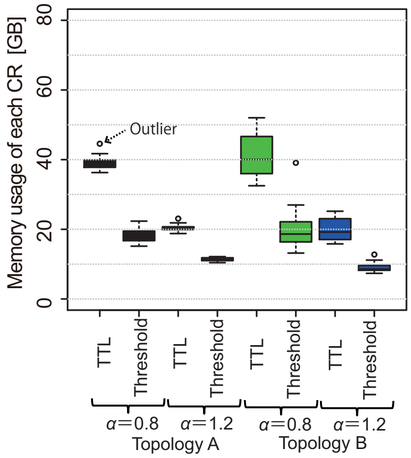


Figure 15: Box plot of memory usage for TTL-based caching and threshold-based caching

results, TTL_o^o at CR_o in origin site o is infinite and the other TTLs are derived as different values for each target network.

In Figure 13 and Figure 14, we compare the total power consumption and the cache performance for two mechanisms using the energy efficient TTLs and thresholds of request rates, respectively. In these results, the total power consumption in the energy efficient TTL-based caching is near to that in threshold-based caching. Moreover, the cache hit ratio is always 100% because TTL_o^o and Th_o^o $(\forall o)$ are infinite and 0. The average hop length in TTL-based caching is slightly smaller than that in threshold-based caching because the memory usage in TTL-based caching is larger than that in threshold-based caching as shown in Figure 15.

VII. Conclusion

We proposed an analytical model to evaluate the cache characteristics of a distributed cache system like CCN. The proposed model is expressed by iterative matrix equations and can evaluate the impact of TTL-based caching on network resources and cache performance on multiplexed caching hierarchies. In the evaluations, we verified the validity of cache characteristics estimated by the proposed model under the assumption that content requests are generated at an independent and identically distributed exponential interval and analyzed the impact on memory usage, power consumption, cache hit ratio, and average hop length when changing the TTL value of content. Furthermore, we introduced the energy efficient TTL to reduce the power consumption of the network and evaluated its effectiveness. Based on the proposed model, we showed that the energy efficient TTL-based caching can achieve a similar power consumption like threshold-based caching that searches for the most energy efficient cache locations and can realize shorter hop length than threshold-based caching.

As future work, we plan on enhancing the model in consideration of the limit of memory size and a different arrival process of content requests. Furthermore, we will study memory control mechanisms based on the theoretical model of TTL-based caching.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content", in Proc. of CoNEXT'09, Rome, Italy, December, 2009, pp. 1–12.

[2] I. Baev, R. Rajaraman, and C. Swamy,"Approximation Algorithms for Data Placement in Arbitrary Networks", in Proc. of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, DC, USA, January, 2001, pp. 661–670.

[3] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the Placement of Web Server Replicas", in Proc. of INFOCOM, Anchorage, AK, USA, April, 2001, pp. 1587–1596.

[4] Z. Drezner, "Facility Location: A Survey of Applications and Methods", Springer, Berlin, 1995.

[5] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks", in Proc. of INFOCOM'10, San Diego, CA, USA, March, 2010, pp. 1–9.

[6] K. Guan, G. Atkinson, and D. C. Kilper, "On the Energy Efficiency of Content Delivery Architectures", in Proc. of the 4th IEEE International Conference on Communications (ICC) Workshop on Green Communications, Kyoto, Japan, June, 2011, pp. 1–6.

[7] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental Evaluation of Memory Management in Content-Centric Networking", in Proc. of IEEE International Conference on Communications (ICC 2011), Kyoto, Japan, June, 2011, pp.1–6.

[8] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: modeling, design and experimental results", IEEE JSAC, 20(7), 2002, pp. 1305–1314.

[9] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework For Network Devices", in Proc. of NETWORKING'09, vol. 5550, Aachen, Germany, May, 2009, pp. 795–808.

[10] T. Harder, V. Hudlet, Y. Ou, and D. Schall, "Energy Efficiency is not Enough, Energy Proportionality is Needed!", in Proc. of DASFAA'11, Hong Kong, China, April, 2011, pp. 226-239.

[11] U. Lee, I. Rimac, D. C. Kilper, and V. Hilt, "Toward energy-efficient content dissemination", IEEE Network, vol. 25, no. 2, 2011, pp. 14–19.

[12] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network", in IEEE NOMEN'12, Workshop on Emerging Design Choices in Name-Oriented Networking, Orlando, Florida, USA, March, 2012, pp. 310–315.

[13] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing", Technical Report, Telecom Paris Tech,. 2011.

[14] S. Imai, K. Leibnitz, and M. Murata, "Energy Efficient Content Locations for In-Network Cachingin Proc. of APCC'12, Jeju, Korea, October, 2012, pp. 554 – 559.

[15] S. Imai, K. Leibnitz, and M. Murata, "Energy-Aware Cache Management for Content-Centric Networking", in Proc. of First International Workshop on Energy-Aware Systems, Communications and Security, Barcelona, Spain, March, 2013, pp. 1623 – 1629.