Modeling the Sojourn Time of Items for In-Network Cache Based on LRU Policy

LIU Jiang, WANG Guoqing, HUANG Tao, CHEN Jianya, LIU Yunjie

State Key Laboratory of Networking and Switching Technology. Beijing University of Posts and Telecommunications, Beijing 100876, China.

Abstract: To reduce network redundancy, in-network caching is considered in many future Internet architectures, such as Information Centric Networking. In in-network caching system, the item sojourn time of LRU (Least Recently Used) replacement policy is an important issue for two reasons: firstly, LRU is one of the most common used cache policy; secondly, item sojourn time is positively correlated to the hit probability, so this metric parameter could be useful to design the caching system. However, to the best of our knowledge, the sojourn time hasn't been studied theoretically so far. In this paper, we first model the LRU cache policy by Markov chain. Then an approximate closed-form expression of the item expectation sojourn time is provided through the theory of stochastic service system, which is a function of the item request rates and cache size. Finally, extensive simulation results are illustrated to show that the expression is a good approximation of the item sojourn time.

Keywords: sojourn time; Markov chain; LRU; steady-state probability.

I. INTRODUCTION

In recent years, Internet has been experiencing a shift from the host-centric towards information-centric communication paradigm [1] in order to overcome the shortcomings of the IP-based Internet. Many new architectures have been proposed, such as DONA (Data Oriented Network Architecture)[2], CCN/NDN(Content-Centric Networking/ Named Data Networking)[3] and ICN (Information-Centric Networking)[4]. Although these architectures differ with respect to the specific approaches, they all have supported in-network caching, as caching can reduce the load on access links and shorten the access time to the requested contents. For example, Content Store (CS) is added in CCN router, and the requested contents will be cached in the CS by using LRU cache policy. Hence, many works [5-13] have been done for the relationship between cache policies and network performance in ICN. The authors in [5-6] have studied the tradeoff between bandwidth and storage by deducing the content miss probabilities and the round trip time. Then the performance of storage management in CCN has been evaluated by authors in [7]. In [8], the authors have modeled the LRU policy at one cache and then extended to several caches to examining the caching dynamics. Considering the universal caching in the future Internet architectures, authors of [9-13] have proposed new cache policies by directly or indirectly cooperating in a cache group to improve the cumulative hit rate.

As one of the most basic cache algorithms, LRU replacement cache policy has been employed in CCN. Most of the researches about

The paper models the LRU cache policy by Markov chain and provides an approximate closed-form expression of the item expectation sojourn time through the theory of stochastic service system, which is a function of the item request rates and cache size.

CCN caching are based on LRU policy [5-9]. The theory study of LRU has a long history. The earliest analytical model about LRU is proposed by authors of [14], who has studied the relationship between LRU miss probability and the search cost of MTF (Move To Front). By studying the connection between the MTF and the coupon collectors' problem, authors of [15] has derived the n-step transition probabilities for the underlying Markov chain. In [16], an analytical model has been proposed to get the asymptotic expression of the miss probability in heavy and light tail distribution of the item popularity with cache size tends to infinite. Under the assumption of the requests' arriving at Poisson moments, the authors have studied the miss probability by approximating the miss sequence of the requests at one cache as an independent renewal process [17].

Although many works have been done to study the closed-form expression of hit/miss probability based on LRU policy by keeping several parameters fixed (i.e. the item size and popularity), the cache size tending to infinite is assumed, which may not be reasonable in practical network. Therefore, different from previous works, we aim to analyze the sojourn time, which is defined as the time that the item survives in the cache from it being cached to being removed, and then derive an analytic approximate expression of the item sojourn time when cache size is smaller than the kinds of items in this paper.

The sojourn time is important in caching policy study. Firstly, it can be used to design TTL (Time To Live) cache policy with a certain hit probability as TTL has positive correlation with the hit probability. Secondly, it can also guide the design of the item placement and replacement policies to increase the cumulative hit probability in the caches, such as in [11, 18]. Thirdly, the miss sequence of item k is an analogy of a renewal process which is started once the item k misses. Then the sojourn time is like the on phase in the renewal process, while the time from item k removed to the next miss is the off phase. According to the renewal theory, we can get

$$\frac{E\omega_k}{E\omega_k + E\delta_k} = \lim_{t\to\infty} P_k(t) \tag{1}$$

where $E\omega_k$ denotes the expectation sojourn time of item k, $E\delta_k$ denotes the expectation time from item k removed to the next miss, and $P_k(t)$ denotes the probability of item in the cache at time t, thus $\lim_{t\to\infty} P_k(t)$ denotes the hit probability of item k. Thus, the research about the expectation sojourn time may provide a new way to get the closed-form expression of hit probability.

The main contributions of this paper are as follows: First of all, assuming the arriving requests follow Poisson distribution, we use a Markov chain to model the behavior of the item in the cache based on LRU policy. Secondly, we get the approximate steady-state probabilities. Thirdly, Due to the item sojourn time is an analogy of the waiting time in stochastic service system, we derive the closed-form expression of item sojourn time by solving the stochastic relationship between the kinds and the number of item requests, which is a function of the item popularity and cache size. Finally, extensive simulation results are illustrated to demonstrate that the expressions is a good approximation of the item sojourn time based on LRU policy, especially when the cache has small cache size and the items have uniform popularity.

The rest of the paper is organized as follows. In Section 2, a model description of the work process of LRU is given. We describe the details of our derivation in Section 3. In Section 4, we give simulation results and analyze the relationship of the item popularity, the cache size, and the sojourn time. We conclude the paper and give insights for future work in Section 5.

II. Model Description

Consider a finite set of items k=1,2,...,M with the same size (For the items with different sizes, they will be split into chunks with the same size in CCN), and a cache with capacity of x items, $x<M$ (when $x\geq M$, the sojourn time of each item is infinity, and the hit probabili-

ty equals 1). The requests for items arrive as Poisson distribution, and requests for item k with the arrival rate of λ_k, k=1,2,...,M. We assume the request rates are non-increasing as it is reasonable after permutating. The notations and meanings used in the paper are shown in Table 1. The LRU policy is described as follows. If at time t the requested item k is at the l-th position of the list in the cache, then it is brought to the first position and items in positions 1,2,...,l−1 are moved one position down. This case is called hit. If the requested item k is not in the cache, then it is replicated and cached at the top of the cache. Items in the cache are moved one position down, and the bottom item is moved out. This case is called miss. When the item k is in the cache, the other items in the cache can be divided into two classes: class A_k includes the items cached above the item k and class B_k includes the items cached below the item k. Clearly, the requests for the class A_k will not change the item k's position, while the requests for the class B_k will make item k move one position down or out. When item k is at the j-th position, we assume the class A_k has a request rate of a_k^j, and the class B_k has a request rate of $b_k^j = \lambda - \lambda_k - a_k^j$. We can get the value range of a_k^j as following.

$$\min_{l=1,\cdots,x}(a_k^l) = a_k^1 = 0, \tag{2}$$

$$\max_{l=1,\cdots,x}(a_k^l) = \max(a_k^x) = \begin{cases} \sum_{j=1}^{x} \lambda_j - \lambda_k, k < x, \\ \sum_{j=1}^{x-1} \lambda_j, k \geqslant x. \end{cases} \tag{3}$$

When the item k is at the position 1, the class A_k is empty which is the expression (2). When the item k is at the position x, the class A_k includes (x−1) items above the item k which is the maximum set of class A_k. In the maximum set, the $\max(a_k^x)$ is the sum of the request rates of the most (x−1) popular items without item k which is shown in expression (3).

III. Deducing the Expression of the Sojourn Time

In this section, we model the work process of LRU replacement policy as homogeneous Markov chain for the requests following Poisson distribution. The steady-state probability of each state is derived by the balance equation. Then we analyze the probability of the item k's waiting time (waiting to be replaced) under the condition of its position. The approximate expression of the average sojourn time is derived finally. One of the most difficult problems is to deal with the stochastic relationship between the kinds and the number of item requests, as the requests for the same item are regard as one kind. Thus, we introduce two parameters σ&ρ to solve the problem. The details are described following.

We model the process as a homogeneous Markov chain, where the state of the chain represents the exact position that the item currently occupies in the cache. Number the Markov chain states of item k as $\varphi = \{0, 1, \cdots, x\}$. State 0 denotes the item is out of the cache, and state j denotes the item is at the j-th position in the cache. As shown in Fig.1, the transfer rate from state $j (j = 1, 2, \cdots, x)$ to the state (j−1) is the rate $b_k^j = \lambda - \lambda_k - a_k^j$ (where state (x−1) denotes state 0 for describing conveniently), and each other state can transfer to the state 1 with rate λ_k.

The transfer rate matrix is

Table I *Notations and meanings*

Notation	Meaning
M	number of different content items
x	item cache size [item]
λ	total item request rate [req/s]
λ_k	request rate of item k
$\vec{\lambda}$	row vector of $(\lambda_1, \lambda_2, \cdots, \lambda_M)$
A_k	the set of items above item k in the cache
B_k	the set of items above item k in the cache
a_k^j	request rate of items in class A_k when item k at position j
b_k^j	request rate of items in class B_k when item k at position j
ST_k	The expectation sojourn time of item k
φ	state space of the Markov chain

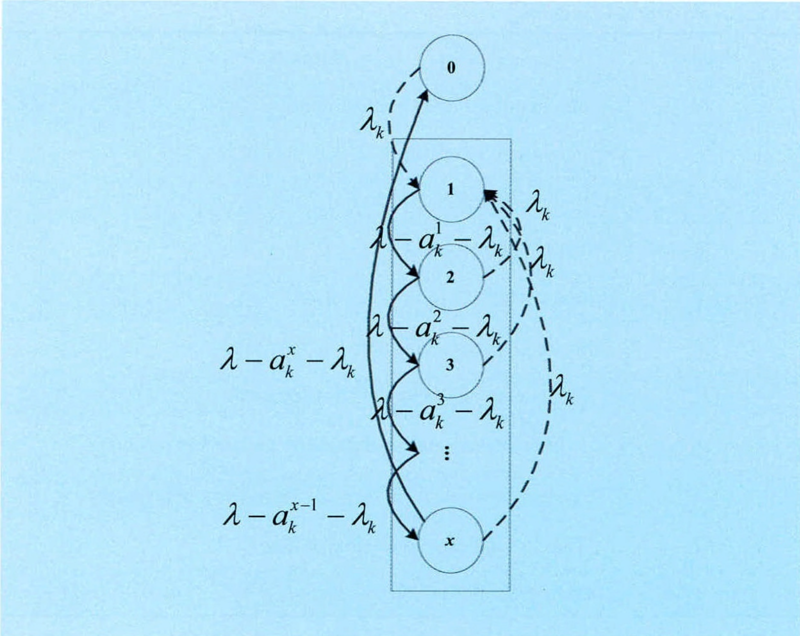


Fig.1 *The state transition diagram*

$$\tilde{Q} = \begin{pmatrix} -\lambda_k & \lambda_k & 0 & 0 & \cdots & 0 & 0 \\ 0 & -b_k^1 & b_k^1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_k & -\beta_k^2 & b_k^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \lambda_k & 0 & 0 & \cdots & -\beta_k^{x-1} & b_k^{x-1} \\ b_k^x & \lambda_k & 0 & 0 & \cdots & & -\beta_k^x \end{pmatrix} \tag{4}$$

where $b_k^j = \lambda - \lambda_k - a_k^j$, $\beta_k^j = b_k^j + \lambda_k$.

We resolve the balance equations

$$\begin{cases} \vec{P}\tilde{Q} = 0, \\ \vec{P}\vec{1}' = 1. \end{cases} \tag{5}$$

where $\vec{P}$ is an $(x-1)$ dimensions row vector of $(P_0, P_1, \cdots, P_x)$ which denotes the steady-state probability of item at state 0,1,...,x respectively, and $\vec{1} = (1, 1, \cdots, 1)$ is an $(x-1)$ dimension row vector. $\vec{1}'$ denotes the transposition of the vector $\vec{1}$.

Then we get

$$P_1 = \frac{\lambda_k}{b_k^1 + \lambda_k} \tag{6}$$

$$P_j = \frac{\prod_{l=1}^{j-1} b_k^l}{\prod_{n=2}^{j} \beta_k^n} P_1, j = 2, \cdots, x. \tag{7}$$

$$P_0 = \frac{\prod_{l=1}^{x} b_k^l}{\lambda_k \prod_{n=2}^{x} \beta_k^n} P_1. \tag{8}$$

Where P_j presents the steady-state probability of item k at the state j.

From the expressions (2-3), we get

$$\max_{l=1,\cdots,x}(b_k^l) = \lambda - \lambda_k, \tag{9}$$

$$\min_{l=1,\cdots,x}(b_k^l) = \min(b_k^x) = \begin{cases} \lambda - \sum_{j=1}^{x} \lambda_j, & k < x, \\ \lambda - \lambda_k - \sum_{j=1}^{x-1} \lambda_j, & k \geqslant x. \end{cases} \tag{10}$$

When $x \ll M$, the interval between $\min_l(b_k^l)$ and $\max_l(b_k^l)$ is smaller. Thus we can assume $b_k^j = b_k$, $j = 1, 2, \cdots, x$, where $b_k \in [\min_l(b_k^l), \max_l(b_k^l)]$, $\beta_k^j = b_k + \lambda_k$. Then, the expressions (6-8) can be shown as follows:

$$P_l = \frac{\lambda_k}{b_k}\left(\frac{b_k}{b_k + \lambda_k}\right)^l, l = 1, 2, \cdots, x. \tag{11}$$

$$P_0 = \left(\frac{b_k}{b_k + \lambda_k}\right)^x. \tag{12}$$

We deduce the expectation sojourn time of the item k in the cache following.

Proposition 1 Given a Poisson distribution request arrival process with intensity λ, the requests of item k with rate λ_k. Items have the same size, and $0<x<M$ is the cache size in number of items, then the expectation sojourn time of the item k in the cache is given by

$$\begin{aligned} ST_k = {} & \frac{1}{\lambda_k}\left(1 - \left(\frac{b_k}{b_k + \lambda_k}\right)^{\sigma+x}\right) - \frac{x}{b_k + \lambda_k}\left(\frac{b_k}{b_k + \lambda_k}\right)^{\sigma+x} \\ & + \frac{\rho + x}{b_k}\left(1 - \left(\frac{b_k}{b_k + \lambda_k}\right)^x\right) \\ & - \frac{1}{\lambda_k}\left(1 - \left(\frac{b_k}{b_k + \lambda_k}\right)^x\right)\left(1 - \left(\frac{b_k}{b_k + \lambda_k}\right)^{\rho+x}\right), x \ll M. \end{aligned} \tag{13}$$

Proof: Let ω_k be the waiting time of the item k in the cache waiting to be replaced, i.e. the sojourn time of item k. $\Pr_j\{\omega_k > t\}$ denotes the probability of the waiting time greater than t under the condition that the item k is at the state j, then we have

$$\Pr\{\omega_k > t\} = \sum_{j=1}^{x} P_j \Pr_j\{\omega_k > t\} \tag{14}$$

To sort the requests for different items, we define $R^k(t)$ as the number of requests for the item k in the open interval $(0,t)$, $R^k(t) \sim \text{poisson}(\lambda_k)$, and $I^k(t) = 1_{\{R^k(t)>0\}}$ the Bernoulli variable associated to the event that arrives at least one request for item k in the open interval $(0,t)$ with $P[I^k(t) = 1] = 1 - e^{-\lambda_k t}$. $S^{k,\bar{\lambda}_k}(t)$ denotes the kinds of item requests in

$(0,t)$ without item k and items in class A_k,

$$S^{\bar{k},\bar{A}_k}(t) = \sum\nolimits_{l=1, l \notin k \cup A_k}^{M} I^l(t) \qquad (15)$$

When the item k is at the state j, there are $(j-1)$ different items above the item k, the requests of which will not change the position of item k. So the waiting time of item k greater than t will happen in two cases: there are no requests for item k during the time interval $(0,t)$, and the kinds of requests for items in class B_k are no more than $(x-j)$; there are requests for item k during the time interval $(0,t)$, the kinds of requests for items in class B_k are no more than $(x-1)$ from the last request for item k to the time t. Thus, we have

$$\begin{aligned}\Pr_j\{\omega_k > t\} &= \Pr_j\{\omega_k > t | I^k(t) = 0\} \Pr\{I^k(t) = 0\} \\ &+ \Pr_j\{\omega_k > t | I^k(t) = 1\} \Pr\{I^k(t) = 1\} \\ &= \Pr[S^{\bar{k},\bar{A}_k}(t) \leqslant x - j] e^{-\lambda_k t} \\ &+ \Pr[S^{\bar{k},\bar{A}_k}(t) \leqslant \xi + (x-1)](1 - e^{-\lambda_k t}) \\ &= \sum_{l=0}^{x-j} \sum_{\vec{1}\vec{v}' = l, v_k = 0} e^{-\vec{\lambda}(\vec{1}-\vec{v})' t} \prod_{i=1, v_i=1}^{M} (1 - e^{-\lambda_i v_i t}) \\ &+ \sum_{l=0}^{\xi+(x-1)} \sum_{\vec{1}\vec{v}' = l+1, v_k = 1} e^{-\vec{\lambda}(\vec{1}-\vec{v})' t} \prod_{i=1, v_i=1}^{M} (1 - e^{-\lambda_i v_i t})\end{aligned} \qquad (16)$$

where ξ is a nonnegative number denoting the kinds of requests for the items in the complement of $k \cup A_k$ before the last request for item k during $(0,t)$. $\vec{1} = (1, 1, \cdots, 1)$ is an M dimension row vector. $\vec{v}$ is an M-dimension row vector with components equaling 0,1 and empty, and for $m \notin A_k$, $v_m = 1$ denotes there are requests for item m, while $v_m = 0$ denotes no requests for item m. For $m \in A_k$, v_m denotes empty and does nothing just occupying a position in the vector. We define the operations of the empty in $\vec{v}$ as following rules: the empty equals zero in the operations of multiplication and division; Any number adds or minus the empty is still empty. For example: when $l = 1, k = 2, M = 3, \vec{v} = (\text{empty}, 0, 1)$, then

$$e^{-\vec{\lambda}(1-\vec{v})' t} = e^{-\vec{\lambda}(\text{empty},0,1)' t} = e^{-(0+\lambda_2+0)t} = e^{\lambda_2 t} \qquad (17)$$

Thus, $\vec{1}\vec{v}' = l, v_k = 0$ denotes the kinds of item requests without item k and there are l items in class A_k arriving. Equation (16) has high computational complexity. The class A_k is dynamic change with the position transfer of item k, which contributes part of the complexity of (16). When the number of items in class A_k is much smaller than that of the total items M (considering each A_k of all states, as $A_k < x \ll M$), then the request rate of complement of $k \cup A_k$ approximates b_k. The other part of the complexity of expression (16) comes from the permutation of the items with different request rates, we introduce two nonnegative parameters $\sigma \& \rho$ to solve the problem. Thus, the approximate expression is proposed following.

$$\begin{aligned}\Pr_j\{\omega_k > t\} &\approx e^{-\lambda_k t} \sum_{i=0}^{\sigma+x-j} e^{-b_k t} \frac{(b_k t)^i}{i!} \\ &+ (1 - e^{-\lambda_k t}) \sum_{i=0}^{\rho+x-1} e^{-b_k t} \frac{(b_k t)^i}{i!}, x \ll M\end{aligned} \qquad (18)$$

where $\sigma>0$, $\rho>0$. Put (18) into (14), we have

$$\begin{aligned}\Pr\{\omega_k > t\} &= e^{-\lambda_k t} \sum_{j=1}^{x} P_j \sum_{i=0}^{\sigma+x-j} e^{-b_k t} \frac{(b_k t)^i}{i!} \\ &+ (1 - e^{-\lambda_k t}) \sum_{j=1}^{x} P_j \sum_{i=0}^{\rho+x-1} e^{-b_k t} \frac{(b_k t)^i}{i!} \\ &= \sum_{i=0}^{\sigma+x-j} (1 - (\frac{b_k}{b_k + \lambda_k})^{x-i}) e^{-\lambda_k t} e^{-b_k t} \frac{(b_k t)^i}{i!} \\ &+ (1 - (\frac{b_k}{b_k + \lambda_k})^x) \sum_{i=0}^{\rho+x-1} (1 - e^{-\lambda_k t}) e^{-\lambda_k t} e^{-b_k t} \frac{(b_k t)^i}{i!}\end{aligned} \qquad (19)$$

Then we get the expectation sojourn time of item k,

$$\begin{aligned}ST_k &= E\omega_k \\ &= \int_0^\infty t d(1 - \Pr\{\omega_k > t\}) \\ &= \frac{1}{\lambda_k}(1 - (\frac{b_k}{b_k + \lambda_k})^{\sigma+x}) - \frac{x}{b_k + \lambda_k}(\frac{b_k}{b_k + \lambda_k})^{\sigma+x} \\ &+ \frac{\rho + x}{b_k}(1 - (\frac{b_k}{b_k + \lambda_k})^x) \\ &- \frac{1}{\lambda_k}(1 - (\frac{b_k}{b_k + \lambda_k})^x)(1 - (\frac{b_k}{b_k + \lambda_k})^{\rho+x}), x \ll M.\end{aligned} \qquad (20)$$

As $0 < \frac{b_k}{b_k + \lambda_k} < 1$, and $\sigma > 0$, $\rho>0$, we ignore the terms containing $(\frac{b_k}{b_k + \lambda_k})^x, (\frac{b_k}{b_k + \lambda_k})^{x+\sigma}$ and $(\frac{b_k}{b_k + \lambda_k})^{x+\rho}$ in (20). Thus, the sojourn time of item k is approximate as $\frac{\rho + x}{b_k}$. From the equations (9-10), b_k has negative correlation with λ_k. Thus

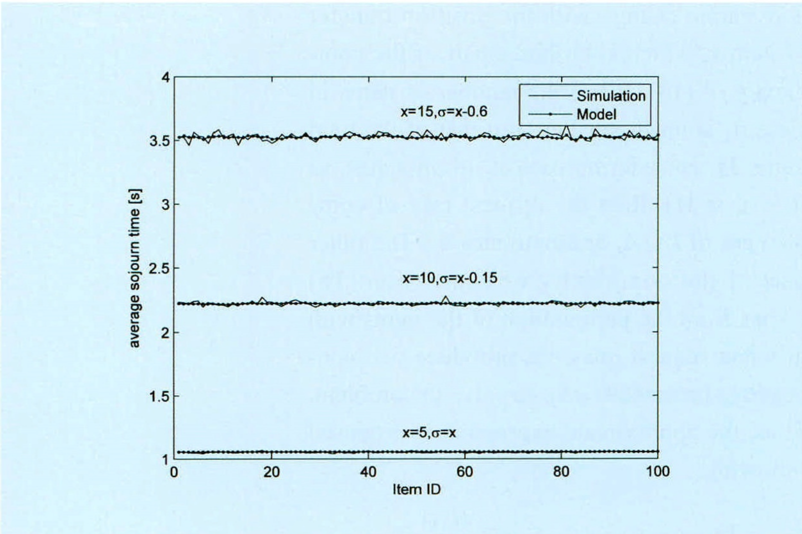


Fig.2 *The sojourn time of each item for different cache sizes with uniform distribution popularity.*

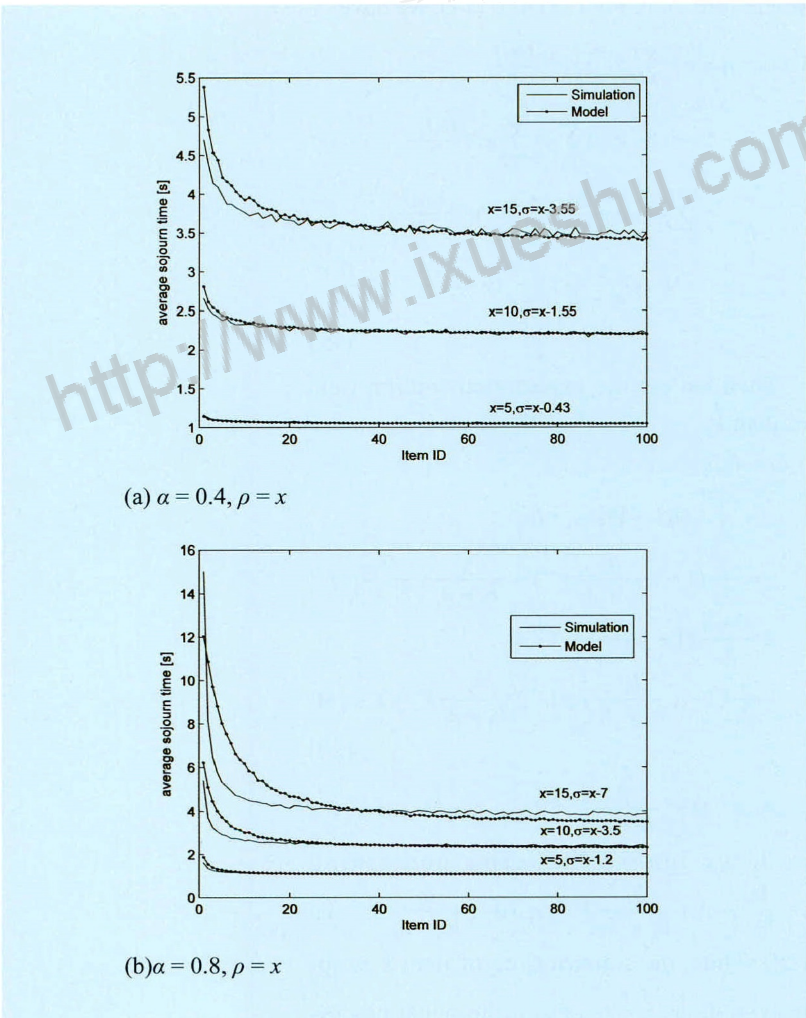


Fig.3 *The sojourn time of each item for different cache sizes with zipf distribution popularity.*

the sojourn time is larger with the larger cache size and the item popularity λ_k.

IV. Simulation and Analysis

This section gathers numerical results of sojourn time obtained by the expression and simulation to examine the validity of the Proposition 1. We use matlab to simulate the process of the LRU policy at one cache to get the statistic average item sojourn time. The cache is empty initially and the requests received by the caches are Poisson distribution with the intensity λ=5req/s, and M=100. The simulation time is 20000 seconds, i.e. 100000 requests arriving at the cache. The simulation results are obtained by

$$ST_k = \frac{\sum_{l=1}^{N}(T^l_{\text{evict }k} - T^l_{\text{insert }k})}{N} \tag{21}$$

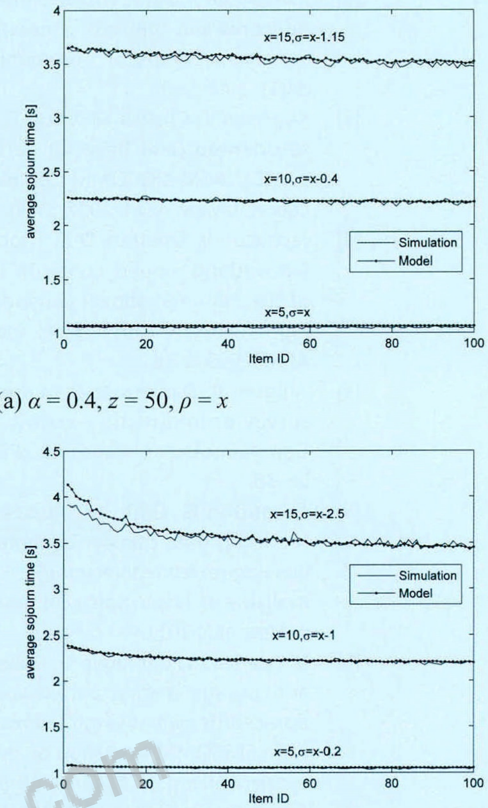
where $T^l_{\text{evict }k}$ denotes the l-th time of item k replicated in the cache, and $T^l_{\text{insert }k}$ denotes the l-th time of item k evicted from the cache. N is the total times of the item k replicated in the cache during the simulation time.

We consider the item popularity follows a uniform distribution firstly. In the cases of the cache size x=5, 10&15, Fig. 2 shows that the model results coincide with the simulation results when the parameter ρ=x and σ=x,(x−0.15)\&(x−0.6) respectively. All the items have almost the same sojourn time at the cache with same cache sizes as they all have the same popularity. Clearly, the item sojourn time is x/λ if each item is requested only one time. The item requests arrive randomly and may arrive several times during a short time, thus the item sojourn time will no less than x/λ. This is shown in the Fig. 2 and the sojourn time has positive correlation with the cache size which is within our intuition.

In Fig. 3, we assume the popularity of item k is q_k following a zipf law popularity distribution, $q_k = c/k^{\alpha}$ with parameter α = 0.4 (Fig. 3(a)) and α = 0.8 (Fig. 3(b)), as the smaller of α, the more even of the items popularity. The figures show that the model results are more precise when the cache size is smaller and the popularity with more even distribution. The sojourn time has positive correlation with the item popularity which is also shown in the Fig. 3. These results are also proved by the Fig. 4, which shows the item popularity following a Mandelbrot-zipf (Mzipf for short) popularity distribution, $q_k = c/(z+k)^{\alpha}$ with parameter α = 0.4, plateau parameter z = 50 (Fig. 4(a)) and z = 5 (Fig. 4(b)) respectively.

Three conclusions can be drawn from the simulation. First, the sojourn time has positive correlation with the cache size and item popularity. Second, the expression of sojourn time are more precise when the cache size is smaller and the popularity with more even distribution. Third, the parameter ρ=x in the distributions simulated, and the difference between x and σ is smaller as the more even popularity distribution or the smaller cache size.

For the designer of TTL cache policy, the most important parameter to be set is the TTL which has positive correlation with the hit probability in a certain range. However, the cache size is always much smaller than the total items, TTL is not the-larger-the-better. Item TTL implies the cache resource allocated to the item, so it is influenced by the cache size and item popularity. The expectation sojourn time employing LRU cache policy provides a good reference to setting TTL in TTL cache policy. We get the hit probabilities of TTL cache policy with TTL_k=ST_k in the Table II when the item popularity follows zipf law distribution with α=0.8.



(a) $\alpha = 0.4, z = 50, \rho = x$

(b) $\alpha = 0.4, z = 5, \rho = x$

Fig.4 *The sojourn time of each item for different cache sizes with M-zipf distribution popularity.*

Table II *Hit probabilities of TTL cache policy designed by sojourn time.*

Cache size	5	10	15
Hit probability	5.114%	9.164%	11.395%

V. Conclusion

The Proposition 1 reflects that the item sojourn time has positive correlation with the cache size and item popularity, thus, the item popularity can be sorted when we get the items sojourn time at a cache employing LRU policy. As a direct influence factor of the hit probability in TTL cache policy, sojourn time can be used to set the preferable items TTL with a better hit probability. Furthermore, as the caches distributed in the content-centric networking, the sojourn time is an index to design cooperate cache policy. Finally, the closed-form expression of item expectation sojourn time may provide a new way to get the expression of hit probability.

There are still some future work to be done. The distributions of the parameters σ and ρ are waiting to be researched deeply. New methods need to be found to estimate the sojourn time and the hit probability. Considering that the future Internet makes cache universal, new storage management policies need to be proposed and evaluated.

References

[1] Paul S, Pan J, Jain R. Architectures for the future networks and the next generation Internet: A survey[J]. Computer Communications, 2011, 34(1): 2-42.

[2] Koponen T, Chawla M, Chun B G, et al. A data-oriented (and beyond) network architecture[C]//ACM SIGCOMM Computer Communication Review. ACM, 2007, 37(4): 181-192.

[3] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content[C]//Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009: 1-12.

[4] Ahlgren B, Dannewitz C, Imbrenda C, et al. A survey of information-centric networking[J]. Communications Magazine, IEEE, 2012, 50(7): 26-36.

[5] Carofiglio G, Gallo M, Muscariello L, et al. Modeling data transfer in content centric networking (extended version)[R]. Research report, available at http://perso. rd. francetelecom. fr/muscariello, 2011.

[6] Muscariello L, Carofiglio G, Gallo M. Bandwidth and storage sharing performance in information centric networking[C]//Proceedings of the ACM SIGCOMM workshop on Information-centric networking. ACM, 2011: 26-31.

[7] Carofiglio G, Gehlen V, Perino D. Experimental evaluation of memory management in content-centric networking[C]//Communications (ICC), 2011 IEEE International Conference on. IEEE, 2011: 1-6.

[8] Psaras I, Clegg R G, Landa R, et al. Modelling and evaluation of CCN-caching trees[M]//NETWORKING 2011. Springer Berlin Heidelberg, 2011: 78-91.

[9] Chai W K, He D, Psaras I, et al. Cache "less for more" in information-centric networks[M]//NETWORKING 2012. Springer Berlin Heidelberg, 2012: 27-40.

[10] Li Z, Simon G. Time-shifted tv in content centric networks: the case for cooperative in-network caching[C]//Communications (ICC), 2011 IEEE International Conference on. IEEE, 2011: 1-6.

[11] Ming Z, Xu M, Wang D. Age-based cooperative caching in information-centric networks[C]//Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 2012: 268-273..

[12] Cho K, Lee M, Park K, et al. WAVE: Popularity-based and collaborative in-network caching for content-oriented networks[C]//Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 2012: 316-321.

[13] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks[C]//Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM, 2012: 55-60.

[14] Palmboom A. Analysis of Demand Paging Algorithms[M]//GI. Gesellschaft für Informatik eV 2. Jahrestagung. Springer Berlin Heidelberg, 1973: 274-283.

[15] Phatarfod R M. On the transition probabilities of the move-to-front scheme[J]. Journal of applied probability, 1994: 570-574.

[16] Jelenković P R. Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities[J]. The Annals of Applied Probability, 1999, 9(2): 430-464.

[17] Jelenković P R, Kang X. Characterizing the miss sequence of the lru cache[J]. ACM SIGMETRICS Performance Evaluation Review, 2008, 36(2): 119-121.

[18] Ramaswamy L, Liu L. An expiration age-based document placement scheme for cooperative web caching[J]. Knowledge and Data Engineering, IEEE Transactions on, 2004, 16(5): 585-600.

Biographies

LIU Jiang, born in 1982, Ph.D., lecturer. His research interests include routing and forwarding, content-centric networking.

WANG Guoqing, born in 1986, Ph.D. candidate. Her research interests include resource distribution and content-centric networking. E-mail: gqwang @bupt.edu.cn

HUANG Tao, born in 1980, Ph.D., associate professor. His research interests include routing and forwarding, content-centric networking. E-mail: htao@bupt.edu.cn

CHEN Jianya, born in 1953, professor. His research interests include wireless access network and next generation internet.

LIU Yunjie, born in 1943, academician of Chinese academy of engineering, Ph.D. supervisor. His research interests include next generation internet and network integration.