A Markov Chain-based Approximation of CCN Caching Systems

Hamza Ammar, Soraya Chellouche, Yassine Hadjadj-Aoul

A Markov Chain-based Approximation of CCN Caching Systems

Hamza Ben Ammar
IRISA Lab, University of Rennes 1
France
Email: hamza.ben-ammar@irisa.fr

Soraya Ait Chellouche
IRISA Lab, University of Rennes 1
France
Email: soraya.ait-chellouche@irisa.fr

Yassine Hadjadj Aoul
IRISA Lab, University of Rennes 1
France
Email: yassine.hadjadj-aoul@irisa.fr

***Abstract*—To address the challenges raised by the Internet usage evolution over the last years, the Content-Centric Networking (CCN) has been proposed. One key feature provided by CCN to improve the efficiency of content delivery is the in-network caching, which has major impact on the system performance. In order to improve caching effectiveness in such systems, studying the functioning of CCN in-network storage is required. In this paper, we propose MACS, a Markov chain-based Approximation of CCN caching Systems. We start initially by modeling a single cache node. Afterwards, we extend our model to the case of multiple nodes. A closed-form expression is then derived to define the cache hit probability of each content in the caching system. We compared the results of MACS to those obtained with simulations. The conducted experiments show clearly the accuracy of our model in estimating the cache hit performance of the system.**

***Index Terms*—Content-Centric Networking, Caching, Markov Chain, Modeling.**

I. Introduction

The last few years witnessed a shift in the Internet usage that switched from a host-centric model to a content-centric approach, especially when dealing with content retrieval and data dissemination [1]. This evolution is mostly driven by the increased popularity of content-oriented services, e.g., Peer-to-Peer file sharing, Video on Demand, video/audio streaming and social networks where users focus more on contents and not on the physical locations from which contents can be retrieved. These new trends in the Internet usage have raised important challenges in the current IP-based infrastructures, especially in terms of content delivery [2].

To address these challenges, the Content-Centric Networking (CCN) [3] has been proposed, and emerged as one of the most interesting alternatives to the existing network architecture. Instead of the end-to-end principle, the CCN paradigm consists in redesigning the future Internet architecture by focusing on named data and leveraging new features, such as in-network caching, multipath connectivity and multicast data delivery. The content name in CCN is the only identifier of data, and the information exchanges based on establishing communication channels are abandoned.

The ubiquitous in-network caching represents certainly one of the most important features, which impacts directly data delivery performance. In fact, CCN nodes are equipped with content store modules and have the ability to cache the data that pass by them. Therefore, the end-users' requests (known as interests in CCN), that are routed toward the Content Providers' servers, can be satisfied by the cached data at the intermediate nodes.

In-network caching as provided by CCN has brought a renewed interest in developing efficient tools to study the caching performance of interconnected systems of caches. This will help, indeed, in giving guidance and offering insights on the behavior of a network of caches. Besides, this can be used to optimize the caching efficiency while increasing the overall system performance. However, evaluating the network caching performance is a difficult task due basically to the huge amount of available data objects and the size of the networks. Several approaches have been proposed over the last years in order to evaluate the performance of caching systems [4] [5] [6] [7] [8]. However, most of them suffer from their limited application scope as they can be applied only to a specific type and size of network topologies.

In this paper, we propose an analytical model based on Markov chains to estimate the cache hit probability under the popular Last Recently Used (LRU) replacement scheme for a system with multiple caching nodes. We considered a network where requests are sent independently through the shortest path to the nearest source, while retrieved contents are cached along the request's path. We first, proposed a model for a single caching node for which we derived the expression of the hit and miss rates. Then, we extend the model to the case of multiple caching nodes, in which the requests may come from other nodes due to a cache miss. Note that the proposed solution can deal with any type of topology or large-scale cache networks, and it can be extended to be applicable to other schemes of caching contents. The comparison of our model with the results obtained using a discrete event simulator, under different network topologies with various parameters' settings, shows that our proposal can estimate with high accuracy the cache hit of the system.

The rest of the paper is organized as follows. We present the related work in section II. Then, we describe our cache system model, named MACS, in section III. Afterwards, we expose the numerical results of our proposal in section IV and, finally, conclude and discuss future works in section V.

The research leading to these results was performed under the CAMION project, which has received funding from Eurostars.

II. Related Work

Many studies have been conducted to deal with the performance evaluation of caches and network of caches.

In [4], Dan and Towsley proposed an algorithm for predicting an approximate buffer hit probability under the LRU replacement policy. The complexity of their algorithm is $o(KB)$ where B is the size of the buffer, and K denotes the number of items having distinct access probabilities. While efficient and precise, the solution is limited to the case of a single cache.

Che et al. developed, in [5], an analytical modeling technique, which was further investigated in [8]. The solution allows identifying a characteristic time approximation for each item in the cache, which was used to get an equation that estimates the cache hit rate per content. Their proposed solution is, however, applicable only to hierarchical caching systems. Moreover, solving the equation for each content are not achievable in practice.

In [6], the authors presented an algorithm for approximating the rate of incoming miss streams for each file at every node in a network using LRU caches. Individual caches are evaluated using the miss probability of each cache, and the entire system's performance is measured in terms of the average number of hops a request traverses until the content is located. However, their proposal was evaluated in cache networks where the number of files available on the system is low.

Psaras et al. have proposed, in [7], a caching model through a Markov chain to estimate the proportion of time a given piece of content is cached in the case of a single router. By using some approximations, they extend their model to cover the case of multiple routers. They evaluated their analytical model through simulation, but it was only done on two simple tree topology that contain just few nodes.

In this paper, we develop an analytical model having a polynomial complexity to address the performance evaluation of a multi-cache system under the LRU replacement policy. Next, we present our approach with all the needed information on the system model and the adopted assumptions.

III. Modeling cache network using Markov chain

A. System description

Let $G = (V, E)$ be the graph representing a general network of caches, where $V = \{v_1, ..., v_N\}$ depicts the nodes of the network and $E \subset V \times V$ is the set of links between the nodes. Each node in the network is equipped with a cache module used to store contents locally. Let $C = \{c_1, ..., c_K\}$ be the set of contents available in the catalog. We assume that all the accessible contents in the system have an identical size and are divided into small packets or chunks, which are in turn of the same size. The cache capacity is then expressed in terms of the number of contents or chunks that can store. All the available contents are stored permanently at one or more servers attached to some nodes within the network. In the rest of the paper and for the sake of readability, we will use the terms node/cache interchangeably as well as the terms catalog/library and content/item/object.

Clients, which are attached to the network nodes, send requests into the network seeking for contents. The pattern of these latter is characterized by the Independent Reference Model (IRM) [4]. Suiting the IRM model, users generate an independent and identically distributed sequence of requests from a fixed catalog of K objects $\{c_1, ..., c_K\}$. Specifically, the probability p_k to request an item k from the library is constant and follows a popularity law, where the contents are ranked decreasingly according to their popularity from 1 to K.

Since, in our work, we address video services, the contents feature a skewed popularity distribution. As already proven in many previous studies, the latter fits the Zipf distribution [9]. If we consider a catalog of K objects, then the probability to request the content of rank r is: $p_r = \frac{1}{r^\alpha}(\sum_{i=1}^{K} \frac{1}{i^\alpha})^{-1}$, where α represents the skew of the distribution and depends on the type of library's objects.

B. Single cache model

Let's consider a single node in a Content-Centric Network operating under the LRU replacement policy. When a user requests for a content of rank r in the catalog, it will generate a cache miss if the content is not present in the cache or a hit otherwise. In the latter case, the object will be sent back to the user. In the case of a cache miss, the client's interest is forwarded to the next nodes on the network towards the server, until the content is found either in an intermediate cache or at the server storing a permanent copy (i.e origin server). Once the object is located, it is sent along the reverse path and cached at each node that passes by it, which represents the default caching scheme used in CCN and known as Leave Copy Everywhere (LCE) [10].

Consider a cache sized to contain N contents. Whenever a local cache hit or a caching decision occurs for a content c_r of a rank r, it will then be placed at the top slot (slot 1) with a probability p_r. Consequently, for any given content $c'_{r'}$ different from c_r and occupying a slot i (i ranging from 1 to N) in the cache, three actions are possible when a request for c_r is received:

- $c'_{r'}$ will be moved down if the requested content c_r is either not present in the cache or occupies the slot j with $j > i$;
- $c'_{r'}$ will remain at the same slot if c_r occupies the slot j with $j < i$;
- $c'_{r'}$ will be evicted from the cache if it occupies the N^{th} and last slot.

This process can be simply modeled as a discrete-time Markov chain, where the state of the chain represents the exact slot that the content currently occupies in the cache going from state 1 to $N+1$. State 1 represents the case where the object is at the top of the cache and state N where it is at the bottom (see figure 1). State $N+1$ represents the case when the item is not in the cache due to a cache eviction, or because it was never stored in the cache (no interests have been received

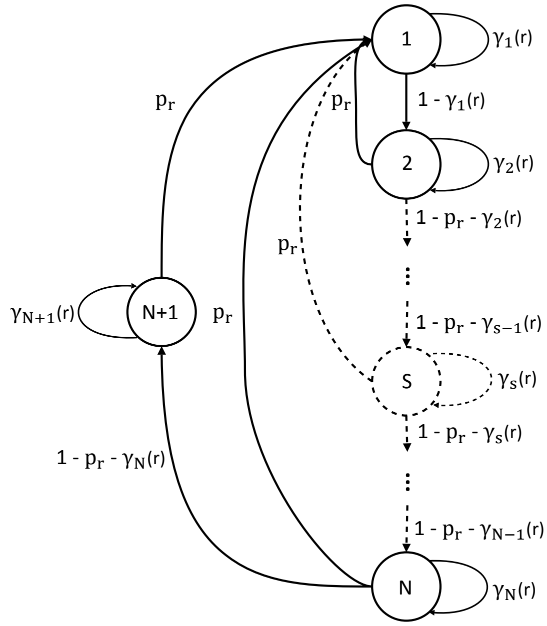


Figure 1: Markov chain-based model for contents with popularity rank r in a single cache

for this item). Let $\pi(r) = (\pi_1(r), \pi_2(r), \ldots, \pi_{N+1}(r))$ be the equilibrium probabilities of the different states of the Markov chain (figure 1). The probability that a content with a rank r is staying in the state s of the chain $(s \in \{1, 2, \ldots, N+1\})$ when a request is received is represented by $\gamma_s(r)$ and it is equal to:

$$\begin{cases} \gamma_1(r) = p_r \\ \gamma_s(r) = \sum_{i=1, i \neq r}^{K} p_i \left(\sum_{j=1}^{s-1} \pi_j(i)\right), 2 \leq s \leq N \\ \gamma_{N+1}(r) = 1 - p_r \end{cases} \tag{1}$$

Let's denote by $T_{i,j}$ the transition probability from state i to state j. Then, $T_{i,j}$ is equal to:

$$\begin{cases} T_{i,i} = \gamma_i(r), 1 \leq i \leq N+1 \\ T_{i,1} = p_r, 2 \leq i \leq N+1 \\ T_{1,2} = 1 - \gamma_1(r) \\ T_{i,i+1} = 1 - p_r - \gamma_i(r), 2 \leq i \leq N \\ T_{i,j} = 0, j \neq \{1, i, i+1\} \end{cases} \tag{2}$$

The equilibrium probabilities of a Markov chain exist only if the chain is ergodic. One way to prove this ergodicity is to find a number p such that any state can be reached from any other one in at most p steps, which is the case of our model, where p is equal to N. Hence, the equilibrium probabilities exist and can be computed using the following system of equations:

$$\begin{cases} \pi_i(r) = \sum_{j=1}^{N+1} \pi_j(r) T_{j,i} \quad , 1 \leq i \leq N+1 \\ \sum_{i=1}^{N+1} \pi_i(r) = 1 \end{cases} \tag{3}$$

If we develop the system of equations defined in (2) and (3), we obtain:

$$\begin{cases} \pi_1(r) = \gamma_1(r)\pi_1(r) + p_r(\pi_2(r) + \ldots + \pi_{N+1}(r)) \\ \pi_2(r) = \gamma_2(r)\pi_2(r) + (1 - \gamma_1(r))\pi_1(r) \\ \pi_3(r) = \gamma_3(r)\pi_3(r) + (1 - p_r - \gamma_2(r))\pi_2(r) \\ \ldots\ldots = \ldots\ldots \\ \ldots\ldots = \ldots\ldots \\ \pi_N(r) = \gamma_N(r)\pi_N(r) + (1 - p_r - \gamma_{N-1}(r))\pi_{N-1}(r) \\ \pi_{N+1} = \gamma_{N+1}(r)\pi_{N+1} + (1 - p_r - \gamma_N(r))\pi_N(r) \\ \pi_1(r) + \pi_2(r) + \ldots + \pi_{N+1}(r) = 1 \end{cases} \tag{4}$$

The equations derived in (4) can be easily solved using (1) to finally get:

$$\begin{cases} \pi_1(r) = p_r \\ \pi_2(r) = \dfrac{p_r(1 - p_r)}{1 - \gamma_2(r)} \\ \pi_i(r) = \dfrac{p_r(1 - p_r)\prod_{j=2}^{i-1}(1 - p_r - \gamma_j(r))}{\prod_{j=2}^{i}(1 - \gamma_j(r))}, 3 \leq i \leq N+1 \\ \pi_1(r) + \pi_2(r) + \ldots + \pi_{N+1}(r) = 1 \end{cases}$$

The closed-form expression $\pi_{N+1}(r)$ represents the cache miss probability for a content c_r in a single cache. Its computational complexity is $o(NK)$ where N represents the cache capacity, and K denotes the cardinality of the catalog. To get the cache hit rate, we simply do: $1 - \pi_{N+1}(r)$.

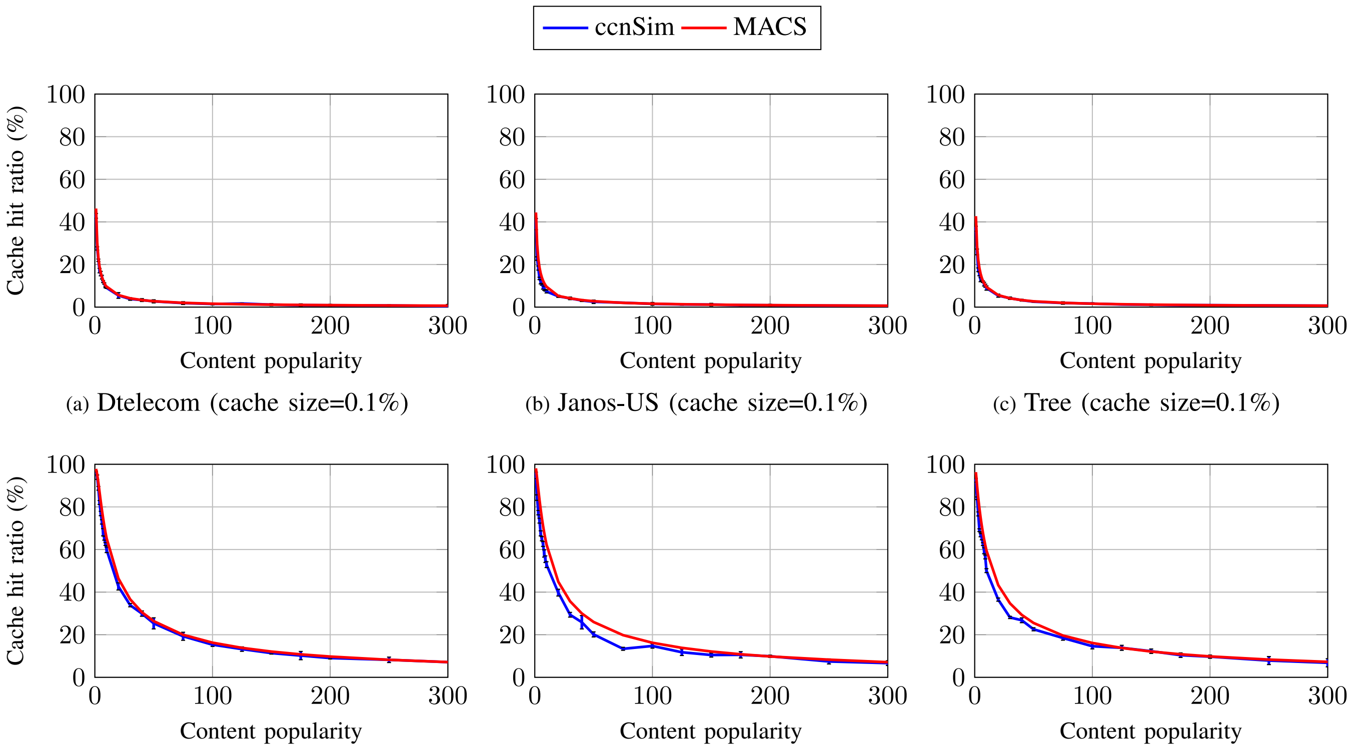
C. Multiple caches model

Now, let's consider a system of multiple CCN nodes where the contents are cached everywhere and forwarded according to the Shortest Path Routing (SPR) algorithm [11]. With SPR, when a client's interest cannot be satisfied by a node, it is forwarded along the shortest path to the closest permanent copy of the requested content. In this case, each node has to take into account, in addition to the local requests, the interests that come from other nodes due to a cache miss (we denote this stream of interests by "miss stream" or η, which will depend on the position of the node within the network). The incoming miss stream ratio for a content r at a node v can be defined as:

$$\eta_r(v) = p_r \times \sum_{u:\textit{NextHop}(u)=v} \pi_{N+1}(r, u)$$

The set $\{u : \textit{NextHop}(u) = v\}$ represents the nodes having v as the next hop in the shortest path to the source. $\pi_{N+1}(r, u)$ is the miss probability of content c_r at node u. So, the total proportion of requests that a node v will receive for a content will no longer be p_r, but another value that we denote as p'_r, which will contain in addition to the local requests, the interests due to a cache miss from previous nodes. For each node v, this value is equal to :

$$\begin{aligned} p'_r &= \frac{p_r + \eta_r(v)}{\sum_{k=1}^{K}(p_k + \eta_k(v))} \\ &= \frac{p_r + \eta_r(v)}{1 + \sum_{k=1}^{K}\eta_k(v)} \end{aligned}$$



(a) Dtelecom (cache size=0.1%)   (b) Janos-US (cache size=0.1%)   (c) Tree (cache size=0.1%)

(d) Dtelecom (cache size=1%)   (e) Janos-US (cache size=1%)   (f) Tree (cache size=1%)

Figure 2: Cache hit probability vs content popularity under different topologies and cache sizes with $\alpha = 0.8$

If we consider again the previous Markov chain (figure 1) and for every node v of the network, we get the equilibrium probabilities by replacing p_r by p'_r:

$$
\begin{cases}
\pi_1(r) = p'_r \\
\pi_2(r) = \dfrac{p'_r(1-p'_r)}{1-\gamma_2(r)} \\
\pi_i(r) = \dfrac{p'_r(1-p'_r)\prod_{j=2}^{i-1}(1-p'_r-\gamma_j(r))}{\prod_{j=2}^{i}(1-\gamma_j(r))}, 3 \le i \le N+1 \\
\qquad \pi_1(r) + \pi_2(r) + \ldots + \pi_{N+1}(r) = 1
\end{cases}
$$

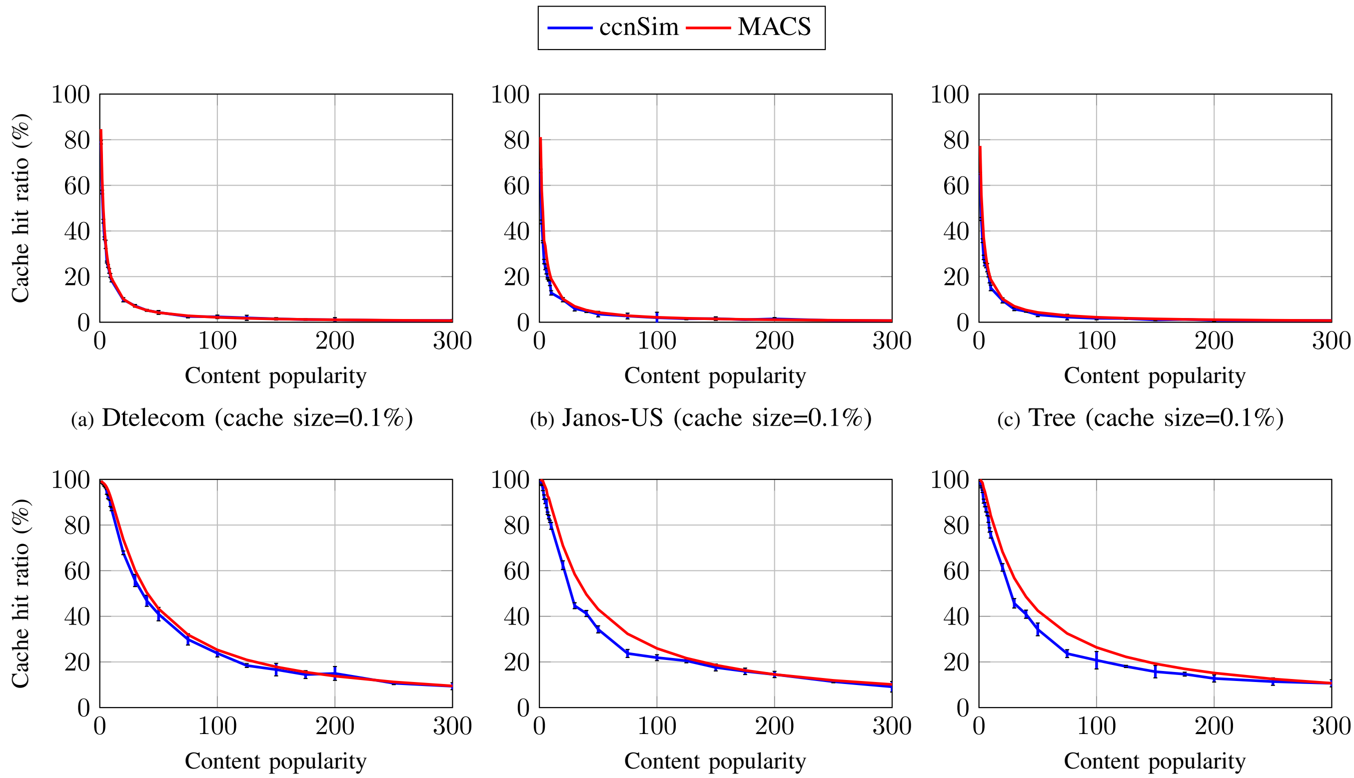
As we mentioned in the previous section, the cache hit probability is defined by: $1 - \pi_{N+1}(r)$.

IV. Model evaluation

A. Simulation environment

In order to evaluate the accuracy of our proposal, we compared the analytical model presented in the previous section with the results of simulations under ccnSim [12], which is a discrete-event and a chunk-level simulator for CCN networks. The accuracy of MACS, compared to the simulation results, can be affected by many parameters. Our focus in the conducted experiments was on the following key settings: network topology, cache size and Zipf law's skew distribution value. We measured during the tests the cache hit rate metric, which represents the ratio of requests that were served by the caches over the total number of sent requests in the network. We chose this metric since the aim of our proposal is to estimate the cache hit rate of a multi-cache system.

The simulations were conducted on three different network designs. The first one is a complete binary tree containing 31 nodes and 30 links. The others are more realistic network topologies named "Janos-US" (26 nodes and 42 links) and "Dtelecom" (68 nodes and 353 links). More details on the used network topologies can be found on [13] and [14]. In the simulation settings, we considered a catalog of contents containing $20,000$ 1-chunk sized objects whose popularity distribution follows the Zipf's law. Permanent copies of the available contents are hosted on one repository attached to one of the network's nodes. We set a uniform cache store capacity on the CCN nodes, which was defined as a proportion of the catalog size. Different simulations were conducted with a cache store size varying from 0.1% to 1.0% of the catalog size with an increment of 0.1%. Due to the lack of space, only two values for the content store capacity will be presented in the following: 0.1% and 1.0%. The total number of clients requests was set to $200,000$ with a Poisson arrival rate per node of one request per second. We tested also different values of the Zipf law's skew parameter α: 0.8, 1.0 and 1.2. As we mentioned previously, the Leave Copy Everywhere (LCE) caching scheme, and the Least Recently Used (LRU) cache replacement policy are used. All the settings used during the simulation are summarized in Table 1. Next, we will expose and compare the cache hit results obtained with our analytical model and with the ccnSim simulation tool.



(a) Dtelecom (cache size=0.1%) (b) Janos-US (cache size=0.1%) (c) Tree (cache size=0.1%)

(d) Dtelecom (cache size=1%) (e) Janos-US (cache size=1%) (f) Tree (cache size=1%)

Figure 3: Cache hit probability vs content popularity under different topologies and cache sizes with $\alpha = 1.0$

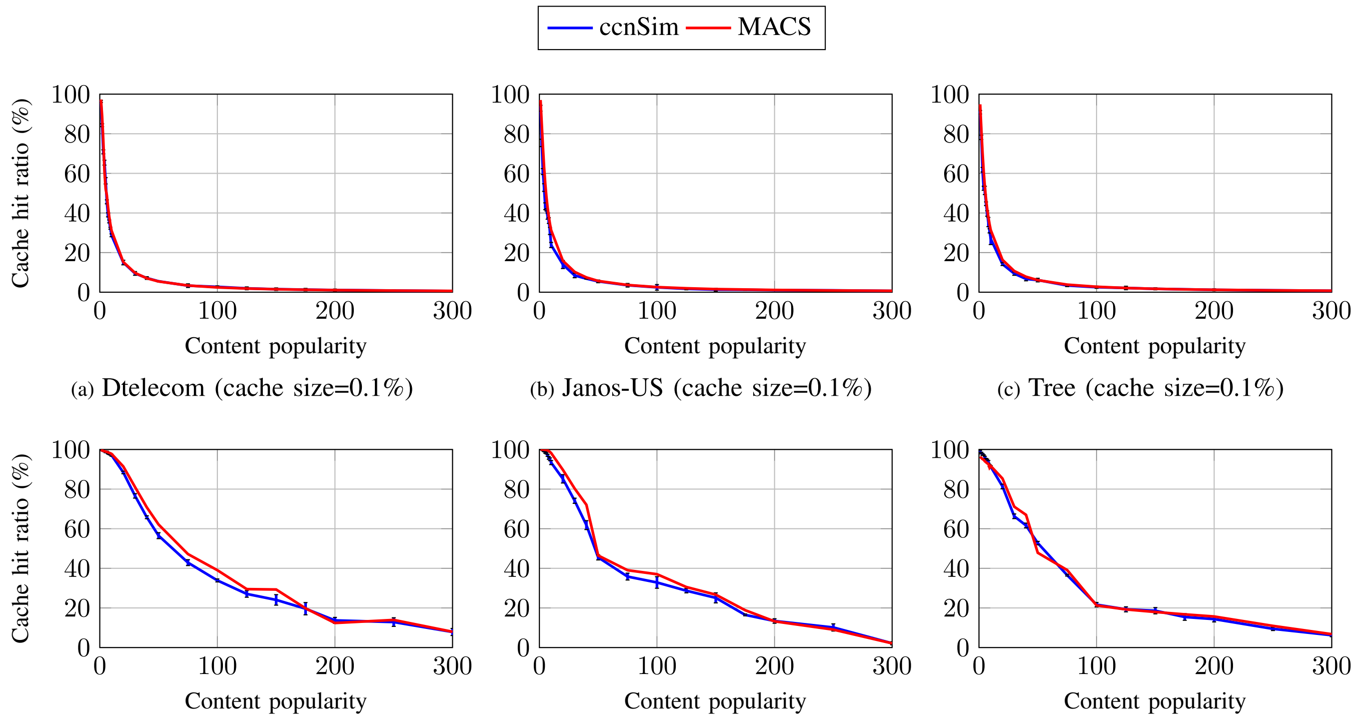
Table I: Simulation settings

PARAMETERS	VALUES
Catalog cardinality	20,000 contents
Total users requests	200,000 requests
Request arrival rate per node	1 request/sec
Content popularity	Zipf distribution (α = 0.8 / 1.0 / 1.2)
Cache capacity	0.1% / 1% of the catalog size
Cache replacement policy	LRU
Caching strategy	LCE

B. Simulation results and analysis

In this section, we display graphs (Figures 2, 3 and 4) showing the values of the network's average cache hit ratio as a function of the content popularity under different scenarios using ccnSim and MACS. The numerical simulation results shown on the graphs represent mean values taken over 10 simulations with error bars representing 99% confidence intervals. For the sake of clarity, we considered in the graphs only the objects whose popularity goes from 1 to 300.

It can be seen from the charts that the cache hit estimation of our model is accurate for the different considered scenarios. First thing that we can see is when α is equal to 0.8 (Figure 2), we get low caching efficiency (with both ccnSim and MACS) compared to the cases where α is set to higher values (Figure 3 and 4). This can be explained by the fact that as the Zipf law's skew distribution parameter decreases, the contents' popularity becomes progressively more uniform, which will result in nodes receiving and caching more distinct items and, thus, increasing the miss probability. When the cache size is set to a low value (0.1% of the catalog), the model performs better even with distinct values of α and for different types of content popularity (Figures 2(a)-(b)-(c), 3(a)-(b)-(c) and 4(a)-(b)-(c)). In the case when the cache capacity is set to 1% of the catalog size, the cache hit ratio in our model is estimated with lower accuracy compared to the other case (cache size of 0.1%). However, the cache hit approximation of the highly-popular objects remains precise in the different scenarios. If we compare the results obtained with the distinct tested networks, we can see that the Dtelecom topology is the most accurate in terms of cache hit estimation with different system configurations. According to the experiments conducted up to now, we can conclude that MACS performs better with a low cache capacity value (as a proportion to the catalog size) and that the α setting does not have major impact on the accuracy of our model. To determine and analyze the impact of the network topology and other parameters on our model performance, further experiments are still needed.



(a) Dtelecom (cache size=0.1%)
(b) Janos-US (cache size=0.1%)
(c) Tree (cache size=0.1%)
(d) Dtelecom (cache size=1%)
(e) Janos-US (cache size=1%)
(f) Tree (cache size=1%)

Figure 4: Cache hit probability vs content popularity under different topologies and cache sizes with $\alpha = 1.2$

V. Conclusion

The Content-Centric Network (CCN) [3] proposal is one of the most promising architectures for the future Internet. A significant feature provided by CCN is the in-network caching, where every node is equipped with a storage capacity in order to cache the flowing data. This key feature has a direct impact on the system performance, and it is important to analyze and evaluate the caching behavior in order to gain insights for optimized CCN caching schemes. We propose in this paper MACS, a Markov chain-based Approximation of CCN caching Systems to estimate the cache hit probability under the popular LRU replacement scheme of a multi-cache system (CCN in our case). The results show that our proposal, compared to the simulations conducted under the ccnSim tool, can estimate with high accuracy the cache hit rate of the system. Our model does not depend on a specific type of network topology, but it is for now applicable only to the case where the LCE caching scheme and the SPR routing engines are used. We intend in the future to extend our model to make it functional with other caching and routing protocols.

References

[1] Cisco, "Cisco visual networking index: forecast and methodology, 2015-2020," White paper, Jun. 2016.

[2] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1024–1049, Jul. 2014.

[3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, Dec. 2009, pp. 1–12.

[4] A. Dan and D. Towsley, "An approximate analysis of the LRU and FIFO buffer replacement schemes," *SIGMETRICS Performance Evaluation Review*, vol. 18, pp. 143–152, Apr. 1990.

[5] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1305–1314, Sep. 2002.

[6] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *IEEE INFOCOM Proceedings*, Mar. 2010, pp. 1–9.

[7] I. Psaras, R. Clegg, R. Landa, W. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *Proceedings of the 10th International IFIP TC 6 Conference on Networking*, May 2011, pp. 78–91.

[8] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," in *Proceedings of the 24th International Teletraffic Congress*, Sep. 2012, pp. 1–8.

[9] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, Oct. 2007, pp. 1–14.

[10] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, vol. 63, pp. 609–634, Jul. 2006.

[11] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "Ospfn: An ospf based routing protocol for named data networking," NDN Technical Report, Jul. 2012.

[12] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: An highly scalable ccn simulator," in *IEEE International Conference on Communications*, Jun. 2013, pp. 2309–2314.

[13] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0–survivable network design library," *Networks - Network Optimization*, vol. 55, pp. 276–286, May 2010.

[14] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1765–1775, Oct. 2011.