Multi-Attribute Caching: Towards Efficient Cache Management in Content-Centric Networks

Sheneela Naz
Center of Research in Networks
and Telecommunications (CoReNeT),
Muhammad Ali Jinnah University,
Islamabad, Pakistan
Email: shahneela@corenet.org.pk

Rao Naveed Bin Rais
Center of Research in Networks and
Telecommunications (CoReNeT),
Muhammad Ali Jinnah University,
Islamabad, Pakistan
Email: naveedbinrais@corenet.org.pk

Amir Qayyum
Center of Research in Networks and
Telecommunications (CoReNeT),
Muhammad Ali Jinnah University,
Islamabad, Pakistan
Email: aqayyum@ieee.org

***Abstract*—Content Centric Networking (CCN) has gained much attention of the research community in past few years. The default caching strategy of CCN is to store at each content-router along the downloading path. While this helps in increasing content availability and quality-of-experience (QoE) by reducing delay and reducing server load, the approach is not resource-efficient as it introduces high cache redundancy. Hence, there is a need to devise efficient caching mechanisms that allows maximum availability of content while consuming minimum possible resources. In this paper, we propose Multi-Attribute Caching Strategy (MAC) for CCN Networks that is based on using multiple parameters instead of a single attribute. These parameters include hop count or distance, node degree, content popularity and available storage space at the nodes. The scheme promises to overcome inefficient cache utilization by intelligently selecting caching locations along the content delivery paths. Simulation results indicate that MAC reduces the cache load at each node while increasing network cache hit rate.**

I. Introduction

During past decade, user requirements and emerging applications have pushed and challenged the traditional communication mechanisms to a great deal. Using a number of emerging applications such as content sharing, video-on-demand, and other P2P services, users are more interested in acquiring a particular content, rather than the location from where content is coming, and most of the time, are oblivious to the source identification as well. Content-Centric Networking (CCN) architecture promises to solve many problems related to the traditional network architecture as it provides a receiver-driven, secure and simplistic model while inherently supporting multiple interfaces per node. CCN provides transparent and ubiquitous in-network caching, name-based routing [5] etc. CCN utilizes the concept of in-network caching in order to enhance the availability of content in the network. As data packet traverses through the network, each Content-Router (CR) on the downloading path can potentially cache the content, as per the default cache policy of the architecture. While the idea helps in increasing content availability and quality-of-experience (QoE) to end user by reducing delay, and reduces load over servers, the approach is not resource-efficient as it introduces high cache redundancy, i.e., cached the content unnecessarily at multiple neighboring nodes. Hence, there is a need to devise efficient caching mechanisms that allows maximum availability of content while consuming minimum possible resources.

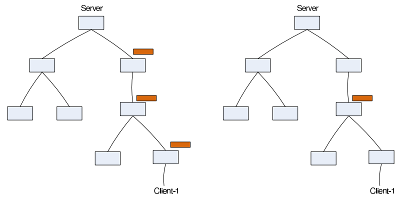
In order to address the problem of efficient content utilization, some approaches emphasize on node coordination to reduce this cache redundancy and improve cache diversity [3]. There have been many studies targeting implicit in-network caching [2], [7]. However, we argue that all these in-network caching strategies are not optimized because they are formulated only on a single parameter. The performance of implicit cache coordination schemes can be improved if we design caching strategies based on multiple parameters.

With this in mind, we propose a novel caching scheme, called Multi-Attribute Caching Strategy (MAC) for CCN that is based on using multiple parameters. These parameters include hop count or distance, node degree, content popularity and available storage space at the nodes. The scheme promises to overcome inefficient cache utilization by intelligently selecting caching locations along the content delivery paths in order to increase chances of getting maximum cache hits. We implement MAC in ndnSIM [1] simulator and evaluate its performance using a number of performance metrics including average cache hits and content re-usage. Moreover, we compare the performance of MAC with a number of contemporary cache coordination schemes and show that MAC significantly improves utilization of network resources while providing comparable performance to its counter parts. Our contributions in this paper can be summarized as follows:

- We propose a multi-attribute caching strategy (MAC) for CCN that determines caching location along the suitable content delivery paths while taking into account multi-dimensional analysis. MAC makes independent caching decisions at individual content-routers.
- We show that MAC determines relevance of each attribute by introducing associated weights.
- We evaluate MAC against some well-known contemporary caching strategies using a variety of performance metrics and parameters.

The remainder of this paper is organized as follows. Section II describes the literature review of some existing caching strategies. Section III illustrates the basic mechanism and design details of new caching scheme MAC while Section IV shows the performances of MAC. Finally, Section V concludes the paper and future work.

978-1-4673-9292-1/16/$31.00 ©2016 IEEE



(a) Basic CCN Cache Strategy (b) MAC Cache Strategy

Fig. 1. The comparison of CCN Default Strategy and proposed MAC Strategy.

II. RELATED WORK

Several recent studies have focused on ubiquitous in-network caching. Some approaches emphasize on node coordination to reduce this cache redundancy and improve cache diversity [9]. The cache coordination schemes can be classified into two categories, explicit cache coordination schemes [2] [8] and implicit cache coordination schemes [4], [11]. In explicit cache coordination schemes, every cache-router requires pre-information about caches state, users access frequency and exchange a lot of information. Along the same lines, the authors in [6] propose the Cooperative In-Network Caching (CINC) scheme, which is based on a hash function, and is shown to reduce cache redundancy and improve cache efficiency, by introducing higher computational complexity. In implicit cache coordination schemes, every content-router does not require pre-information and exchanges only small information with other content-routers. ProbCache scheme is also proposed as implicit cache coordination scheme in which the requested content is cached at every node with probability p along the path. Zhongxing et al. [7] proposed an age-based cooperative cache scheme in which content age is automatically calculated at every content-router between subscriber and provider. Moreover, Guoyin [11] proposes an optimal cache placement strategy(OCPCP)based on content popularity and OCPCP [11] improves the cache gain and reduces redundancy of the cache contents.

To summarize explicit cache schemes, resources are efficient enough, but computation overhead is added, while in implicit cache schemes, every cache-router is required to take decision efficiently and carefully due to partial information available at each cache-router.

III. MULTI-ATTRIBUTE CACHING STRATEGY (MAC) FOR CONTENT-CENTRIC NETWORKING

A. Overview

The default cache decision strategy of Content-Centric Networking (CCN) is to copy the content at each content-router along the downloading path while this helps in reducing congestion, end-to-end delay and improves delivery speed with high cache redundancy, it often leads to inefficient utilization of network resources, as shown in Figure 1(a). One way to optimize this content-cache is to consider multiple attributes regarding state of a content-router before taking cache decisions. For example, if we consider both, the degree of connectivity of a node and its distance from potential clients then we can take wiser decisions regarding content storage to achieve an optimal state as shown in Figure 1(b).

In order to solve shortcomings of current cache strategy in CCN, we propose a Multi-Attribute Caching Strategy for Content-Centric Networking (MAC). The main idea of MAC is to calculate the content lifetime and to decide whether to cache the content or not by taking into consideration node-degree, hop-count, content-popularity and available cache-space as the measurement metric.

B. MAC Design Details

1) Composite Cache Decision Score: In MAC, every CR along the downloading path calculates its composite content-cache score using eq 1. The aspiration of this scheme is to reduce cache redundancy, network resource consumption and improve content delivery. For this, we consider four parameters such as 1) hop count/distance, 2) content popularity, 3) degree of node, 4) available storage space. Every parameter is assigned an associated weight, which is used to optimize the content cache decision. The content cache decision equation can be mapped to Weighted Sum Model (WSM) [10]. Content cache score of CR 'N' is calculated as.

$$F_s(N) = w_1 \times C_p + w_2 \times D_n + w_3 \times H_c + w_4 \times A_{sp} \quad (1)$$

C_p, D_n, H_c, A_{sp} are real numbers within range [0,1] using Z-Score normalization method; their higher values represent a strong likelihood of that parameter. Weight w is defined as associated weight of different parameters, its value can vary within range [0,1]. Sum of all weights used for different parameters is always equal to one. Every parameter is assigned an associated weight to represent its relative importance within an optimized combination. We calculate our weights according to the rank sum method which is defined as

$$W_i = \frac{n - r_i + 1}{\sum_{k=1}^{n} n - r_k + 1} \quad (2)$$

Where W_i is the normalized weight for the i^{th} objective, n is the total number of objectives under consideration $(k = 1, 2, 3, ..n)$, r_j is the rank of the j^{th} objective.

The overall or composite performance score value is calculated from eq 3. The obtained score value is compared with $threshold$ value. If calculated score value is greater than the $threshold$ value, then CR stores the content. Otherwise, it forwards the content to the next CR that storing content locally. We generalize the formula to calculate content cache decision score as follows.

$$cD_{Score}(N) = \sum_{i=1}^{n} (w_i \times x_i) \quad (3)$$

where x_i is number of inputs parameters of node N and w_i is assigned weights of inputs parameters.

$$cD_{Score}(N) = \begin{cases} CacheContent & if cD_{Score}(N) > \mu \\ \text{NoCache} & \text{if } \mathrm{cD}_{Score}(N) \leq \mu \end{cases}$$

where μ is the $threshold$ value.

2) Content Lifetime: The pseudo-code of content-lifetime calculation at node 'n' is given in Table I & II.

When CR receive content data from a server then algorithm-1 (Table I) is processed. In this algorithm, first CR decides either to cache the content or not. This decision is based on content decision score value. If the content decision score value is greater than the threshold μ, then content is cached with the content-lifetime. Upon caching a content, a router implements ContentLifeTime () function to generate the content's lifetime. Where as h_{cf} value is hop count frequency, which is calculated as $h_{cf} = \frac{m}{n}$ where m is position of current

TABLE I. CONTENT-LIFETIME CALCULATION AT CR 'N' WHEN RECEIVED-CONTENT-DATA (RCD)

ALGORITHM-1

Let μ is the content cache decision threshold
LT$_{new}$ = *ContentLifeTime*$(Content)$;
IF $cD_{Score}(N) > \mu$ then
    CacheContent(LT$_{new}$, $Content$)
ENDIF
Forward(Content);

ContentLifeTime(Content) {
IF the router is the first node along the path then
    LT$_{old}$ = $GetInitLT()$
ELSE
    LT$_{old}$ = $GetLT(Content)$
ENDIF
    LT$_{new}$ = $LT_{old} + (1 - LT_{old}) * h_{cf}$
return LT$_{new}$;

TABLE II. CONTENT-LIFETIME CALCULATION AT CR WHEN RECEIVED CONTENT REQUEST

ALGORITHM-2

Let β is the constant value
LT$_{old}$ = *GetLT*$(Content)$;
LT$_{new}$ = $LT_{old} + (1 - LT_{old}) * \beta$
    SetLT$_{new}(LT_{new}, Content)$
ENDIF
Forward(Content);

CR and n is total number of CR's. Set the content's lifetime into 'content-lifetime' field and forward the information to the next router.

Although the concept of assigning content-lifetime is not new, in [7] paper lifetime is assigned to every content before it is being saved on any router. However it is taken as static value based on predefined weights. However in MAC, when CR receive the content request from the subscriber (or any CR) and if content is found in an intermediate CR then algorithm-2 is calculated. This algorithm ensures that a router keeps the data longer because it update the content-lifetime. Updating lifetime of already saved content this way has the the benefit to avoid unnecessarily occupation of resources of less popular content for unlimited time period.

MAC caching scheme avoids caching the content at every CR. In this way, the memory is efficiently utilized by minimizing content redundancy while maintaining acceptable performance.

IV. SIMULATION METHODOLOGY

In this section, we present in-depth evaluation to show the effectiveness of MAC scheme compared to some well known cache management schemes. We use the open-source ndnSIM simulator [1] which implements NDN protocol stack for NS-3 network simulator.

Simulation Parameters:

a) Network Cache Hit Rate: The network cache hit ratio is the ratio of content served from the intermediate CR along the path before the content requests arrive at the repository server.

$$AverageHit_{Rate} = \frac{\sum_{i=1}^{N} Hits(i)}{\sum_{i=1}^{N} Req(i)} \tag{4}$$

TABLE III. SIMULATION PARAMETERS

Attribute	Value
Number of different content	200
Content size	1024B
Cache size	50
Link delay	10ms
Bandwidth	10Mbps
α of Zip's distribution	0.3 /0.9
Request frequency	3/s
Simulation time	200s
Simulation runs	10
Threshold μ value	0.6

$Hits(i)$ represent the number of hits on the $CR_{(i)}$ and $Req_{(i)}$ represent the number of received content requests on the $CR_{(i)}$.

b) Average Buffer Usage: We measure the impact of MAC by average buffer usage that is measured by used cache space along the path.

$$AverageBuff_{(usage)} = \frac{\sum_{i=1}^{N} CacheSize(i)_{Used}}{\sum_{i=1}^{N} CacheSize(i)} \tag{5}$$

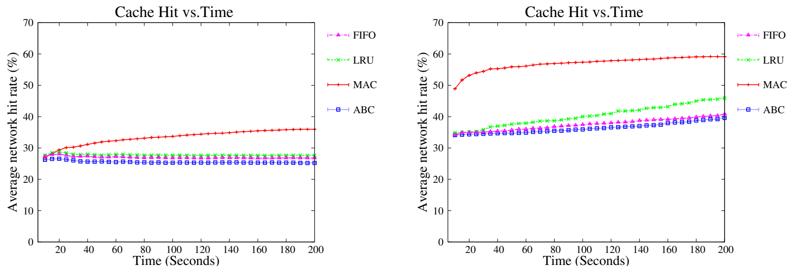
$CacheSize(i)$ used represents the used cache space on $CR_{(i)}$ and $CacheSize_{(i)}$ represents the total count of cache size along the path.

Simulation Scenario: To make a comparison, we use ABC, LRU and FIFO competes with MAC in the same scenario. First in First out (FIFO) and Least Recently Used (LRU) are default caching scheme. Implicitly cache coordination policy is used in age-based cooperative cache scheme (ABC). We run simulations on network topology which consists of 13 consumers, 14 content-routers (CR) and a single server to satisfy the requested interests. In our experiments, all the CR configured with the same cache capacity and 200 different content objects in the network. A best route policy is used in the network and Zipf-like distribution is also implemented for request distribution in evaluating caching performance. Table III lists the performance parameters used for the evaluation. Following are some selected parameters which we use in this scenario.

- *Hop Count/Distance:* Hop Count based distance estimations is an important element for content cache placement strategies. The probability of content cache is inversely proportional to the distance from the requester and provider.
- *Content Popularity:* Content popularity is measured as the number of requests count for each content name.
- *Degree of Node:* The probability of content cache is directly proportional to the node degree.
- *Available Cache Space:* Available cache space defines how to allocate the node cache space across different cache contents.

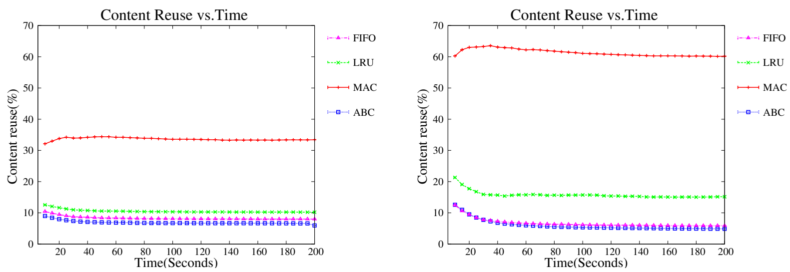
Evaluation & Results:

c) Cache Hit Rate: The cache hit rate is the response of the request from the intermediate CR before the request arrives at the source server. The cache hit rate is formulated as the ratio of hit counts to total request counts. Request distribution is assumed to be Zipf(α) with α = 0.3 or 0.9, which is used in our experiments.



(a) (Network Cache Hit Rate with α = 0.3) (b) (Network Cache Hit Rate with α = 0.9)

Fig. 2. Cache Hit Performance Comparison between Proposed Scheme and Others



(a) (Content Reusability with α = 0.3) (b) (Content Reusability with α = 0.9)

Fig. 3. Comparison of Content Reusability with Proposed Scheme and Others

A MAC caching scheme achieves a higher cache hit rate(at = 0.3) i.e. 50% as compared to other schemes (i.e. 30% of LRU, 27% of FIFO, 25% of ABC) at simulation time 150s. At the start of the simulation all caching techniques have the same cache hit rate because all techniques needs to learn through a large number of requests. After 20s, MAC reaches to the steady state. MAC caching scheme improves the network cache hit rate as compared with the existing chaining schemes like FIFO, LRU and ABC as shown in Figure 2. The reason of this MAC improvement is to store the contents at their prime locations and increase the content lifetime when the request of the same content is generated again. So, MAC is able to cache more several contents in the network.

d) Content Re-usability: The Content re-usability can be calculated by the ratio of hit count to the total occupied buffer space. Figure 3 shows the MAC scheme improves the content re-usability as compared to the other caching schemes. At 40 seconds all caching schemes reach to the steady state. At this stage, all caching schemes have a lower content re-usability rate as compared to MAC that is approximately 13% of LRU scheme, 10% of FIFO and 8% of ABC scheme. FIFO, LRU and ABC show the lower content reuse rate because α = 0.3, smaller α shows the same content popularity and these schemes cannot respond quickly to the dynamic content request.

V. Conclusion

In this paper, we proposed a novel efficient caching strategy, named Multi-Attribute Cache (MAC), which performs multi-dimensional analysis while making content storage decisions. It take concepts considering different aspects of a CCN network such as, global topology, node connectivity, and content-based information. Moreover, in MAC relative importance of each concept is determined through weights. These, attributes with associated weights are then employed to determine the probability to store a content, at each router on returning path. If the probability is greater than a certain threshold value content is stored, otherwise, it is simply forwarded to next downstream content-router (CR). We show that MAC is able to demonstrate better performance than many contemporary cache management schemes by offering better average network hit rate and improved content reuse at CR.

References

[1] Ndnsim project [eb/ol]. In *http://ndnsi m.net/*, 2012.

[2] César Bernardini, Thomas Silverston, and Olivier Festor. Mpc: Popularity-based caching strategy for content centric networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3619–3623. IEEE, 2013.

[3] Sem Borst, Varun Gupta, and Anwar Walid. Distributed caching algorithms for content distribution networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[4] Giovanna Carofiglio, Massimo Gallo, Luca Muscariello, and Diego Perino. Modeling data transfer in content-centric networking. In *Proceedings of the 23rd international teletraffic congress*, pages 111–118. International Teletraffic Congress, 2011.

[5] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 181–192. ACM, 2007.

[6] Zhe Li and Gwendal Simon. Time-shifted tv in content centric networks: The case for cooperative in-network caching. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–6. IEEE, 2011.

[7] Zhongxing Ming, Mingwei Xu, and Dan Wang. Age-based cooperative caching in information-centric networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 268–273. IEEE, 2012.

[8] Zhongxing Ming, Mingwei Xu, and Dan Wang. Age-based cooperative caching in information-centric networking. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pages 1–8. IEEE, 2014.

[9] Dario Rossi and Giuseppe Rossini. Caching performance of content centric networks under multi-path routing (and more). *Relatório técnico, Telecom ParisTech*, 2011.

[10] EK Zavadskas, Z Turskis, J Antucheviciene, and A Zakarevicius. Optimization of weighted aggregated sum product assessment. *Elektronika ir elektrotechnika*, 122(6):3–6, 2012.

[11] Guoyin Zhang, Bin Tang, Xianghui Wang, and Yanxia Wu. An optimal cache placement strategy based on content popularity in content centric network.