Sensing Content Correlation-aware In-network Caching Scheme at the Edge for Internet of Things

Ngoc-Thanh Dinh
Soongsil University, Seoul, Korea
thanhdcn@dcn.ssu.ac.kr

Nhu-Ngoc Dao
Chung-Ang University, Seoul, Korea
dnngoc@uclab.re.kr

Younghan Kim
Soongsil University, Seoul, Korea
younghak@ssu.ac.kr

ABSTRACT

Existing caching schemes process content objects (COs) individually based on the exact matching without considering the semantic correlation among content objects. We argue that this approach is inefficient in the Internet of Things (IoT) due to the highly redundant nature of IoT device deployments and the data accuracy tolerance of IoT applications. Therefore, the cache of a different CO having a high semantic correlation with another requested one by an application can be reused if the CO meets the data accuracy requirement of the application. In this case, caching both COs is inefficient. This paper extends the concept of cache hit and proposes a caching scheme considering the semantic content correlation of nodes to evaluate, construct, and enable the re-usability of available cached items to serve more diverse requests based on information correlation. For demonstrating its benefits, we implement the proposed scheme on the top of LFU (Least Frequently Used) for IoT data caching at the edge. Obtained experimental results show that the proposed scheme achieves a significant cache hit ratio improvement in comparison with LFU.

1 INTRODUCTION

In existing ICN caching schemes [4], a cache hit occurs only when the requested content name and the cached content are matched exactly. This approach is inefficient in IoT, where a sensor normally produces a type of content over time, due to the following reasons. Sensing data of different IoT nodes may have a high correlation due to the redundant nature of IoT device deployment. In addition, IoT applications and users are interested in receiving sensing content with a given data accuracy requirement [3]. It means that a given level of error tolerance is acceptable. Therefore, a cached content object (CO) of a sensor can be reused to serve similar application interest requests for other sensors of the same type (i.e., temperature, humidity, image, video, accelerometer,...) as long as the CO meets data accuracy requirements of applications.

For that reason, this paper extends the concept of a cache hit considering the semantic correlation among COs as follows. For a given application content request, a cache hit occurs at a node when the node contains a cached CO that satisfies the requirement of the request, not necessarily the exact object that the application requests for. In particular, if a cached item, regardless of any content publisher of the same type has a high correlation with the requested one, that satisfies the application requirement, a cache hit also occurs.

This paper presents a novel content correlation-based caching (CC) scheme which is proposed to evaluate, construct, and enable the re-usability of available cached items to serve more diverse requests based on information correlation. The objective of CC is to improve the cache hit ratio within a limited cache capacity by 1) caching only necessary items to optimize the storage, thus reserving caching capacity for diverse items 2) exploiting available cached items to serve eligible requests. In the former, CC is used to assess further the cache decision for optimizing. If requests for a CO c can be served by reusing available cached items of others, then c is not necessarily cached. In the latter, CC discovers content correlated communities (CCC) of nodes with given correlation levels, constructs CCC cache tables and implements a cache matching mechanism based on not only the exact name but also the information correlation.

For demonstrating benefits of CC, we implement CC on the top of LFU (Least Frequently Used) [**?**] for IoT data caching at the edge. Obtained experimental results show that the proposed scheme achieves a significant cache hit ratio improvement in comparison with LFU. The concept of correlation-based caching can be extended for other scenarios like advertising contents, searching, or multimedia applications, which are our future works.

2 THE PROPOSED CC CACHING SCHEME

2.1 Application model

Assume we have a set N of n application requests for sensing content. Each request a_i contains a set of sensing data types of interest st_i provided by a wireless sensor network (WSN) and a specification for the Data Accuracy (DA) level DA_i. For example, some applications request content with a DA level of 95% while other applications may accept a DA level of 90%. n summary, the set of application requests is denoted as $N = (a_1, st_1, DA_1), (a_2, st_2, DA_2), ..., (a_n, st_n, DA_n)$.

2.2 Node Naming and Content Object Naming

We use the naming structure presented in our previous work [2] for sensor nodes and their content objects. The name includes two parts. The first part expresses the real-world category name of sensors, called the category prefix (CP). The latter part is the ID which makes the name persistent and unique. The name of a sensor is also used to name its content objects (COs) because a sensor normally produces a type of content over time. A CO has an additional sequence number which indicates the version of the CO. For example, a temperature sensor has the name of "Temp::8734"

ICN '19, September 24–26, 2019, Macao, China

ACM ISBN 978-1-4503-6970-1/19/09…$15.00
https://doi.org/10.1145/3357150.3357413

while one of its COs has the name with a sequence number of "Temp::8734/p=0045".

2.3 CCC discovery

The CCC of a node k (i.e., CCC_k^{da}) is defined as a set of nodes h that have sensing content correlation greater than or equal to da compared to k. Therefore, the sensing data of k can be exploited for serving application content requests for nodes in CCC_k^r with the DA requirement $\leq da$. The CCC discovery was presented in detail in our previous study [3].

2.4 CCC Caching Table

Based on the set N of application requests, the edge performs the CCC discovery for sensor types and their corresponding DA requirements. The edge then constructs a CCC caching table as shown in Table 2.

Table 1: Illustration of an CCC caching table

Prefix	Correlation	CCC Lists	Cache pointer
Temp	L1 (ID1, ID2, ID5)	97%	ID1*
Temp	L2 (ID1, ID2, ID5 ID11, D17)	94%	ID1*
Temp	L3(ID3, ID4, ID9)	94%	ID4*
Temp	L1 (ID1,âĂę, ID13, ID14)	91%	ID1*
Temp	L2 (ID3,âĂę,ID10)	91%	NULL
Temp	L3(ID6, ID7, ID15)	91%	ID6*

The CCC caching table consists of lists of CCCs with their corresponding content types and DA requirements. Each list has a cache pointer which is (1) a pointer pointing to a cached CO by a member of the list, or (2) NULL if there is no node in the list having cached COs.

2.5 CC Caching Decision

CC can be implemented with any existing cache replacement scheme to improve the scheme' performance. In this paper, we implement CC on the top of LFU [**?**]. When a new CO c of a sensor arrives at the edge and the LFU makes a decision to cache, CC is used to assess further to optimize the cache decision of LFU as follows.

CC first checks lists in the CCC caching table that contain the name of c. If there is no list containing the name of c or there are application requests for c with a high DA requirement that the existing CCCs containing c cannot satisfy, the system executes caching for c. If there is a set s of lists containing the name of c, CC checks if any list in the set has no cache pointer. In other words, there is no available cached item for nodes in the lists. If it occurs, CC executes caching for c. CC then updates the caching table by adding a cache pointer to the cache of c for the related lists. If all lists in s have a cached CO already, the system ignores and does not cache c

2.6 Name Matching and Cache Hit

When the edge receives an application request with a CO interest name c and a DA requirement, the edge checks its content store (CS). If c is cached in the CS, the edge returns to the request with the cached CO. If there is no cache for c, the edge further explores the CCC caching table. If the table has any list containing c, that has a cached CO with the correlation which is greater than or equal to the DA requirement. The edge responds with the cached CO indicated by the cache pointer. If there is no list satisfies or no cache, the request is forwarded based on FIB following CCN/NDN [1].

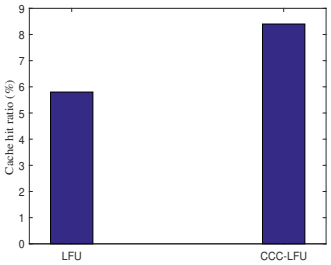

Figure 1: Cache hit ratio of LFU and CCC-LFU

In this way, we extend the concept of a cache hit in ICN. For a given application content request, a cache hit occurs at a node when the node contains a cached CO that satisfies the requirement of the request, not necessarily the exact object that the application requests for. If a cached item, regardless of any content publisher of the same type has a high correlation with the requested one, that satisfies the application requirement, a cache hit also occurs.

3 EVALUATION AND DISCUSSION

We implement a prototype of CC in Contiki CCN-Lite [1] on the top of LFU. We then perform simulations using COOJA simulator [1] with an edge node and 1000 sensors of temperature, humidity, and accelerometer types. Sensors are deployed randomly with sensor data collected from the real-world IntelLab sensor deployment [5] for a natural correlation among sensors. The edge cache can store a maximum of 20 COs. We generate randomly 1000 application content requests following a Zipf-like distribution. Obtained results presented in figure 1 show that CCC-LFU achieves a significant improvement in term of the cache hit ratio compared to LFU.

Due to resource constraints, we perform experiments with light-weight sensing data and limited cache capacity. Experiments can be conducted similarly for other types of sensing data like image or video. We plan to conduct extensive experiments to show more benefits in terms of data retrieval delay, diversity metrics, and energy efficiency for sensors. For future works, we extend the concept of content correlation-based caching for other scenarios like advertising contents, searching, and multimedia applications.

REFERENCES

[1] Bengt et al. Ahlgren. 2016. Demo: Experimental Feasibility Study of CCN-lite on Contiki Motes for IoT Data Streams. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. ACM, New York, NY, USA, 221–222.
[2] N. Dinh and Y. Kim. 2013. Potential of information-centric wireless sensor and actor networking. In *2013 International Conference on Computing, Management and Telecommunications (ComManTel)*. 163–168.
[3] N. Dinh and Y. Kim. 2019. An Energy Efficient Integration Model for Sensor Cloud Systems. *IEEE Access* 7 (2019), 3018–3030.
[4] A. Ioannou and S. Weber. 2016. A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking. *IEEE Communications Surveys Tutorials* 18, 4 (2016), 2847–2886.
[5] S. Madden. 2018. *Intel Berkeley Research Lab Data.* [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html.