Received: 16 July 2017 | Revised: 14 November 2017 | Accepted: 16 December 2017
DOI: 10.1002/dac.3521

RESEARCH ARTICLE

WILEY

An efficient in-network caching decision algorithm for Internet of things

Dong Doan Van[1,2,3] | Qingsong Ai[1,2]

[1] School of Information Engineering, Wuhan University of Technology, Wuhan, 430070, Hubei, China

[2] Key Laboratory of Fiber Optic Sensing Technology and Information Processing (Wuhan University of Technology), Ministry of Education, Wuhan, China

[3] Faculty Electrical and Electronic Engineering, Ho Chi Minh City University of Transport Ho Chi Minh, 700000, Vietnam

Correspondence
Qingsong Ai, School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China.
Email: qingsongai@whut.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 51675389 and 51475342

Summary
Recently, content-centric networking (CCN) has become one of the important technologies for enabling the future networks. Along with its recognized potentialities as a content retrieval and dissemination solution, CCN has been also recently considered as a promising architecture for the Internet of things (IoT), because of 2 main features such as named-based routing and in-network caching. However, IoT is characterized by challenging features: small storage capacity of resource-constrained devices due to cost and limitation of energy and especially transient data that impose stringent requirements on the information freshness. As a consequence, the intrinsic caching mechanisms existing in CCN approach do not well suit IoT domains; hence, providing a specific caching policy at intermediate nodes is a very challenging task. This paper proposes an effective multiattribute in-network caching decision algorithm that performs a caching strategy in CCN-IoT network by considering a set of crucial attributes including the content store size, hop count, particularly key temporal properties like data freshness, and the node energy level. Simulation results proved that our proposed approach outperforms 2 cache management schemes (probabilistic least recently used and AlwaysCache–first in first out in terms of improving total hit rate, reducing data retrieval delay, and enhancing content reusability in IoT environment).

KEYWORDS
AlwaysCache-FIFO, content-centric networking, Internet of things, LRU

1 | INTRODUCTION

The Internet of things (IoT) is a paradigm where all billions of devices communicate at the same time with each other and the humans over the Internet to implement some predefined objectives. These devices can be equipped with identifying, sensing, actuation, and processing capabilities, ranging from resource-constrained to the powerful devices.[1-3] Contrary to traditional networks, IoT is a challenging environment, mainly due to resource-constrained devices, heterogeneous access technologies, and special traffic patterns (eg, unique, heavy, and transient data). Some evolutionary approaches and standardization activities are pursued to provide IP-based networking functionalities for IoT (eg, 6LoWPAN, RoLL, and CoRE).[4] Although, these efforts have been illustrated through their valuable achievements, they still face great challenges. It is still very difficult to simultaneously service the large number of processing under the stringent requirements of IoT, complex mobility, and multicast in the case of large-scale deployment.

As the number of connected devices is growing rapidly and becoming ubiquitous, there is the need to provide a connectivity model for IoT environment to support more efficient and large-scale deployment for its applications. Hence,

https://doi.org/10.1002/dac.3521

Key Findings

- To propose effective multiattribute in-network caching decision algorithm that is based on a set of decisive attributes for Internet of things applications to select the powerful nodes/devices along delivery paths to store the data items.
- To determine the importance of each attribute by weight vector calculation, which greatly supports more accurate caching decisions.
- To evaluate the impact of different key Internet of things attributes on the proposed approach in comparison with typical caching strategies.

the research community is considering a novel communication model called content-centric networking (CCN) as a fully promising solution in the future. This network uses application data names directly to achieve the information retrieval by using named data instead of host to host in the Internet's current conversation model, in which named content is uniquely identified and independently retrieved from its location.[5] Interest packets (*IntPk*) and data packets (*DataPk*) are 2 types of packets that identify the content. Once a CCN node receives an *IntPk* message, the *DataPk* will be sent back to the consumers to ascertain if an appropriate content is found; otherwise, the *IntPk* is forwarded to upstream towards other potential content sources.

In this CCN scheme, some key features have been explored to take several advantages in IoT domain like uniquely identified, retrieved independently from its location, and content-based security. In parallel, the fact that IoT is increasingly focusing on data and information rather than point-to-point communications may lead to the adoption of CCN architectures and principles. As a result, CCN helps to improve energy efficiency, reduce bottleneck link, retrieval delay time, and limit massive access to resource-constrained devices, which is a clear indicator to prove the prospects of the CCN-IoT integration onwards.[6,7] However, this integration is still at its infancy, and therefore, there exist numerous challenges that need to be addressed concerning naming, security, discovery and delivery, and an especially caching aspect that will be intensely investigated in this paper.

In relation to in-network caching, the CCN-IoT integration network helps to speed up data-retrieval time, reduces the number of path-lengths, and improves energy efficiency. On the other hand, IoT is a challenging environment as mentioned previously, which is characterized by generated data content (eg, small size, a short lifetime “transient”) to device capabilities (eg, low battery level and small storage size). So far, caching decisions are likely to be dependent on several single metrics such as data freshness, request rate, cache size, and even communication cost, which lacks comprehensive methodologies for its assessment and improvement.

For these reasons, an efficient caching decision and replacement policy considering a set of attributes of IoT are the main objective of this paper. To the best of our knowledge, this is a pioneering research work considering all the key peculiarities of IoT in a caching policy. The main contributions of this paper are summarized as follows. We propose effective multiattribute in-network caching decision (MACD) algorithm that is based on a set of decisive attributes for IoT applications to select the powerful nodes/devices along delivery paths to store the data items. We determine the importance of each attribute by weight vector calculation, which greatly supports more accurate caching decisions, especially in case some attributes vary over time. Moreover, we evaluate the impact of different key IoT attributes on the proposed approach in comparison with some typical caching strategies.

The rest of the sections in this paper are organized as follows. Section 2 highlights some of the related studies that pertain to this paper. Section 3 shows our MACD algorithm, and we evaluate and discuss the results in Section 4. And finally, the paper is concluded in Section 5.

2 | BACKGROUND AND RELATED WORK

2.1 | CCN: Overview

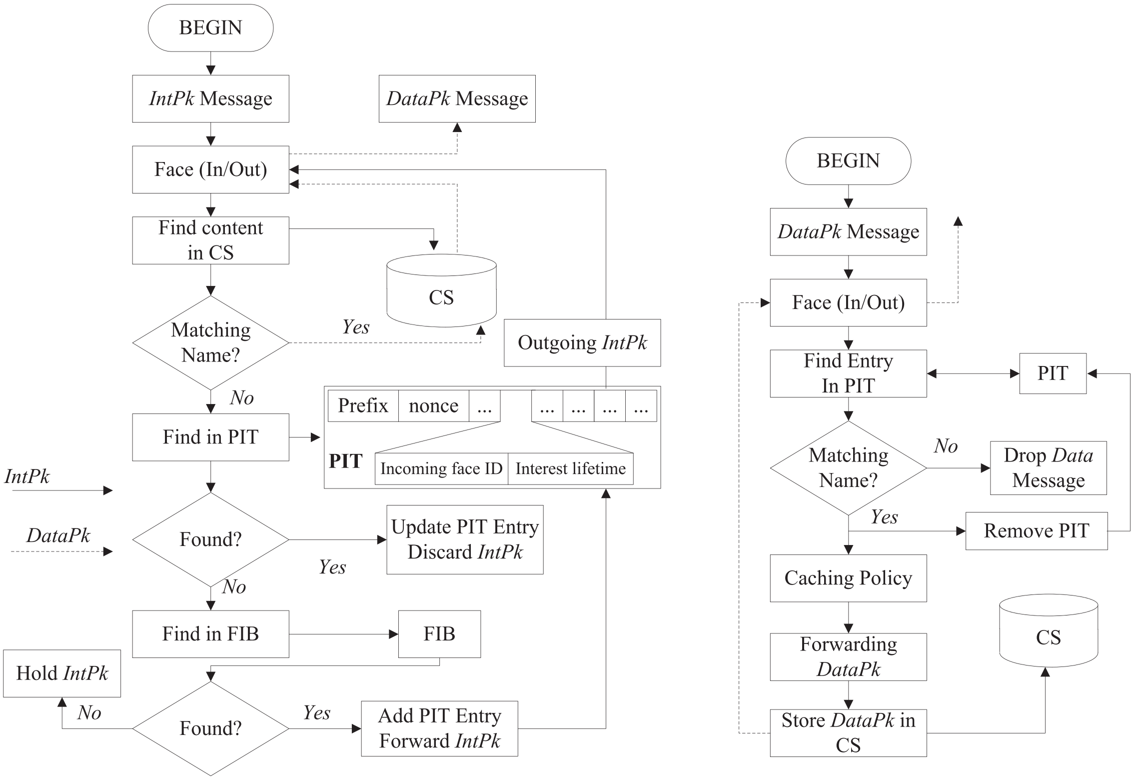
The CCN presented a simple and effective communication model. In brief, CCN was developed to perform efficient reuse of the content based on smart caching of popular content near the requesting users. In CCN, 2 types of packets are envisioned to identify a content, which is typical hierarchical by the sign “/” and human readable (eg, monitor/

temperature/ area_01/sensor-001). They are called interest packets (*IntPk*) and data packets (*DataPk*). In typical CCN, each node consists of 3 main data structures, the forwarding information base, pending interest table (PIT), and content store (CS). Once a CCN node receives an *IntPk*, it looks up to the CS. If an appropriate content is found, the *DataPk* will be sent for a request; otherwise, the *IntPk* will be checked in PIT (see the structure of PIT shown in Figure 1a). Pending interest table keeps track of unsatisfied *IntPks*. After PIT creates a new entry for an unsatisfied *IntPk*, the *IntPk* is forwarded to upstream towards a potential content source based on forwarding information base's information. A returned *DataPk* will be sent to downstream and stored on CS. Content store provides a cache to store the content available at the node and content received from other nodes based on the caching policy of the node for a certain time. When the “caching” deadline expires, the content is removed to cope with the limited size of content storage. When CS is about to get full or receive a new content, it stores the new content according to the underlying replacement policy to leave space for the new content. Least recently used (LRU), least frequently used, and first in first out (FIFO) are few notable examples of typical replacement policies for CCN. In summary, CCN flow diagram[8] is shown in Figure 1.

2.2 | Related work

In CCN, an important issue in on-path caching is how each node makes a caching decision to increase the cache hit rate of content delivery, improve network resource use, and speed up content distribution.[9] Some caching decision and replacement strategies have been applied to solve this problem.[10-13] These traditional strategies are mainly applied for static data items (eg, multimedia files), popular contents, and a sheer number of users as a feature of content delivery and request forwarding system. However, they have been shown to be inefficient and quite expensive due to the peculiarities of IoT environment. Only a few preliminary studies focus on caching in IoT systems but do not propose a different and comprehensive caching policy.[14-16]

The feasibility of deploying CCN in-network caching for IoT has started gaining momentum by the research community. Information-Centric Networking Research Group[17] has been proposed as a network paradigm, which enables native content awareness by an underlying network including content searching/resolution and caching. In Baccelli et al,[18] the first experiment with named data networking on the real IoT deployment is provided. In this works, caching



(A) Interest message (*IntPk*) flow

(B) Data message (*DataPk*) flow

FIGURE 1 Content-centric networking flow diagram. CS, content store; FIB, forwarding information base; PIT, pending interest table illustrated to be highly beneficial even with constrained nodes (eg, small storage capacity) by comparing the implementation of data retrieval from various consumers in the presence of standard named data networking in-network caching and when caching is disabled. In addition, Quevedo et al[19] also built a scheme for constrained-resource devices based on the key idea and fundamental CCN model for more efficient edge networks, in which weak network devices with constrained resources map the overcapacity task in terms of storing, publishing, and retrieving supper routers/strong devices.

Since IoT contents are transient, the information freshness is a very important parameter that highly affects the performance of caching strategies existing in CCN approach. In Song et al,[14] a new consumer-driven freshness approach for CCN was analyzed. In this scheme, content can be considered as valid in the CS by establishing a certain period. Old contents are dynamically discarded from a CS that honors the freshness declared by the producer. As a result, a freshness value included in both interest and data packets was explored to conduct the accuracy of caching and retrieval operations.

On the other hand, the work in Vural et al[15] proposed a distributed probabilistic caching algorithm for more efficient usage of the available caching space, alleviating the load of nodes and bandwidth consumption, where nodes automatically update their caching probability by using information about the network topology, the freshness value, and the rate of incoming requests. However, this scheme was mainly designed for the case of large storage capacity; it was not well supported for constrained devices and low power networks. To solve this limited storage, the different caching and replacement policies were evaluated in CCN-IoT wireless network in Hail et al.[16] In particular, probabilistic caching coupled with LRU gives the highest performance in terms of retrieval delay and interest retransmission. Vice versa, "always" caching strategy leads to the high level of content redundancy and poor use of available cache resources. So far, CCN in-network caching in IoT is still in the early stages of implementation, and a very limited number of studies have focused on this domain both theory research and experimental implementation. Meanwhile, the existing caching decision and replacement policies normally focus on the well-known strategies (AlwaysCache-FIFO, AlwaysCache-LRU, etc) with some key features of the content separately. The precious researches were not specially mentioned to all key attributes of the peculiarities of IoT for the accurate caching decision that is fully considered in the paper.

3 | MACD ALGORITHM

In the following, the MACD approach is presented, including several steps. In the first step, the algorithm will collect the information related to the crucial attributes that are presented in detail. Then, a discussion on the proposed scheme will give a caching decision for each of the content traversed intermediate nodes (detailed in section 3.2).

3.1 | The crucial attributes in IoT

Internet of things is a challenging environment as described above, thus the selection of elements is extremely important, which in turn affects the caching decision. In this paper, the focus is on several key attributes such as data freshness, the cache size, hop count, and energy level of the nodes. These parameters are calculated according to the expression detailed hereafter.

The key properties of IoT data items are small in size and transient in nature, which are different in comparison with multimedia files in almost all of the current networks. When caching a transient data, the item is considered to be valid if it is stored for a certain period known as data item lifetime (T), which is defined and generated by its source or producers. The source node generates data item in a pull-based manner for each content, eg, only upon receiving a request packet based on that content identifier. In the content identifier field, each data item that is generated for a special content also contains a lifetime field indicating the duration for which the value carried in the item is valid after its generation time. Therefore, when a data item is traversed to an intermediate node,[20] it will be evaluated to determine whether or not cache to the CS by freshness value that is given by 1.

$$Freshness = \frac{T - data_{age}}{T}, \tag{1}$$

$$data_{age} = \sum_{i=1}^{n} \left(D_{path(i)} + d_{caching(i)} \right), \tag{2}$$

where $data_{age}$ donates the period between the arrival at the ith node (N_i) and the generation at the source. $D_{path(i)}$ and $d_{caching(i)}$ are the time delay on the path between 2 nodes and the period during which the data item reside at an intermediate node N_i, respectively. Freshness value interval is from 0 to 1 ($0 \leq Freshness \leq 1$) as in Figure 2A, where the value 0 and 1 mean that the data item is fully nonfresh or fresh, respectively. When total time delay ($data_{age}$) is larger than the lifetime T, data item will be forwarded to the adjacent node towards the consumers without any caching performance, and in this case, freshness will be a negative value as in Figure 2B.

Like freshness, hop count is one of the important parameters for content cache placement policies. The cache weight increases as the packet is getting close to its destination.[21] In addition, to reduce the data retrieval delay and improve the content dissemination in networks, the closer the nodes to the consumers, the higher probability to cache the data items. The hop count (H_c) is presented path length from the consumer to the source, which is expressed as 3

$$H_c = \frac{c_i}{n}, \tag{3}$$

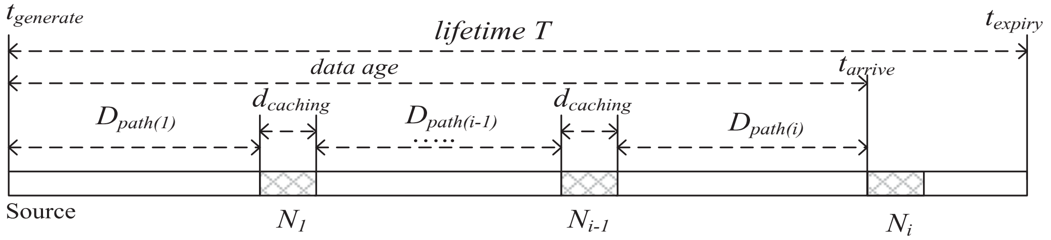
where c_i is the positions of the current node (N_i) on the path from the source towards the consumers and n is the total number of nodes on one transmission path.

In contrast to the traditional networks, IoT is characterized not only by powerful devices but also by resource-constrained devices. In addition, the nodes in CCN can act as the source, which decides to cache the content and forwarding it to consumers. As a result, integrating CCN into IoT environment would have high energy consumption for these constrained devices. While the energy level of the node varies over time, the only node with a high-energy level is considered as a potential entity, which can perform caching of the data item with high priority. By following this assumption, the node will expense a little energy consumption because of overhearing and forwarding operation. Similar to freshness value, to easily evaluate and select the candidate nodes, the energy of the node (E) is also modeled as a normalized parameter ($0 \leq E \leq 1$), where the values 0 and 1 are very weak and full battery, respectively.

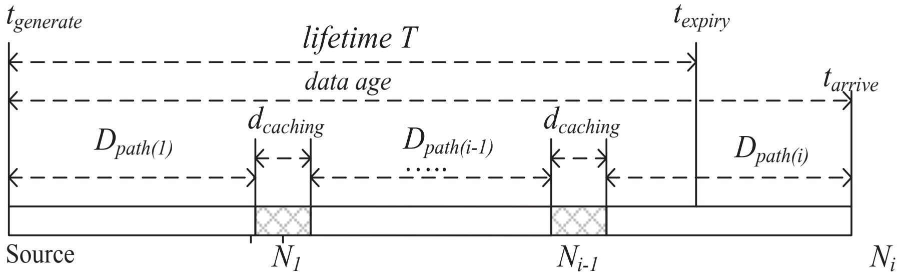
The impact of average cache size availability is measured by the used cache space along the path, expressed as follows 4:

$$Cachesize_{available}(i) = \frac{Cachesize_{total}(i) - Cachesize_{used}(i)}{Cachesize(i)}, \tag{4}$$

where $Cachesize_{total}(i)$ and $Cachesize_{used}(i)$ represent the total cache size and used cache space on the node (i); $Cachesize(i)$ is the total cache space along the path with n nodes, which is available to store until the node (i).[22] The average cache size available is a value that is also normalized ranging from 0 to 1, in which 0 and 1 represent for full and empty cache size, respectively. The remaining parameters used in this paper are listed in Table 1.



(A) Fresh



(B) Non-Fresh

FIGURE 2 Data item freshness

TABLE 1 Definition of parameters

Parameters	Definition	Unit
$t_{generate}$	Data generation time	Seconds
t_{arrive}	Data arrive time	Seconds
t_{expiry}	Data expiry time	Seconds
D_{path}	A path time delay	Seconds
$d_{caching}$	Caching time delay	Seconds
E	Energy level of the node	-

- means that no unit.

3.2 | Multiattribute in-network caching decision

The main objective of this approach is to reduce resource energy consumption, cache redundancy, and efficient caching for transient data by selecting the potential nodes that have strong capability to store the data item. The selected nodes must provide the best caching capability for IoT data items considering a set of attributes including data freshness, the cache size, hop count, and energy levels of the nodes. The following are the steps in this approach;

Step 1. Evaluation of the whole nodes in the transmission path

To enhance the performance of this approach, a preliminary assessment is made as to whether the current node meets caching capability for the running application, which is based on the minimum threshold of different parameters as shown in Table 2. Of these, the minimize threshold of energy level of the node is assumed to be remaining 10% of total energy, which ensuring to maintain for overhearing and forwarding operations. If any one of these parameter value is below its minimum threshold, the candidate nodes/devices just only forward data without caching performance; otherwise, the next evaluation is triggered in step 2.

Step 2. The attributes weight calculations

To optimize the caching decision, the weights are investigated in association with these normalized parameters. According to the main challenging features of IoT such as transient data and resource-constrained devices, it is realized that freshness, energy level of the node, and cache size are more important than the rest of the parameters, which have a decisive impact on the caching performance. The weight vector w_j $(1 \leq i \leq m)$ is calculated in 5 according to the analytical hierarchy process (AHP) approach in Chandavarkar and Guddeti.[23]

$$\mathrm{w}_j = \frac{1}{m} \sum_{p,q=1}^{m} \frac{a_{pq}}{\sum_{p=1}^{m} a_{pq}} \tag{5}$$

$$\mathbf{A_{m\times m}} = \mathbf{a_{pq}} = \begin{bmatrix} 1, & if\, L_p = L_q \\ L_p - L_q + 1, & if\, L_p > L_q \\ \dfrac{1}{L_q - L_p + 1}, & if\, L_p < L_q \end{bmatrix} \tag{6}$$

TABLE 2 The threshold of different attributes

Attribute	Min-Threshold	Unit
Energy level	10	%
Data freshness	0 $(T = Data_{age})$	-
Cache size availability	0 $(Cachesize = Cachesize_{used} full\ storage)$	-
Hop count	0 $(c = 0,$ *supplied by internal cache*$)$	-

- means that no unit.

In the reciprocal matrix A, the pairwise comparison $a_{pq} \in \{1/9, 1/7, 1/5, 1/3, 1, 3, 5, 7, 9\}$ depends on the perceived 1-dimensional vector L_m as defined by the decision maker with the number of attributes (m). The attributes p and q are mapped to any one of the linguistic values $\in\{1, 3, 5, 7, 9\}$ to denote the scale of importance (very low, low, medium, high, and very high), respectively, as shown in Table 3.

After the weight vector is defined, verification of the weight will be implemented with the aim of checking the consistency and reliability of the calculated attribute weights w_j, and this is evaluated using the consistency ratio (CR), which is given by 7.

$$CR = \frac{\text{Consistency Index } (CI)}{\text{Random Consistency Index } (RI)}, \tag{7}$$

$$\text{where } CI = \frac{\lambda_{max} - m}{m - 1}, \tag{8}$$

$$\lambda_{max} = \sum_{j=1}^{m} \left(w_j \times \sum_{p=1}^{m} a_{pj} \right), \tag{9}$$

where RI is the index of the matrix coherence. The value of RI is 0.52, 0.89, 1.11, 1.25, 1.35, 1.40, 1.45, 1.49, 1.51, 1.54, and 1.56, with the different sizes of the matrices (or number of attributes m) being 3, 4, 5 ..., 12, and 13, respectively.[23,24]

According to the calculation above, the weight vector is finally given by w_j = [0.52, 0.20, 0.20, 0.08] as shown in Table 4. Meanwhile, the weights of the freshness, energy levels, and cache size are higher than path-length, which represent the different important levels of attributes in caching performance. The calculated consistency ratio (CR) is very low and is fulfilled by the condition ($CR = 0.02 \leq 0.1$). Therefore, the calculated weight attributes are acceptable, since it ensures a high consistency and reliability of this value.

1. Caching decision making

All attribute values are modeled and normalized (x_j) with the same range [0, 1] that is shown in section 3.1. Generally, content caching score at node N (S_N) is expressed in association with the weight vector in step 2 as follows 10.

$$S_N = \sum_{j=1}^{m} w_j \times x_j(N) \tag{10}$$

A caching decision is based on a comparison between the obtained score value and the defined threshold μ. If score value is greater than the threshold, the content will be stored in the CS; otherwise, it will be only forwarded to the next node. Therefore, the probability of caching the potential nodes on the transmission path will be examined to make a right caching. As a result, the memory of the device and the CS of the node are efficiently used to store IoT data items. The proposed caching procedure is summarized in Figure 3.

TABLE 3 Scale of relative importance

Scale of Importance for Comparison Pair, a_{pq}	Linguistic Value
Very high	9
High	7
Medium	5
Low	3
Very low	1
Intermediate values between the 2 adjacent judgments	2; 4; 6; 8

TABLE 4 Reciprocal matrix using Equations 4 and 5

Attribute	Linguistic Value	*F*	*E*	*CS*	*PL*	Weight, w_j
Freshness (*F*)	7	1	3	3	5	0.52
Energy level (*E*)	5	1/3	1	1	3	0.20
Cachesize (*CS*)	5	1/3	1	1	3	0.20
Path length (*PL*)	3	1/5	1/3	1/3	1	0.08
Total w_j						1.00

4 | SIMULATION AND ANALYSIS

Having described the proposed approach, to illustrate its effectiveness, we first describe the context as well as the simulation setup; we then analyze and evaluate the results that are achieved.

4.1 | Simulation model

The simulation model is implemented using the Optimized Network Engineering Tools Modeler 16.0.[25,26] Content-centric networking is overlaid on top of the IP layer. Indeed, we embedded the CCN processing modules for access point and all intermediate nodes. We assume that the simulation imposes restriction on the intermediate nodes. In fact, they are the physical device used as router, such as the low-cost single-board computer like Arduino, RaspberryPi, or Odroid. To highlight the improvements provided by this mechanism in CCN with IoT environment, the simulated context is specially designed to facilitate the achievement of the research objectives. Hence, multihop path between the number of producers and consumers in the mesh topology is evaluated as in Figure 4. For example, consumers are interested in the latest upgraded content (eg, a hospital needs to update vital signs of a remotely monitored patient's information

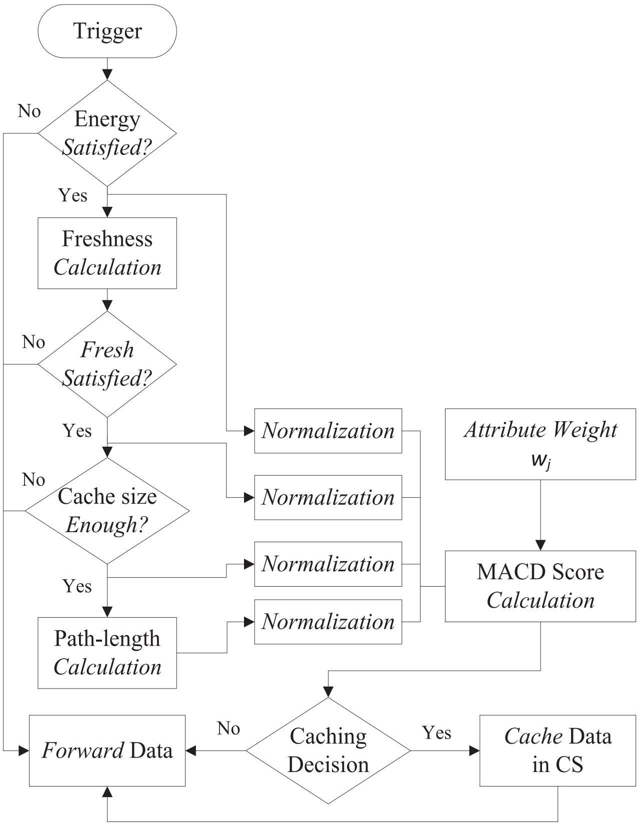


FIGURE 3 Multiattribute in-network caching decision (MACD) flowchart. CS, content store

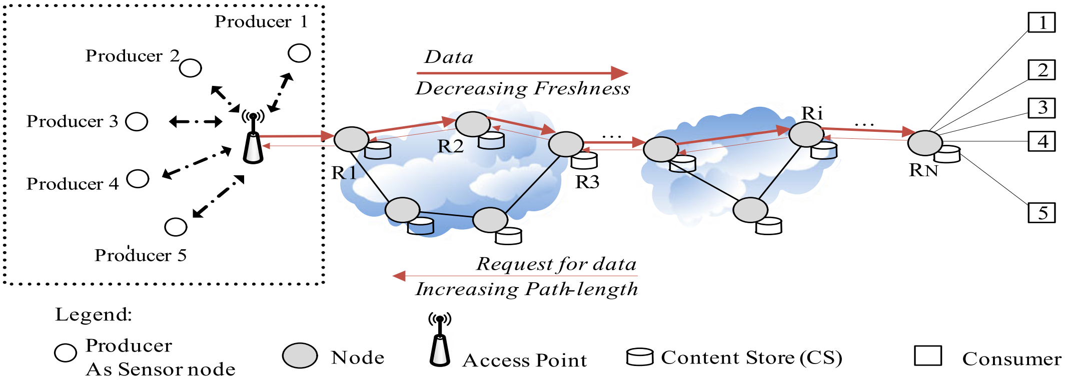


FIGURE 4 Topology of the simulated scenarios

or a display screen periodically asking the latest temperature from sensors of producers). The number of nodes is varied (n = 5, 10, 15, 20, and 25). The producers are equipped with WiFi 802.11b standard that is covered by an access point covering 40 × 50 m. Producers and consumer were set up without caching capabilities to avoid content from being supplied by their internal cache.

In relation to the freshness value, each of the content has a different data item lifetime (eg, $T = 1, 5, 10, 20, 30$ s). The energy level of the node is also varied during simulation time and always guarantees to maintain the basic operation of the node such as forwarding the content (do not run out of energy during simulation). Table 5 lists the rest of the simulation parameters.

4.2 | Simulation results

4.2.1 | Hit rate of the proposed approach in comparison with others

The effect of small cache size in content retrieval is a decisive factor for the accuracy of caching performance in IoT environment. To illustrate the effectiveness of this proposed approach in this respect, we perform an evaluation of the average hit rate on the path (is a fraction of total hit rate and total requests from customers) in comparison with 2 caching polices: Prob-LRU and typical AlwaysCache-FIFO.

Cache hit rate is the probability to achieve a cache hit from CS of vehicles instead of the original servers, which is defined as the fraction of cache hits to total number of request messages. In Figure 5, it can be observed that the average hit rate of the nodes on the path in this approach is higher than Prob-LRU and AlwaysCache-FIFO scheme. This circumstance is caused by the fact that some key attributes such as small cache size, hop count, and the node energy level

TABLE 5 Simulation parameters

IoT data item	Interest packet (*IntPk*) data size (*DataPk*)	32 byte 256 bytes
Scenario	Start time–stop time	50 + Uniform(0,10)s − 200 s
	Number of nodes	5-25
	Number of producers	5
	Number of consumers	5-20
Links	D_{path}	10 ms
CCN	Caching policy	Always-FIFO, Prob-LRU, MACD
	Cache size	1;2;4 KBytes
Threshold, μ	Value	0.6
Power transmission	Access point	27 dBm
	WiFi, 802.11b	

Abbreviations: CCN, content-centric networking; FIFO, first in first out; LRU, least recently used; MACD, multiattribute in-network caching decision.

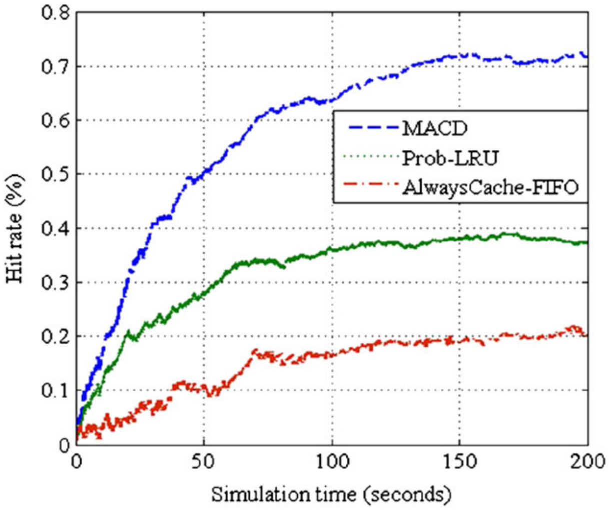


FIGURE 5 Comparison of hit rate for different approaches. FIFO, first in first out; LRU, least recently used; MACD, multiattribute in-network caching decision

are often invalid by using Prob-LRU where the data freshness and the rate of incoming requests only were considered, especially AlwaysCache-FIFO policy that leads to inefficient storage and high cache redundancy.

4.2.2 | The freshness evaluation

Since the freshness is one of the stringent requirements in IoT, changing its value leads to the various effectiveness of caching decision. To evaluate the impact of this metric, the rest parameters are fixed (eg, hop count and cache size). The energy of the node is always set at a high level during the simulation time while increasing the number of consumers. The data item lifetime of the content is set to 5 seconds and changes the time delay in ascending order. In this way, an amount of content in the total data received by the node will not be satisfied by the freshness condition. Figure 6 shows that with the optimization of cache size used, the percentage of reused content of MACD approach is always higher than Prob-LRU and AlwaysCache-FIFO (reaches approximately to 79.1% to reuse content in case 20 consumers request data item). This result is caused by that fact that, when the caching decision performance is only based on the changed freshness parameter, the frequently updated data items with a fresher value in the proposed approach are easy to satisfy the consumer's request. The LRU policy is likely to be matched with fresher content by removing the LRU packets from the CS, but much less effective in the case of very stringent freshness requirements. Severely, the nodes often cache the expired data item with the AlwaysCache-FIFO policy that leads to waste the CS capacity and inefficient use of IoT network resources.

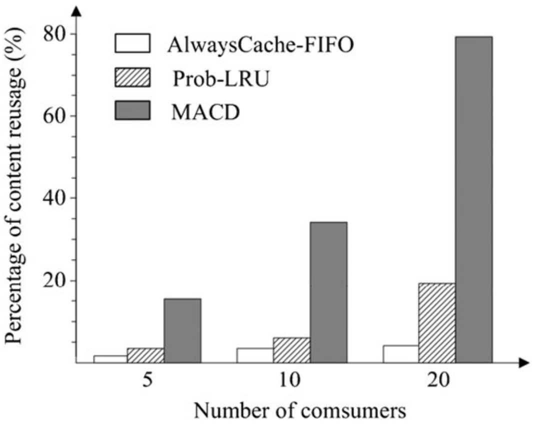
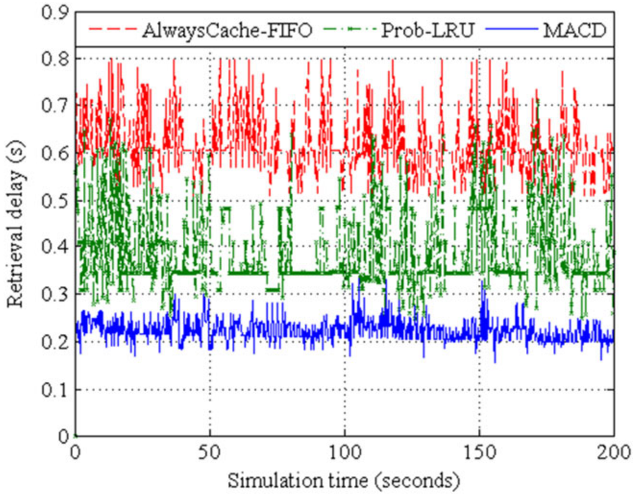


FIGURE 6 Comparison of content reusability with proposed approach and others. FIFO, first in first out; Prob-LRU, probabilistic least recently used; MACD, multiattribute in-network caching decision

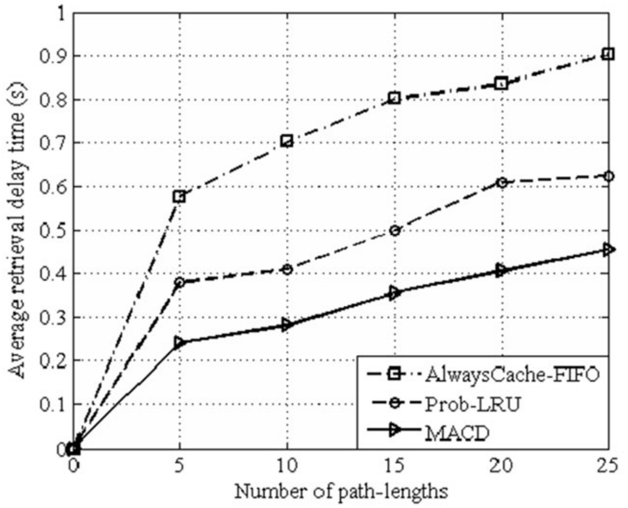
4.2.3 | Data retrieval delay evaluation

In IoT, the cached content is often discarded because of short data item lifetime and long delay time. Thus, it is important to investigate the data retrieval delay, which is defined as the time between sending and receiving of the IntPk data item by the consumers. The effective caching in powerful nodes allows increasing the possibility to respond to the consumer requesting, in which the content is frequently updated based on the freshness value. Moreover, the proposed approach considers the node energy level and cache size, which does not only avoid running out of energy for the weak battery of the nonpotential nodes but also decrease delay time of the unnecessary caching performance. Hence, it reduces the data retrieval delay time significantly as shown in Figure 7A.

Furthermore, the impact of multihop communication in caching performance is also investigated to clear the superiority of the proposed approach in this respect. To evaluate this metric, the number of nodes on a path is changed to simulate the varying path length. For each path length, we have measured the delay time, which is computed as the time since the consumer sent the interest to the reception data item. In particular, the hop count (or path length) along the transmission path is taken to be directly proportional to the delay time, including propagation and caching delay. The increasing of the path length will spend more time delay, which is a direct cause leading to lessening the freshness. Beside the facts as explained in Figure 7A, considering the node closer to the consumer with reducing the time delay in the proposed approach would increase the content reusability in the potential nodes. Therefore, the average data retrieval delay time is significantly reduced in comparison with the rest of others schemes as shown in Figure 7B.



a. Data retrieval delay comparison for different approaches



b. Average data retrieval delay when varying the number of path-lengths

FIGURE 7 (A) Data retrieval delay comparison for different approaches and (B) average data retrieval delay when varying the number of path lengths

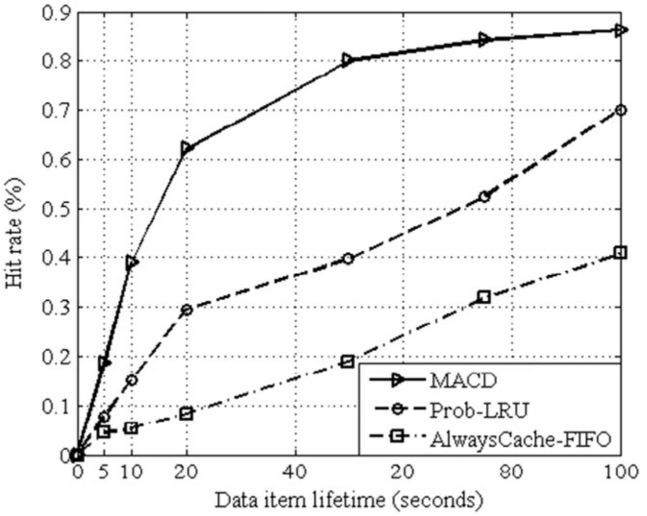
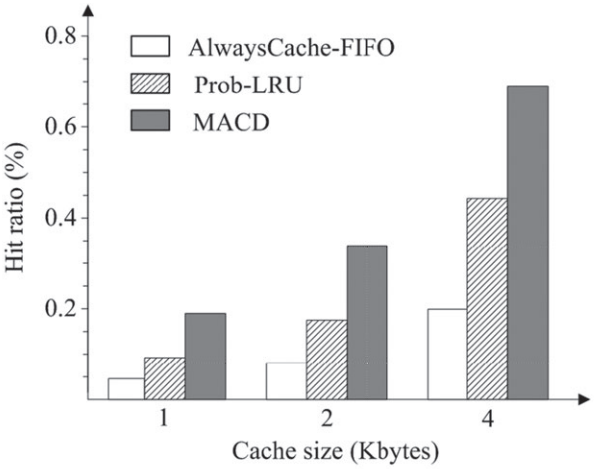
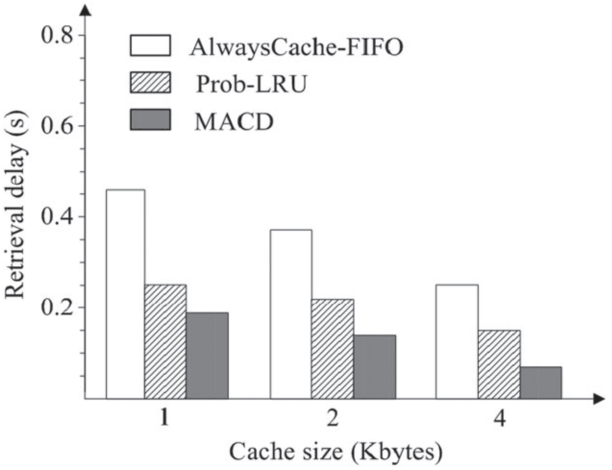


FIGURE 8 Effect of data item lifetime on hit ratio. FIFO, first in first out; Prob-LRU, probabilistic least recently used; MACD, multiattribute in-network caching decision



(A) Cache hit ratio



(B) Data retrieval delays

FIGURE 9 Comparison of hit ratio and retrieval delays for different approaches when varying cache size. FIFO, first in first out; Prob-LRU, probabilistic least recently used; MACD, multiattribute in-network caching decision

4.2.4 | Impact of the data item lifetime on hit rate, retrieval delay, and interest retransmission

To further investigate the effect of the key temporal property in IoT on the different caching policies, the changes with respect to lifetime are installed in terms of the values ($T = 5$, 10, 20, 50, 75, and 100 s), meanwhile keeping stability for the rest of the parameters such as energy of the node, the number of path-lengths, and the number of consumers. As a result, when the data item lifetime is lower, cache hit ratio of MACD approach shows more superior respect to others scheme, as shown in Figure 8. These superiority trends decrease with increasing the data item lifetime. That is to say, the proposed approach strongly illustrates the preeminence of in-network caching for IoT in case stringent requirements vary over time. This is due to the fact that the IoT items quickly expire with low lifetime value; even they expire in the case of transmission in Figure 2B (even in some case, they are expired during transmission). Hence, expired data items are no longer useful and must be discarded in MACD approach meanwhile they may be stored in the CS by Prob-LRU and AlwaysCache-FIFO scheme.

4.2.5 | The impact of cache size on hit rate and retrieval

The presence of resource-constrained devices in IoT with little-to-none caching capabilities affects greatly the efficiency of in-network caching, even though cache size is a stable attribute and provides the versatile caching capability by CCN scheme. Figure 9 shows the performance metric when varying the cache size that is only a few KBytes. The small cache size results in low performance of all investigated caching strategies (eg, in the case hit ratio of AlwaysCache-FIFO is slightly increased from 4.5% to 20% with the cache size varying from 1 KByte to 4 KBytes, while Prob-LRU from 9% to 43% [Figure 9A]). Certainly, the node/devices decrease the capability to store the content that leads to low response probability due to small cache size, thus lengthening the retrieval delays (Figure 9B).In the presence of small cache size, the negative effects of Prob-LRU are even exacerbated with respect to AlwaysCache-FIFO strategy. Vice versa, the proposed approach improves the data diversity in the network and provides a more efficient use of available cache resources by discarding nonfresh packets.

5 | CONCLUSION

In this paper, a novel efficient caching decision approach is proposed for IoT with CCN integration. In particular, this paper proposed the effectiveness of a caching approach by considering all the important attributes associated with their weights to select powerful nodes/devices for content storage. The content will be stored at the powerful nodes if higher freshness, a higher level of the node/device energy, larger CS size, and closer to the consumer are considered. The obtained results illustrate the remarkable effectiveness of the proposed approach in terms of improving total hit rate, reducing data retrieval delay, enhancing content reusability, and alleviate the load on the node/constrained devices in the IoT domain, which is very useful in the challenging IoT environment. As for future work, it is planned to extend the simulation scenario by considering mobile nodes and different content request in other applications. Additionally, we plan to precisely investigate the relative importance of attribute weights and expand the calculation of the weight vectors for each popular IoT application.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the National Natural Science Foundation of China (grants 51675389 and 51475342).

ORCID

Dong Doan Van http://orcid.org/0000-0003-1968-0870

REFERENCES

1. Whitmore A, Agarwal A, Li DX. The Internet of things—a survey of topics and trends. *Information Systems Frontiers*. April. 2015;7(2): 261-274.

2. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A survey on Internet of things: architecture, enabling technologies, security and privacy, and applications. 4662(c). *IEEE Internet of Things Journal*. 2017;PP(99):1-17. https://doi.org/10.1109/JIOT.2017.2683200
3. Correia N, Mazayev A, Sch G, Martins J, Barradas A. Resource design in constrained networks for network lifetime increase. 4662(c). *IEEE Internet of Things Journal*. 2017;PP(99):1-15. https://doi.org/10.1109/JIOT.2017.2678517
4. Sheng Z, Yang S, Yu Y, Vasilakos AV, Mccann JA, Leung KK. A survey on the IETF protocol suite for the Internet of things: standards, challenges, and opportunities. *IEEE Wireless Commun*. 2013;20(6):91-98.
5. Zhang L, Estrin D, Burke J, et al. Named data networking (NDN) project. PARC, *Tech. Rep. NDN-001*, October 2010.
6. Parisis G, Sourlas V, Katsaros KV, Chai WK, Pavlou G, Wakeman I. Efficient content delivery through fountain coding in opportunistic information-centric networks. *Computer Communications*. March. 2017;100(1):118-128.
7. Dong DV, Qingsong A, Dung OM. MS-CCVN: multi-source content centric vehicular networking. *J Commun*. December. 2016;11(12):1079-1088.
8. Ahmed SH, Bouk SH, Kim D. Content-centric networks: an overview, applications and research challenges, 2016. https://doi.org/10.1007/978-981-10-0066-9
9. Fang C, Yu FR, Huang T, Liu J, Liu Y. A survey of green information-centric networking: research issues and challenges. *IEEE Communications Surveys & Tutorials*. 17(3):1455-1472. thirdquarter 2015
10. Zhang G, Li Y, Lin T. Caching in information centric networking: a survey. *Comput. Netw*. 2013;57(16):3128-3141, November.
11. Kim Y, Yeom I. Performance analysis of in-network caching for content-centric networking. *Computer Networks*. September. 2013;57(13):2465-2482.
12. Feng BH, Zhou HC, Feng B, Zhou H, Gao S, You I. An exploration of cache collaboration in information-centric network. *International Journal of Communication Systems*. 2014;27(9):1243-1267.
13. Yan H, Gao D, Su W, Chao H. A forwarding strategy of counteracting redundancy data in named data networking. *International Journal of Communication Systems*. 2015;28(18):2289-2310.
14. Song Y, Ma H, Liu L .Content-centric internetworking for resource-constrained devices in the Internet of things. *In: IEEE International Conference on Communications (ICC)*, pp. 1742-1747, June. 2013.
15. Vural S, Navaratnam P, Wang N, Wang C, Dong L, Tafazolli R .In-network caching of internet-of-things data. *In: IEEE International Conference on Communications (ICC)*, pp. 3185–3190, June. 2014.
16. Hail MAM, Amadeo M, Molinaro A, Fischer S. On the performance of caching and forwarding in information-centric networking for the IoT. *In: Aguayo-Torres M., Gómez G., Poncela J. (eds) Wired/Wireless Internet Communications. WWIC 2015. Lecture Notes in Computer Science*, vol. 9071, 2015, Springer, Cham.
17. Yin J, Alvisi L, Dahlin M, Iyengar A. Engineering web cache consistency. *ACM Trans. Internet Technol*. August 2002;2(3):224-259.
18. Baccelli E, Mehlis C, Hahm O, Schmidt TC, Wählisch M. Information centric networking in the IoT: experiments with NDN in the wild. *ACM conf. Information-Centric Networking*, 2014.
19. Quevedo J, Corujo D, *Aguiar R*. R. Consumer driven information freshness approach for content centric networking. *IEEE Infocom NOM Workshop* 2014.
20. Vural S, Wang N, Navaratnam P, Tafazolli R. Caching transient data in Internet content routers. *IEEE/ACM Transactions on Networking*. September 2016;(99):
21. Chai WK, He D, Psaras I, Pavlou G. Cache “less for more” in information-centric networks (extended version). *Computer Communications*. April 2013;36(7):758-770.
22. Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information-centric networks. *In Proc. 2nd Ed. ICN Workshop Inf.-Centric Netw*, pp. 55–60, 2012.
23. Chandavarkar BR, Guddeti RMR. Simplified and improved analytical hierarchy process aid for selecting candidate network in an overlay heterogeneous networks. *Wireless Personal Communications*. 2015;83(4):2593-2606.
24. Tzeng GH, Huang JJ. *Multiple attribute decision making: methods and applications*. Boca Raton: *CRC Press, Taylor &Francis Group*; 2011.
25. OPNET Modeler. Available: www.opnet.com
26. Chen M. Opnet network simulation. *Press of Tsing hua University*, ISBN 7-302-08232-4, 2004.

How to cite this article: Doan Van D, Ai Q. An efficient in-network caching decision algorithm for Internet of things. *Int J Commun Syst*. 2018;31:e3521. https://doi.org/10.1002/dac.3521