IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2020, accepted July 5, 2020, date of publication July 9, 2020, date of current version July 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008193

Energy-Efficient Centrally Controlled Caching Contents for Information-Centric Internet of Things

HAMID ASMAT[1,3], FASEE ULLAH[2,3], MAHDI ZAREEI[4], (Member, IEEE), ATIF KHAN[5], AND EHAB MAHMOUD MOHAMED[6,7], (Member, IEEE)

[1]Department of Information Technology, University of Haripur, Haripur 22620, Pakistan
[2]Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China
[3]Department of Computer Science and IT, Sarhad University of Science and Information Technology, Peshawar 25001, Pakistan
[4]School of Engineering and Sciences, Tecnologico de Monterrey, Zapopan 45201, Mexico
[5]Department of Computer Science, Islamia College Peshawar, Peshawar, 25120, KP, Pakistan
[6]Electrical Engineering Department, College of Engineering, Prince Sattam Bin Abdulaziz University, Wadi Addwasir 11991, Saudi Arabia
[7]Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan 81542, Egypt

Corresponding author: Fasee Ullah (faseekhan@gmail.com)

This work was supported by the School of Engineering and Sciences, Tecnologico de Monterrey.

ABSTRACT The IoT devices produce a massive amount of content compared to the host-centric (IP-based) network. The IP-based network has no strengthened capabilities to handle the growing traffic of the IoT network. The existing research is focusing on proposing Information Centric Network (ICN) framework with potential replacement of IP-based network. Using the caching technique in ICN based IoT significantly strengthen the resource-constrained IoT devices, by reducing the access time with optimized energy consumption. Thus, this paper proposes a caching scheme for IoT contents, known as Central Control Caching (CCC) Scheme. This scheme positions a central entity between different autonomous systems that manage caching and updates the content, by keeping information in a table regarding different contents present on different autonomous systems. This information is utilized when cached contents are exchanged among different autonomous systems. Furthermore, when contents becomes stale, the same information is used to identify the autonomous systems with a copy of the outdated content. The simulation results show that the proposed scheme has outperformed in terms of reduced energy consumption, response latency, and cache-hit compared to state-of-the-art works.

INDEX TERMS Caching, Internet of Things, information centric network, update.

I. INTRODUCTION

The Internet of Things (IoTs) connect a lot of miscellaneous devices for remotely operating and information exchange like smartphones, smartwatches, washing machines, smart refrigerators and smart vehicles [1], [2]. The internet will connect 3.9 billion devices by the end of 2022, which is 32 percent increase in devices as compared to 2017, stated in the Cisco visual networking [3]. This dramatic increase in number of IoT devices will increase IoT contents, which cannot be efficiently handle by the existing IP-based Internet. Moreover, the existing IP-based network has limited number of IP addresses and is not able to scale up the network by assigning IP addresses to every IoT devices for accessing and sharing information on network [4]. These IP-based Internet limitations degrade the performance of IoT devices caused by high energy consumption and delay in response [5]. Hence, the research community is keen on finding solutions to the problems by applying Information-Centric Network (ICN), which is the alternative way for IP based internet and is considered a suitable candidate for IoT networks [6].

ICN is a content-centric networking architecture which uses name rather than its location for accessing the content. This means that the contents are accessed with their names regardless the concept of IP addressing in this architecture. Therefore, this ICN approach is feasible for IoT networks with features like simplified content access and distribution, content-based security, mobility, in-networking

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman.

caching [7], [8]. These features are beneficial for all networks, but we choose to focus on in-network caching. ICN caches (temporary buffer) the contents in between the publisher and the subscriber on the intermediate nodes of network for easy access with benefits such as reduced content retrieval time, reduction in requirements for bandwidth, reduction in number of accessing towards subscribers, and reduction in energy requirements of IoT devices.

ICN was designed for the traditional Internet, where content is static for a long time without modification / updation. However, IoT-based content is different and frequently created and modified at some intervals. These contents are known as transient contents [9]. Our scheme uses transient IoT content that loses freshness very rapidly and become outdated. ICN caching in IoT, faces three major challenges: finding cache, updating content and reducing access time to IoT publishers. It is essential keep cached content at an optimum location because it defines network efficiency. The caching of content node, such as LCE [10], is not feasible as it increases content redundancy. In [11]–[13], the authors proposed caching of contents on fewer nodes or in proximity to the subscribers to reduce redundancy and response latency. In this paper, we select a node connected directly to the subscriber that can cache the contents. Updating the cached contents with low overhead is also important for ICN-based IoT networks. Moreover, the reduction of total number of accesses towards the publisher will reduce the total energy requirements of the IoT publisher. Various caching schemes for traditional internet content have been developed that are large, static and not periodically updated. These schemes can not handle transient IoT contents, as they do not take freshness into account in caching the content, energy and memory resources.

several studies have been focused on implementing ICN caching in IoT. For example, Pre-caching strategy based on content relevance is proposed in [14]; A Deep learning based ICN-based IoT caching is implemented in [15]; [16] proposes clustering based IoT Caching; a probabilistic caching is implemented in [17]; [18] proposes a probabilistic caching technique known as Pcasting for wireless network; [19] discusses the freshness of the IoT contents; a sleep-based energy efficient caching scheme has been implemented in [20]; a lifetime-based caching scheme is implemented in [21]. Several other paper have been discussed in related works section. These studies have implemented caching in IoT, but in most of these studies the cache hit is lowered, response latency is increased because the content is retrieved from the IoT devices, which means making these devices less energy efficient. Furthermore, no information have been given about their implementation on the network with autonomous systems. on the hand, our proposed scheme implements caching in autonomous systems, in a way that a network with multiple autonomous system is aware of the contents cached in it. This content awareness will increase the cache hit, response latency and energy efficiency of the IoT devices.

To date, there are different architectures employed to implement ICN that are: NDN, COMET, CONVERGENCE, CCN, MOBILITY FIRST [22]. However, the CCN architecture [6] is selected for improvement of the proposed framework of this paper because the most existing electronic gadgets work on caching that are computer machines, smartphones, and, sensor nodes [23]–[25]. CCN is a content-oriented framework, which cache the content with a hierarchical naming approach. Hence, these aspects of ICN can be seen useful for IoT devices with limited resources, hence, we choose CCN architecture for deploying a caching scheme for a resource constrained IoT environment.

This paper proposes an energy efficient Central Controlled Caching (CCC) scheme. The location of the caching node is an intermediate node that is connected to the subscriber. Furthermore, this scheme also implements a Network Content Controller (NCC) that will control the flow of cache contents between different Autonomous Systems (AS). This entity will keep a record of all the contents cached on different ASs'. All the interest generated from any network will go through NCC. The nodes in the proposed scheme maintains a content Information table (CIT). This CIT table updates the route information using broadcast approach such as a Link State Routing table [26], [27]. Moreover, this table will also maintain information of the contents caching of different nodes included content name and port number for efficient access with the assistance for easier finding content within a network. Every generated content will have certain amount of time for the next update. When this content reaches NCC, a timer is set for forthcoming update of content. For the eviction of content, this scheme uses the least frequently used (LFU). The proposed scheme reduces the energy requirement hence making IoT devices more energy efficient. This is done by reducing the number of interest towards IoT devices, because in IoT nodes transmission module consumes more energy as compared to other parts [28], [29]. Moreover, this scheme will increase the cache hit on the network and decrease response latency due to location of the contents is known via CIT table.

The contributions of this paper are:

- CCC scheme will make IoT node more energy efficient by decreasing the number of access time.
- CCC deploys NCC node, which has NCT table holding information of the contents cached on different AS. This table will manage the interest for contents present on different AS or IoT.
- A CIT table is introduced, which makes the nodes informed of the contents cached inside AS. This will make retrieval fast of the content cached on other nodes inside AS. Also, the location information will always be available with the cache node.
- A centrally controlled update is deployed by CCC, in order to reduce the overhead of updating multiple cached copies present on different AS.

- CCC is compared with LCE [10], LCC [21], and IoT-Cache [30] and the simulation shows that the proposed scheme performs better than the exiting schemes in-terms of energy Consumption, response latency and cache-hit.

The rest of the paper is structured as follows. ICN caching and its benefits to the network are discussed in section II. Section III addresses some of the previous work for caching IoT contents. A new caching technique for IoT contents is discussed in section IV, simulation and results are explained in section V, and conclusion is given in section VI.

II. CACHING AND ITS BENEFITS

The most significant feature of ICN is caching by improving the content distribution in network. This improvement decreases the access time in retrieval of the content distribution which efficiently impacts in energy reduction for IoT devices [31]. Moreover, the caching also reduces the response time of the content by placing the content near to the subscriber reducing bandwidth requirements. However, the caching contents is not suitable for caching IoT content without considering freshness parameter [32]. There are several caching schemes suggested for cache transient IoT contents that offers ICN for future Internet with benefits. The following benefits efficiently strengthen IoT environments.

A. SERVICES SEPARATION OF THE PUBLISHER AND SUBSCRIBER

If there is any interest made by subscriber for contents directly affect the publisher. This massive traffic generation from various subscribers lead complications in unusual traffic flow in the network with inefficient utilization of bandwidth [22]. It is observed that IoT devices with limited energy resources cannot accommodate the large number of interests. Thus, the implementation of caching on the intermediate nodes reduces traffic burden flow with the satisfied subscriber's interests in future.

B. REDUCTION OF CONTENT RETRIEVAL TIME

The caching service provides content copy near to the subscriber via use of the intermediate nodes, which reduces the retrieval access time.

C. CONTENT ACCESSIBILITY

Using ICN caching, the accessibility of the content can be increased by placing copies of single content on different location of the network, which will make it easier for subscribers to retrieve the content from the nearest location. Furthermore, if a copy of a content is evicted from one location it can be retrieved from another location on the network. This makes the accessibility of content in ICN better than the conventional Internet.

D. REDUCED ENERGY REQUIREMENTS

Most of the IoT devices are battery powered. This makes energy a very important performance parameter when designing a protocol for IoT. Energy requirements of IoT devices are decreased by caching, wherein the contents are present on the nodes near the subscriber. Therefore, the caching will effect the total battery life of IoT network.

E. HIDING PUBLISHER FROM INTRUDERS

Caching hides the publisher from the intruder because every time a request is made for a content, the intermediate node provides the content, irrespective of where the publisher is located. This increases the user privacy because both subscriber and publisher does not need to be aware of each other.

F. MOBILITY

Mobility is the feature supported by ICN [33]. Most of the IoT devices are mobile devices. If a mobile node is accessing a content from one cache node while moving, and suddenly it loses connection from the cache. In this situation, the remaining content will be accessed of the upcoming cache of a node.

G. BANDWIDTH UTILIZATION

The efficient way of the Bandwidth utilization is the most important resource of network. However, the caching can reduce the need of bandwidth due to the contents are mostly available from a cache by a near node. This makes the use of bandwidth efficiently in ICN compared to Conventional Internet.

III. RELATED WORK

Several caching schemes are discussed in [32]. These schemes are suitable for conventional internet contents which are stored for a longer duration on servers and are infrequently updated. Applicability of these schemes on transient contents generated by IoT devices is not suitable because they do not regard freshness and energy requirement of the IoT. Schemes for Caching the IoT contents [7], [34], [35] should be energy efficient because of the restricted energy resources. Furthermore, these schemes consider updating of transient content regularly but with minimum overhead. Several caching techniques are explained below.

The first-ever study that discussed caching of transient IoT content is presented in [9]. In this authors have used the hop count between publisher/subscriber and freshness of content for calculating the caching probability of the content. Network load is reduced using this scheme but it fails to provide any method for energy efficiency. The authors of the previous study also proposed the use of internet content routers for caching transient IoT contents in [36]. Caching of content is done by applying a trade-off between multi-hop and freshness. Similarly, a consumer-driven freshness is proposed in [19]. In this the publisher assigns a freshness value to the content which is based on subscribers required freshness. Thus freshness is controlled by the subscriber which can be considered as an advantage of this scheme. The downside of the proposed scheme is that the freshness requirements of the different subscriber for the same content may be different so content downloaded once may not be suitable for another subscriber, hence most of the request made by subscribers will be forwarded to the IoT device for a fresh content. This will reduce the cache-hit, and increase the energy requirements of the IoT devices. Moreover, in [18], a probabilistic caching scheme (Pcasting) is proposed for the IoT environment. In this scheme, the caching decision is based on residual energy and total remaining memory. This scheme works irrespective of routing protocols. Pcasting reduces IoT energy consumption but suffers from redundancy. Furthermore, a sleep based caching scheme is studied in [20]. sleep/wake cycle is assumed for IoT nodes in the proposed scheme. Caching of contents is only done when the node is in wake mode. In the proposed scheme the energy is conserved by keeping IoT nodes in sleep for a larger part of the time. This scheme increases the total lifetime of the network but it also increases the processing overhead because of two different replacement techniques used by the scheme,which are Max Diversity Most Recent (MDMR) and Priority-MDMR. Moreover, a very strict sleep wake cycle is needed for the prpopsed scheme. In another study [37], the author suggests a benefit based caching distributed algorithm (BCDA). In this scheme, a subset of the nodes is selected to cache a content. With this prior knowledge of the location, the content can be easily updated. The scheme shows content consistency but has problems with processing load which reduces its energy. In another research [38], a smart caching scheme is proposed. This scheme uses a smart table to store information about the contents, such as the total number of interests and the current freshness of the content. This scheme only caches content with high interests and freshness, however every node will cache the content increasing redundancy. In a recent study [17], a probabilistic caching scheme for IoT network is proposed. This scheme uses IoT devices as caching helpers alongside the edge servers. IoT device will cache the contents and serve limited subscribers. This proposed scheme will improve the response latency because a cache helpers will be located near to the subscriber, but it will reduce the cache hit and energy efficiency because if a subscriber requests for a content from a cache helper which is already connected to the maximum allowed subscribers, it will do a cache-miss and the request will be sent to the IoT publisher, this may increase the number of accesses making these IoT publishers less energy efficient. Moreover in [15], a Federated deep reinforced learning based cooperative edge caching (FADE) is proposed. In FADE, intermediate nodes are trained so that it can handle the problem of duplicate traffic on the network. The advantage of this scheme is that it reduce the response latency because of fast convergence. However, the training of nodes requires excessive resources, and attribute on which the training is done changes quickly, so constantly training the nodes is required which reduces the energy efficiency of the network. Furthermore, a Pre-caching scheme for Information Centric IoT in [14]. The proposed scheme splits the content in to chunks distribute it on different intermediate nodes. Every intermediate node is aware of all the chunks which needs to cache or forwarded to the downstream. Popular chunks are pushed towards the edge of the network. The drawback of this scheme is the that chunks of a single content are places on different nodes due to which the response time of the content is increased because the collection of all the chunks in to a single content will increase delay. In another study [16], a Packet Update Caching (PUC) Scheme is proposed. The proposed scheme uses clustering. The strategy's caching position is a cluster head. In addition to caching, purging techniques are used to delete content and the content is secured via encapsulation. The benefit of this strategy is that it distributes content efficiently to mobile consumers. The problem lies in the update of content because this strategy checks for the updated content on the Publisher which increases the number of accesses making the IoT devices less energy efficient.

In [39], an energy-aware and latency Guaranteed dynamic resource caching (EASE) scheme is proposed which only cache popular content. According to the authors caching of popular content makes the IoT devices energy efficient. This scheme can effectively manage popular contents but it may not cache a slightly less popular content, which will in turn make the IoT devices less energy efficient and decrease cache-hit because these slightly less popular contents will then be retrieved from the IoT devices making them less energy efficient. In another study, an IoT lifetime-based cooperative caching (LCC) [21] scheme is proposed. This scheme uses content lifetime and interest rate for making a caching decision. For adjustment of content threshold according to user interest, an auto configuration method is used. The downside is that intruder scan falsify the interest rate making an unpopular content popular and this scheme also suffers from redundancy. In a recent study, a cooperative update mechanism is proposed [40]. This approach uses a timer to update the content cached. Also, if multiple nodes want the same content to be updated, a cooperative caching will is used to add the update confirmation message from every node with the same content. The downside to this strategy is that it confirms content updates even before the content expires, which increases the number of accesses towards IoT nodes making this scheme energy greedy. Furthermore, in [30], an IoTCache, scheme is proposed. In this scheme the authors pre-fetches and evicts the content based on the predicted popularity of the content. The proposed scheme effectively calculate the popularity of the content with a given dataset. The downside of this scheme is that, in case of of multiple popular contents, which content will be evicted from the cache. Furthermore, due to an increasing dataset, calculating the popularity will utilize more resources such as energy and memory.

IV. THE PROPOSED SCHEME

This paper proposes an energy efficient caching scheme for transient IoT contents, known as Centrally Controlled Caching (CCC). This scheme reduces the number of accesses towards the IoT devices making them more energy efficient. The proposed scheme divides the network into autonomous systems (ASs). These ASs are connected with a central

Network Content Controller (NCC) node. For connecting NCC and AS we apply a T1 line, which has constant data rate of 1.544 Mbps. NCC manages the retrieval of contents from IoT nodes and AS, update cache contents, and is keeping track of cached contents on different nodes. The caches keeps contents inside AS, which are connected to the subscriber. All the nodes present in AS will have a Content Forwarding Table (CFT). This CFT table keeps the location information of the contents inside AS. Additionally, the proposed scheme update the existing stored cache content. The content update is performed via a NCC node.

To explain the proposed scheme, lets assume a subscriber connected to AS 1 and sends an Interest for some content *C*. When the intermediate node in AS 1 receives the Interest, it first checks its own content store (CS). If the content *C* is available it will reply back with the content. If the content is not available in the CS, it will check the availability of the content in the AS 1 and for this the node checks its CIT table, if there is an entry for the content, the node will forward the Interest to the node which already has cached that content *C*. If the CIT does not have any record of the content *C*, then the Interest is forwarded to the NCC through the gateway node. NCC will look for the content in different ASs' in the network using it network content table. If the content is found in any AS for example content is cached on some node in AS 2. The NCC will extract the content from AS 2 and forward it to the AS 1. If the content is not cached in any AS then NCC will download the content from the IoT nodes or the publisher and forward it to AS 1.

The proposed scheme uses CCN architecture. However, its default setting is unable to address the content freshness and energy consumption of IoT nodes. For this purpose a new structures known as CIT table is added to the intermediate node along side the default FIB, PIT and CS. CIT makes the node aware of the content location inside an AS. Unlike, FIB which maintains path which will take the Interest to the data source, CIT maintains the information of the cached contents inside an AS, which makes the node aware of the content available in the AS, such as if a subscriber makes a request for a content cached on another node inside an AS, the node will use this CIT table to identify the location and retrieve the content.

The proposed scheme content. is explained in the following.

A. NETWORK CONTENT CONTROLLER (NCC)

The NCC node Performs several essential functions, as described in the following.

- NCC retrieves contents from IoT nodes. Any interest made for a content which is not cached on any AS will go through the NCC. It will retrieve the content from IoT nodes, and will send it to AS from where the interest has generated.
- NCC maintains a network content table (NCT). The NCT table keeps the information of all the cached contents in AS of the network, as shown in Table 1. This information in NCT table includes type of IoT device, name of AS in which content resides, port number through which the AS in accessible, total lifetime for which the content will be considered valid (assigned by the publisher), and a count down timer. The count down timer is set according to the Lifetime, for instance if lifetime of content is 300 secs, the timer will also be set to 300 seconds. The NCT information is used when an interest is made for a content which is not cached in AS, but it is present in another AS. This interest will reach NCC, where it will use the NCT information and will retrieve the content from a particular AS and forward it to the target AS. After the content is forwarded to the target AS, NCC will generate an entry in the NCT table for the transmission of content as shown in the second entry of Table 1, in which content P is again entered in table with different AS and port number. The timer for this entry will be set to the remaining time of the lifetime such as the remaining time is 150 sec as shown in Table 1. Furthermore, if there is no record of the content in the NCT, only then the interest will be forwarded to the IoT publisher. This table will assist in reducing the number of Interests forwarded towards IoT publisher.

TABLE 1. Network Content Table.

type	AS ID	Port#	Lifetime	Timer
temperature	AS1	1	5 secs	300 secs
temperature	AS3	3	5 secs	150 secs
humidity	AS2	3	10 secs	600 secs
humidity	AS3	5	10 secs	600 secs

- NCC is also responsible for updating the content in network. When the countdown timer reaches zero, then it is the function of NCC to initiate and update the content on different ASs' of network.

B. THE CONTENT

The contents in the proposed scheme are periodic in nature, which needs to be updated regularly. The content uses the default hierarchical naming structure of CCN architecture. This naming includes the name of IoT type, lifetime of the content for example *IoTtype/temperature/lifetime/300*. The example shows a type of IoT device from which a content is being retrieved. The lifetime of this content is 300 seconds, which mean that after this lifetime the content will be considered as stale, and a new content will be downloaded from the producer. When an updated content is downloaded from the producer it will have the same name, because the new copy will only have the new temperature values. When ever an NCC downloads a content from the Publisher it will assume it as an updated content and after download it will be forwarded to the cache nodes inside the AS, where it will simply replace the existing content with the new one.

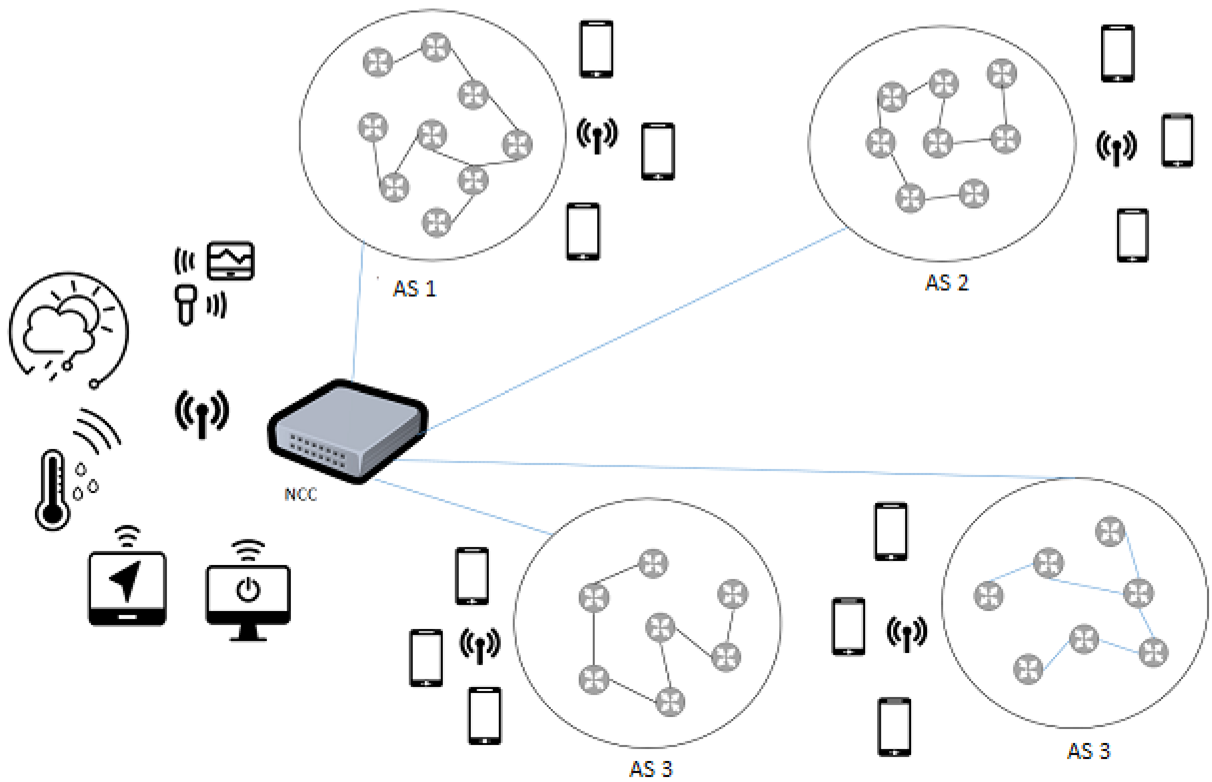

FIGURE 1. Network with multiple Autonomous System and Central Controller.

C. AUTONOMOUS SYSTEM (AS)

The proposed network of CCN architecture is divided in different ASs', as shown in Figure 1. These ASs' contain nodes wherein each node is connected with another node. The nodes are randomly deployed and are connected using an Ethernet connection. All the nodes inside an AS, which are connected to a subscriber can cache a content. Additionally, these nodes also have CIT table besides a forwarding information table (FIB) and pending interest table (PIT). The CIT of a node will store all the information regarding contents cached on other nodes inside an AS. Nodes in an AS will share their CIT information with each other. An Autonomous system is connected to the NCC through a gateway node. This node is same as all the nodes in the AS, but its only difference is that it connects to the NCC. Furthermore, this node can also cache a content if any subscriber connected to it makes a request for a content.

D. CONTENT INFORMATION TABLE (CIT)

Every cache node inside AS has a CIT table and this table stores information of location of the content which is shared with nodes in the same AS. The Information are: content name, name of the node and the port number through which a particular node is accessible as shown in Table 2. Moreover, the CIT table works on the principles of the link-state routing [27], which means that a node will share information if there is any change performed. For instance, a new content is cached or a content is evicted. The interest is directed towards NCC if a content is not present in AS.

TABLE 2. Content Information Table.

Type	Node name	Port#
temperature	R1	3
humidity	R3	1
traffic	R5	5

E. THE CACHING SCHEME

The location for caching is the node connected to the subscriber, or a node is connected with a particular subscriber for making of interest of content. Any node in AS can cache a content if a subscriber make a interest for a content. let $A_1 \ldots\ldots, A_k$ be a number of AS. For every AS A_j can be expressed below.

$$A_j = \bigcup_{i=1}^{l_j} C_{j,i} \tag{1}$$

Eqn. 1 shows that AS is the union of all the caches $C_{j,i}$ inside AS. Where every cache $C_{j,i}$ provides contents to $m_{j,i}$ subscribers. We denote a subscriber with $S^P_{j,i}$ where $j \in \{1, 2, \ldots .l_j\}$, $i \in \{1, 2, \ldots .k\}$, and $P \in \{1, 2, \ldots .m_{j,i}\}$ With the given Information. We can define three different conditions of caching content as expressed below.

$$d(S^P_{j,i}, C_{j,i}) = \begin{cases} 1 & \text{if } j_1 = j_2, i_1 = i_2, \\ 2 & \text{if } j_1 = j_2, i_1 \neq i_2 \\ 3 & \text{if } j_1 \neq j_2 \end{cases} \quad (2)$$

This Eqn. 2 has 3 different conditions for caching the content. In the first condition, the content is assumed to be in the local cache memory. In the second condition, the content is present inside AS but not in the local memory. The content is not present in the network, is the third condition. Hence, the interest made will not be satisfied by any node in AS, and will approach to go out towards NCC. The mentioned conditions has presented in the proposed algorithm (Algorithm 1) for the caching scheme.

Algorithm 1 Proposed Caching Scheme

INPUT: Interest r^c arrives at cache from subscriber
OUTPUT: Cache Content
DATA:
P is the Content
C_s is the cache node connected to subscriber
r^c is the Interest for the content
BEGIN:
if $P \in C_s$ **then**
    return P
else if $P \notin C_s$ **then**
    Search content in CIT
else if $P \exists CIT$ **then**
    return P
else if $P \notin C_s \wedge P \nexists CIT$ **then**
    $NCC \leftarrow r^c$
else if $P \exists NCT$ **then**
    return P
else if $P \nexists NCT$ **then**
    transmit r^c to the Publisher
    Return P
else if $P == 1 \wedge P \leqslant C_s$ **then**
    Cache content P
else if $P > C_s$ **then**
    Use LFU replacement scheme
    Cache content P

F. CONTENT UPDATE

The proposed CCC gives a method for updating a cached content. NCC maintains a timer for each content which is cached on network. When this timer reaches to zero for any content, an update message U_m is sent towards AS containing a copy of the particular content. This message only asks for the nodes willingness to get the updated copy of content. When U_m reaches the gateway node of AS, it is then forwarded to the nodes with the outdated content. These nodes are identified using CIT table of the gateway node. Once the nodes receives U_m, these nodes look for the number of unsatisfied interests for the content. The nodes then replies to the message showing the willingness of nodes to get the updated version of content, if the number of interest are greater or equal to 1 in any cache. If these messages are more than one in any AS then only one message will be forwarded to NCC by the gateway node. In case a cache node does not have any interests for that content, it will simple delete the content. The information regarding number of interests is available with the Pending Interest table (PIT). The content downloaded will have the same. The whole process has depicted in the proposed algorithm 2. Now, to formalize the mentioned working method, let's a timer t_i in NCT for content P in any AS A_{js} where $P \in A_j$ and $A_{js} \subseteq$ of A_j. If a timer $t_i = 0$, it sends the updated message to A_{js}, where $A_{js} \subseteq$ of A_j. As we know from Eqn. 1 that AS is a union of different cache nodes $C_{j,i}$. As mentioned earlier that update message is sent to subset of A_j, which is A_{js} in this case, as expressed below.

Algorithm 2 Update Method for Cached Content

INPUT: Timer T expires for content P in NCT
OUTPUT: Updated Content
DATA:
U_m = update message
A_{js} = Subset of Autonomous Systems
C = Cache node
C_{Ps} = cache node with Content P but also a subset of node C
P is the content
ml_i = Memory Location of Cache nodes
r^c is the Interest
BEGIN:
$send(U_m, A_{js})$
if $A_{js} \leftarrow U_m$ **then**
    $send(U_m, C_{Ps})$
else if $C_{Ps} \leftarrow U_m$ **then**
    Look for Content in Memory Location ml_i
else if $P = ml_i$ **then**
    Check for Pending Interests $r^c forP$ in PTI
else if $r^c \geq 1$ **then**
    Return $P + 1$
else if $r^c = 0$ **then**
    Delete P

$$A_{js} = \bigcup_{i=1}^{l_j} C_{j,i} \quad (3)$$

When the update message reaches to A_{js}, it is then transmitted towards a subset of nodes having a copy of content P, such as $C_js,_{is} \subseteq C_{j,i}$. When the message is received by the cache node, then it looks the content in memory location r_u, such as $C_{js,is,ru} = P$, where $u \in \{1, 2, 3 \ldots .q\}$. If $access(P) \geq 1$ is true for any r_u, as expressed below.

$$C_{js,is,ru} = P + 1 \quad (4)$$

Eqn. 4 shows the content updation. However, the content will be removed if the condition becomes false such as If

$access(P) = 0$ is true for any r_u then

$$C_{js,is,ru} = 0 \tag{5}$$

G. ENERGY EFFICIENCY

As mentioned earlier, the proposed scheme reduces the energy requirements of the IoT devices. This is done by introducing new tables in the proposed scheme. The first table is NCT table added to the central controller or NCC. This table is used to keep track of the contents cached on the network. If a user makes a request for a content which is cached on another AS, NCC will use the information in NCT table and retrieve the content from that AS. Hence preventing the Interest from reaching the IoT node. Furthermore, if the same content becomes stale on multiple ASs' and an update is required, so NCC will access the IoT device for the updated content once, and will transmit the updated copy to all the ASs' where the old and outdated copy exists. In CCC, IoT device will only be accessed when there is an Interest for a content which is not available on any AS or if an outdated copy needs to be updated. This low access rate makes the CCC more energy efficient.

V. PERFORMANCE EVALUATION

This section presents the simulations performed for evaluating the performance of the proposed CCC scheme in the realistic environment. It has been categorized into three folds. Firstly, the simulation setting are discussed. Secondly, the performance evaluation metrics are defined. Thirdly, the comparative analysis of simulation results of the proposed scheme compared with LCE [10], LCC [21], and IoTCache [30] is presented.

A. SIMULATION SETTING

The proposed scheme is simulated in ndnSim-based NS3 simulator [41]. The computer machine is core *i7* with microprocessor of 3 GHz processor, RAM is 8GB, and the operating system is the Ubuntu 18.04. These are the basic requirements to run a simulator. Moreover, the simulation coverage area is 1000 x 1000 meters for network. The number of nodes employed are 1050 in this simulation. Among these nodes, one node is elected for NCC, 50 nodes are IoT content producers, 100 nodes are cache nodes which are further equally divided among ASs' with 10 nodes each, and 900 subscriber are divided among the cache nodes at 9 subscribers each. These subscribers generate contents using a Zipf distribution. The value range of α is in 0.2 to 1.1. The size of cache is in between 500 MB to 1 GB. The size of the content is 1 chunk and is kept constant throughout the simulation. Simulation has run for 10 times and the average result of these 1o times have used in the final result.

B. SIMULATION METRICS

There are three simulation performance metrics used to carry out the results from the simulation. The following are the detailed explanation of performance metrics.

1) ENERGY CONSUMPTION

Energy consumption is total energy consumed by the IoT devices. This scheme uses the energy consumption with respect to two different cache sizes to assess the caching scheme in terms of total energy consumption of the IoT devices. An energy efficient caching scheme will conserve energy on IoT devices, by reducing the number of accesses. The total energy consumption can be calculated as

$$EC_T = \sum_{i=1}^{n_i} m_j(L(e_{bt} + T_e c)) \tag{6}$$

where n is the total number of IoT devices, L is the content size, e_b is the energy consumption for transmitting 1 bit of content to the NCC, $T_e c$ is the total energy used to generate one complete content, and m_j is the number of times an IoT device is activated.

2) RESPONSE LATENCY

Response latency is the amount of time consumed from the Interest made till delivery of the content. This research gives an average response latency with respect to two cache sizes. A good caching decision will always reduce the response latency of the network. If the response latency is high it will effect the performance of the network. The Response latency can be calculated as

$$RL_{avg} = \frac{\frac{\sum_{i=1}^{RL_i}}{|r^c|}}{|sub|} \tag{7}$$

where RL_i is the response latency of a single content, $|r^c|$ is the total number of interests generated, and $|sub|$ is the total number of subscriber who have made interests for contents. This will give us the average response latency of this network.

3) CACHE-HIT

Cache-hit is the number of interests satisfied by the cache nodes in the network. We can say that higher the cache-hit more energy efficient the IoT devices will be.This metric shows the performance of the caching decision policy. In this paper, we calculate the average cache-hit as expressed below.

$$CH_{avg} = \frac{\sum_{i=1}^{|N|} t_i - \sum_{i=1}^{|N|} m_i}{|r^c|} \tag{8}$$

where t_i is the total content provided by the caches, m_i shows the total number of content provided by the IoT devices and r^c shows the total number of interests.

C. ANALYSIS OF RESULTS

The energy consumption at cache size 500 MB is depicted in Figure 2.The average energy consumption values with at cache size 500 MB with respect to different values of α are given in Table 3. The average values show that the difference of energy consumption is small of the proposed scheme CCC compared to LCC, LCE, and IoTCache. When the cache size is increased to 1 GB, we see noticeable decrease in

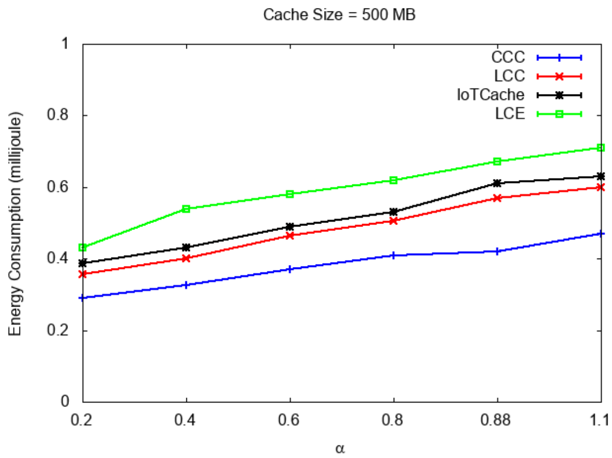

FIGURE 2. Energy Consumption at Cache size 500 MB.

TABLE 3. Energy Consumption at cache size 500 MB and 1 GB.

α	500 MB			
	CCC	LCC	LCE	IoTCache
0.2	0.290	0.355	0.430	0.388
0.4	0.327	0.400	0.540	0.430
0.6	0.370	0.465	0.580	0.490
0.8	0.410	0.505	0.620	0.530
0.88	0.420	0.570	0.670	0.610
1.1	0.470	0.599	0.710	0.630
α	1 GB			
	CCC	LCC	LCE	IoTCache
0.2	0.203	0.316	0.450	0.356
0.4	0.215	0.387	0.513	0.437
0.6	0.255	0.413	0.553	0.458
0.8	0.315	0.492	0.610	0.529
0.88	0.355	0.499	0.678	0.551
1.1	0.380	0.497	0.715	0.545

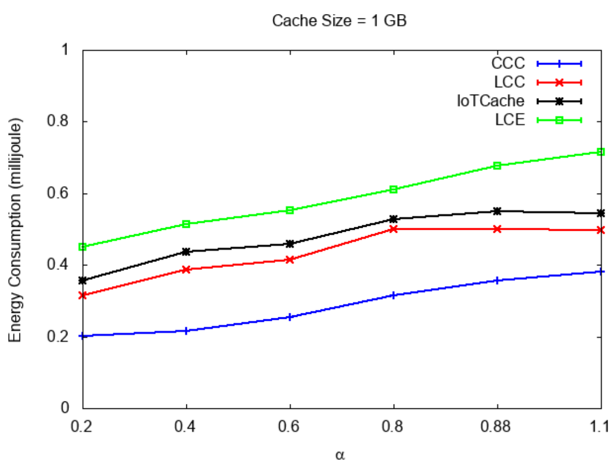

FIGURE 3. Energy Consumption at Cache size 1GB.

the energy consumption of CCC as compared to LCC,LCE and IoTCache as shown in Figure 3. The average energy consumption values with respect to α at cache size 1 GB is shown in Table 3.

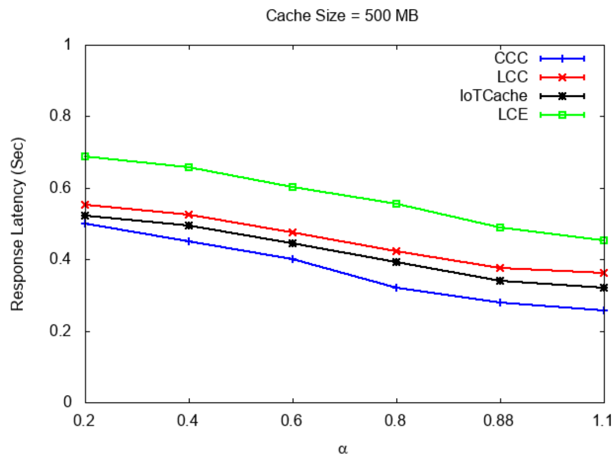

FIGURE 4. Response Latency at Cache size 500 MB.

Similarly, the response latency of the proposed scheme is better than LCC, LCE, and IoTCache. At cache size 500 MB the response latency of CCC is near to that of IoTCache, but significantly better than LCC and LCE as shown in Figure 4. The average response latency values are given in Table 4. At cache size 1 GB, initially the difference between CCC and IoTCache is small, such as in Table 4 at α value of 0.2 and 0.4 the average latency of CCC and IoTCache are 0.490, 0.435 and 0.510, 0.490 respectively. As the network grows and value of α increases the response latency difference between CCC and IoTCache becomes significant as shown in Figure 5.

TABLE 4. Response Latency at cache size 500 MB and 1 GB.

α	500 MB			
	CCC	LCC	LCE	IoTCache
0.2	0.501	0.553	0.689	0.523
0.4	0.450	0.525	0.657	0.495
0.6	0.400	0.474	0.601	0.444
0.8	0.320	0.423	0.555	0.393
0.88	0.280	0.375	0.490	0.341
1.1	0.258	0.361	0.454	0.321
α	1 GB			
	CCC	LCC	LCE	IoTCache
0.2	0.490	0.555	0.602	0.510
0.4	0.435	0.535	0.593	0.490
0.6	0.380	0.495	0.578	0.468
0.8	0.280	0.425	0.533	0.400
0.88	0.180	0.375	0.492	0.335
1.1	0.140	0.352	0.465	0.300

Furthermore, CCC also beat LCC, LCE and IoTCache in achieving better Cache-hit with both cache size as shown in Figure 6 and 7. looking at the average cache hit values in Table 5, it can be observed that cache hit with both cache sizes, the difference in cache hit is very minimal at α values 0.2 and 0.4. At $\alpha = 0.6$ the difference of cache hit becomes noticeably greater than LCC, LCE, and IoTCache, Which means longer the network is in operation, more the cache hit of CCC. Increase in cache hit also means that decreased number of interests will reach the IoT publisher, making them

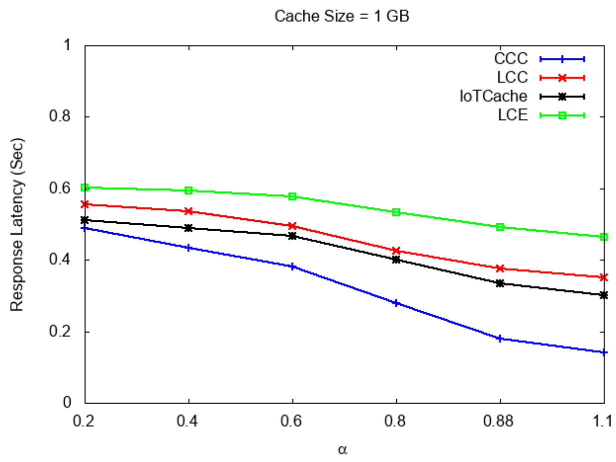

FIGURE 5. Response Latency at Cache size 1GB.

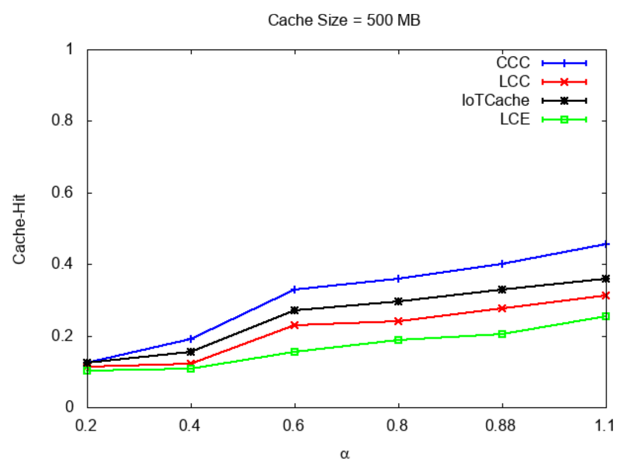

FIGURE 6. Cache-Hit at Cache size 500 MB.

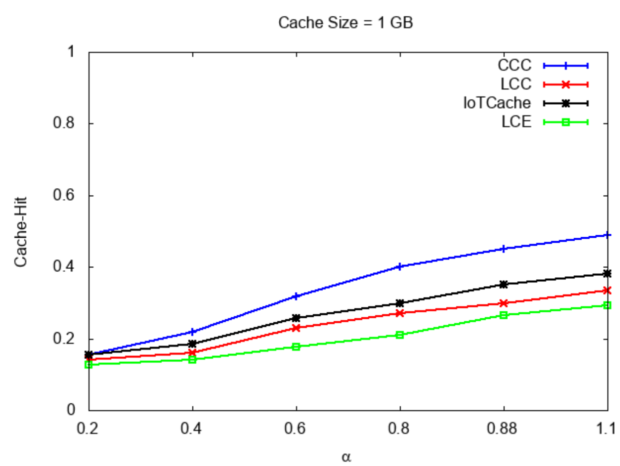

FIGURE 7. Cache-Hit at Cache size 1GB.

more energy efficient. From the result it can be concluded that when the size of cache is increased then the proposed CCC scheme gives better results as compared to a smaller cache size.

TABLE 5. Cache-hit at cache size 500 MB and 1 GB.

α	500 MB			
	CCC	LCC	LCE	IoTCache
0.2	0.124	0.114	0.102	0.124
0.4	0.190	0.122	0.109	0.155
0.6	0.330	0.229	0.155	0.270
0.8	0.360	0.240	0.189	0.295
0.88	0.400	0.275	0.205	0.330
1.1	0.455	0.311	0.255	0.360
α	1 GB			
	CCC	LCC	LCE	IoTCache
0.2	0.155	0.140	0.128	0.155
0.4	0.219	0.161	0.142	0.185
0.6	0.319	0.228	0.178	0.258
0.8	0.400	0.272	0.210	0.299
0.88	0.450	0.299	0.265	0.350
1.1	0.490	0.335	0.294	0.380

VI. CONCLUSION

This paper proposes a CCC scheme for IoT Environment. The proposed scheme is an energy efficient caching scheme which reduces the energy requirements of the IoT devices. For this purpose a central controlling entity is used to track the transmission of contents between ASs' of a network and extracts the content from AS rather than forwarding the interest towards IoT devices. For tracking of content an NCC table is used which keeps the record of all the contents cached on different ASs'. Moreover, every node inside as AS also has CIT table, which keeps record of all the contents inside an AS. The CIT table will help find a cached content inside an AS. Furthermore, a method for updating the transient IoT contents is also proposed. This method updates the content when the timer of the content reaches zero. The simulation results show that proposed CCC scheme performs remarkably better than the existing caching schemes in terms of energy Consumption, response latency, and cache-hit.

In the future work, we have intended to extend the proposed scheme to the environment where the IoT devices are mobile. Because mobility is one of the key feature of IoT devices.

REFERENCES

[1] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, "Cache-aware computation offloading in IoT systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 61–72, Mar. 2020.

[2] M. I. A. Zahed, I. Ahmad, D. Habibi, and Q. V. Phung, "Green and secure computation offloading for cache-enabled IoT networks," *IEEE Access*, vol. 8, pp. 63840–63855, 2020.

[3] C. V. N. Index. (2017). *Cisco Visual Networking Index*. Accessed: May 3, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[4] Z. Zhang, H. Ma, and L. Liu, "Cache-aware named-data forwarding in Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[5] M. A. Naeem, M. A. U. Rehman, R. Ullah, and B.-S. Kim, "A comparative performance analysis of popularity-based caching strategies in named data networking," *IEEE Access*, vol. 8, pp. 50057–50077, 2020.

[6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol.*, 2009, pp. 1–12.

[7] I. Din, B.-S. Kim, S. Hassan, M. Guizani, M. Atiquzzaman, and J. Rodrigues, "Information-centric network-based vehicular communications: Overview and research opportunities," *Sensors*, vol. 18, no. 11, p. 3957, Nov. 2018.

[8] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 320–351, 1st Quart., 2020.
[9] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of Internet-of-Things data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 3185–3190.
[10] N. Laoutaiis, S. Syntila, and L. Stavrakakis, "Meta algorithms for hierarchical Web caches," in *Proc. IEEE Int. Conf. Perform., Comput., Commun.*, Apr. 2004, pp. 445–452.
[11] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks," in *Proc. Int. Conf. Res. Netw.* Berlin, Germany: Springer-Verlag, 2012, pp. 27–40. [Online]. Available: https://dl.acm.org/doi/proceedings/10.5555/2342042
[12] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable ICN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 147–158, 2013.
[13] A. Dabirmoghaddam, M. M. Barijough, and J. Garcia-Luna-Aceves, "Understanding optimal caching and opportunistic caching 'at the edge' of information-centric networks," in *Proc. 1st ACM Conf. Inf.-Centric Netw.*, 2014, pp. 47–56.
[14] M. Zhang, B. Hao, R. Wang, and Y. Wang, "A pre-caching strategy based on the content relevance of smart device's request in information-centric IoT," *IEEE Access*, vol. 8, pp. 75761–75771, 2020.
[15] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet Things J.*, early access, Apr. 9, 2020, doi: 10.1109/JIOT.2020.2986803.
[16] I. U. Din, S. Hassan, A. Almogren, F. Ayub, and M. Guizani, "PUC: Packet update caching for energy efficient IoT-based information-centric networking," *Future Gener. Comput. Syst.*, vol. 111, pp. 634–643, Oct. 2020.
[17] S. Zhang and J. Liu, "Optimal probabilistic caching in heterogeneous IoT networks," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3404–3414, Apr. 2020.
[18] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless Internet of Things," in *Proc. Int. Conf. Recent Adv. Internet Things (RIoT)*, Apr. 2015, pp. 1–6.
[19] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 482–487.
[20] O. Hahm, E. Baccelli, T. C. Schmidt, M. Wahlisch, and C. Adjih, "A named data network approach to energy efficiency in IoT," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.
[21] Z. Zhang, C.-H. Lung, I. Lambadaris, and M. St-Hilaire, "IoT data lifetime-based cooperative caching scheme for ICN-IoT networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.
[22] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2014.
[23] O. Waltari and J. Kangasharju, "Content-centric networking in the Internet of Things," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2016, pp. 73–78.
[24] K. B. S. Chenna and C. K. Srinivasan, "Survey on optimization in IoT," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 1924–1928.
[25] J. François, T. Cholez, and T. Engel, "CCN traffic optimization for IoT," in *Proc. 4th Int. Conf. Netw. Future (NoF)*, Oct. 2013, pp. 1–5.
[26] D. Liu, *Cisco Router and Switch Forensics: Investigating and Analyzing Malicious Network Activity*. Rockland, MA, USA: Syngress, 2009.
[27] M. N. Lakhmir, A. A. Qureshi, M. I. Channa, F. A. Jokhio, and S. Nizamani, "Performance evaluation of link state routing protocol in an enterprise network," *Bahria Univ. J. Inf. Commun. Technol.*, vol. 8, no. 1, p. 88, 2015.
[28] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.
[29] G. Girban and M. Popa, "A glance on WSN lifetime and relevant factors for energy consumption," in *Proc. Int. Joint Conf. Comput. Cybern. Tech. Informat.*, May 2010, pp. 523–528.
[30] B. Chen, L. Liu, M. Sun, and H. Ma, "IoTCache: Toward data-driven network caching for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10064–10076, Dec. 2019.
[31] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, Aug. 2009.
[32] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1443–1474, 2nd Quart., 2018.
[33] C. A. Kerrche, F. Ahmad, M. Elhoseny, A. Adnane, Z. Ahmad, and B. Nour, "Internet of vehicles over named data networking: Current status and future challenges," in *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*. Cham, Switzerland: Springer, 2020, pp. 83–99. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-22773-9#about
[34] I. U. Din, H. Asmat, and M. Guizani, "A review of information centric network-based Internet of Things: Communication architectures, design issues, and research opportunities," *Multimedia Tools Appl.*, vol. 78, no. 21, pp. 30241–30256, Nov. 2019.
[35] I. U. Din, M. Guizani, S. Hassan, B.-S. Kim, M. K, Khan, M. Atiquzzaman, and S. H. Ahmed, "The Internet of Things: A review of enabled technologies and future challenges," *IEEE Access*, vol. 7, pp. 7606–7640, 2019.
[36] S. Vural, N. Wang, P. Navaratnam, and R. Tafazolli, "Caching transient data in Internet content routers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1048–1061, Apr. 2017.
[37] M. Ha and D. Kim, "On-demand cache placement protocol for content delivery sensor networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 207–216.
[38] R. Shrimali, H. Shah, and R. Chauhan, "Proposed caching scheme for optimizing trade-off between freshness and energy consumption in name data networking based IoT," *Adv. Internet Things*, vol. 7, no. 2, pp. 11–24, 2017.
[39] X. Sun and N. Ansari, "Dynamic resource caching in the IoT application layer for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 606–613, Apr. 2018.
[40] T. Miwa and S. Kimura, "Cooperative update mechanism of cache update method based on content update dynamic queries for named data networking," in *Proc. 7th Int. Symp. Comput. Netw. Workshops (CANDARW)*, Nov. 2019, pp. 33–39.
[41] ndnSIM. (2017). *ndnSIM for NS3*. Accessed: Jan. 10, 2020. [Online]. Available: http://ndnsim.net/2.2/

HAMID ASMAT is currently pursuing the Ph.D. degree in computer Science from Sarhad University Peshawar, Pakistan. He is also working as a Lecturer with the University of Haripur. He has published many research articles in reputed journals and conferences. His research interests include ICN, the IoT, smart cities, caching.

FASEE ULLAH graduated from the Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia, in 2017. He worked as an Assistant Professor with the Department of Computer Science and IT, Sarhad University of Science and IT-Pakistan. He is currently working as a Postdoctoral Fellowship with the University of Macau. He has published many research articles in reputed impact factor journals and conferences. His research areas include wireless body area networks, wireless sensor networks, cloud security, smart cities, big data analytics, and the Internet of Things. He was a recipient of the Chancellor Award and the Best Student Award at UTM during his Ph.D., for his excellent contribution to wireless communication and health monitoring. He is also providing reviewing services to the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, the IEEE SENSOR JOURNAL, ACM, and the *International Journal of Distributed Sensor Networks*.

MAHDI ZAREEI (Member, IEEE) received the M.Sc. degree in computer network from the University of Science, Malaysia, in 2011, and the Ph.D. degree from the Communication Systems and Networks Research Group, Malaysia-Japan International Institute of Technology, University of Technology, Malaysia, in 2016. In 2017, he joined the School of Engineering and Sciences, Tecnologico de Monterrey, as a Postdoctoral Fellow, where he has been started working as a Research Professor, since 2019. His researches mainly focus on wireless sensor and ad hoc networks, energy harvesting sensors, information security, and machine learning. He is a member of the Mexican National Researchers System (level I). He is currently serving as an Associate Editor for IEEE ACCESS and *Ad Hoc & Sensor Wireless Networks Journal*.

ATIF KHAN received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 2004, and the Ph.D. degree in computer science (text mining) from Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia, in 2016. Since 2016, he has been working as an Assistant Professor with Islamia College Peshawar, Pakistan. He is currently a Technical Committee Member of many international conferences and a Reviewer of many international conferences, journals. His current areas of research interest include data mining, text mining, sentiment analysis and opinion mining, recommender systems, and machine learning. He was a recipient of the Best Student Award and the Pro-Chancellor Award at UTM during his Ph.D., for his excellent contribution in the field of text mining.

EHAB MAHMOUD MOHAMED (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from South Valley University, Egypt, in 2001 and 2006, respectively, and the Ph.D. degree in information science and electrical engineering from Kyushu University, Japan, in 2012. From 2013 to 2016, he has joined Osaka University, Japan, as a Specially Appointed Researcher. Since 2017, he has been an Associate Professor with Aswan University, Egypt. He has also been an Associate Professor with Prince Sattam bin Abdulaziz University, Saudi Arabi, since 2019. He is currently the General Chair of the IEEE ITEMS'16 and the IEEE ISWC' 18. He is also a Technical Committee Member of many international conferences and a Reviewer of many international conferences, journals, and transactions. His current research interests are 5G, B5G and 6G networks, cognitive radio networks, millimeter-wave transmissions, Li-Fi technology, MIMO systems, and underwater communication.