

Article

Dynamic and Energy Efficient Cache Scheduling Framework for IoMT over ICN

Abdullah Alourani ¹, Muhammad Sardaraz ², Muhammad Tahir ^{2,*} and Muhammad Saud Khan ³

- ¹ Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University, P.O. Box 6640, Buraidah 51452, Saudi Arabia; ab.alourani@qu.edu.sa
- ² Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
- ³ Department of Computer Science, Air University Aerospace and Aviation Campus, Kamra 43570, Pakistan
- * Correspondence: m_tahir@cuiatk.edu.pk

Abstract: The Internet of Medical Things (IoMT) is the network of medical devices, hardware infrastructure, and software applications used to connect the healthcare information technology. Massive traffic growth and user expectations cause challenges in the current exhausting models of IoMT data. To reduce the IoMT traffic, Information Centric Network (ICN) is a suitable technique. ICN uses persistent naming multicast communication that reduces the response time. ICN in IoMT provides a promising feature to reduce the overhead due to the distribution of commonly accessed contents. Some parameters such as energy consumption, communication cost, etc., influence the performance of sensors in the IoMT network. Excessive and unbalanced energy consumption degrades the network performance and lifetime. This article presents a framework called Dynamic Cache Scheme (DCS) that implements energy-efficient cache scheduling in IoMT over ICN to reduce network traffic. The proposed framework establishes a balance between the multi-hop traffic and data item freshness. The technique improves the freshness of data; thus, updated data are provided to the end-users via the effective utilization of caching in IoMT. The proposed framework is tested on important parameters, i.e., cache-hit-ratio, stretch, and content retrieval latency. The results obtained are compared with the state-of-the-art models. Results' analysis shows that the proposed framework outperforms the compared models in terms of cache-hit-ratio, stretch, and content retrieval latency by 59.42%, 32.66%, and 18.8%, respectively. In the future, it is intended to explore the applicability of DCS in more scenarios and optimize further.

Keywords: ICN; cache scheduling; wireless network; energy efficiency; IoT; big data; algorithm



Citation: Alourani, A.; Sardaraz, M.; Tahir, M.; Khan, M.S. Dynamic and Energy Efficient Cache Scheduling Framework for IoMT over ICN. *Appl. Sci.* **2023**, *13*, 11840. <https://doi.org/10.3390/app132111840>

Academic Editors: Stefano Silvestri and Francesco Gargiulo

Received: 12 October 2023

Revised: 25 October 2023

Accepted: 27 October 2023

Published: 29 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) represents a network comprising interconnected devices, namely actuators and sensors, designed to measure various environmental variables and execute actions as per predefined directives [1]. Currently, there are billions of interconnected IoT devices generating an extensive volume of data, which has a significant impact on conventional Internet traffic patterns. The characteristics of IoT depict that anyone can be connected to the network at any time through any path from anywhere [2]. The IoT sector is presently experiencing remarkable growth and is capturing the attention of researchers due to its rapid expansion into diverse domains, including smart retail, smart agriculture, smart homes, smart health, and many others [3]. IoT devices collect and process data to understand the environment and make efficient and accurate decisions to improve daily life activities in different aspects [4]. IoT devices are considered resource-constrained in terms of memory, computing, and battery power [5].

Information Centric Network (ICN) is an effective technique for IoT networks and provides independent location content names in-network caching, which gives ICN valuable contribution to data dissemination [6]. ICN not only reduces the load on data producers but

also overcomes delays via the concept of unique location [7]. ICN provides persistent naming multicast communication which reduces the response time and provides the concept of cache to minimize the network traffic [8]. Massive traffic growth and user expectations cause challenges in the current exhausting scheme of IoT data networks [9]. Millions of IoT devices are connected over the Internet which poses challenges for researchers. Sometimes many IoT devices request the same data item concurrently. It is necessary to minimize redundancy over the network and fetch fresh data items [10].

The number of IoT devices has increased significantly and a huge increase is expected in the future. IoT needs certain parameters for better performance and efficient resource utilization over the Internet [11]. One of the primary requirements of IoT is addressing the content over the Internet via unique content name rather than by IP address [12]. The content consumers and devices search for the content name instead of the IP address. If data are cached between the content producer and the content consumer, then instead of retrieving data items from the content producer (source node), data might be available in intermediate nodes. Hence, overall load on the source nodes is minimized and the content consumer can retrieve the data item directly from the caching node instead of the content producer. In this way, content consumers can obtain the data without activating the source node. If the IoT data item is cached properly between the intermediate nodes, the IoT network can achieve the advantage in terms of energy efficiency and source node load, and consumers will obtain fresh data quickly [13]. Connected devices often retrieve identical content, such as health reports, while running various applications. This redundancy contributes to network congestion and affects data freshness.

This article presents a framework for energy-efficient cache scheduling in the IoMT network. The proposed framework uses a multi-hop traffic load and freshness of the data items to achieve the desired goals. The key finding of this research is the Integration of IoMT with ICN, which enhances energy-efficient cache scheduling, balanced data traffic load, and data freshness. The proposed framework addresses the challenges associated with growing IoMT traffic and outperforms existing models in key performance aspects. The results obtained are compared with Tag-Based Caching Strategy (TCS) and Client-cache (CC) strategy. Results' analysis shows that the proposed framework outperforms the other methods on selected parameters. This research has the potential to significantly impact the efficiency and sustainability of IoMT in the healthcare sector and offers a promising avenue for future advancements and applications by a further exploration of DCS's applicability in diverse scenarios and its ongoing optimization. The rest of the paper is organized as follows. Section 2 presents the related work followed by materials and methods in Section 3. Section 4 presents an experimental evaluation with detailed results and discussion. Finally, Section 5 concludes the article.

2. Related Work

IoMT comprises intelligent devices, including wearables and medical monitors that are utilized in healthcare monitoring spanning personal use, homes, communities, clinics, and hospitals. These devices enable real-time location tracking, telehealth services, and various other functionalities. IoMT facilitates secure wireless communication among remote devices via the Internet, enabling swift and adaptable medical data analysis. Its impact on the healthcare industry is multifaceted, with significant benefits observed when deploying IoMT in various contexts, whether in a home setting, on an individual's body, within communities, or even within hospital facilities to obtain the most current data [14,15]. IoMT consumers consistently seek the latest published data due to its frequent updates, where IoT data are cached away from the network edge [16]. IoT devices are small and the data collected from billions of devices is so large that it can disrupt regular network traffic. Therefore, ICN can be used to enhance the network scalability of IoMT [17]. The integration of ICN with IoT is more suitable as it results in reliable data transmission and consumes less bandwidth as compared to IP-based network communication [18]. ICN plays a vital role in ensuring content availability over the network and facilitating rapid

content access [2]. Several techniques have been proposed to address cache scheduling including the implementation of caching mechanisms with different objectives, such as data size, data lifetime, sensing cache, time-based cache, and collaborative caching. Some models utilize cloud computing to store medical data in cloud storage, where additional operations like scheduling and resource utilization prediction are performed [19–22].

Tag-based Caching Strategy (TCS) uses tag filters for the matching and lookup of the requested contents for dissemination to the target node [23]. In TCS, all network nodes are linked with specific lists of tags to identify highly requested content. Tag filters are generated from the tag list through a hash function to enhance content distribution. As the network node receives a request from a user, the corresponding tag filter decides whether to transmit the content cache to the intermediate location or not. In this caching strategy, the tags linked with the required contents are checked by the tag filters inside the CS, and all the tags are hashed and mapped in a counter to find the most requested content. If a tag counter for specific content crosses the threshold, then the content is considered popular. As a result, all the nodes check the tags to identify the content location and decide whether the content that needs to be cached reaches the preferred location or not. TCS appears to be a promising approach for improving content distribution efficiency in networks. However, its complexity and resource requirements, as well as the need for careful parameter tuning, are potential drawbacks that need to be carefully considered. Another article [24] presents the Client Cache (CC) strategy in which the cached contents inside the network nodes are observed and considered valid. The on-path caching approach is selected to cache transmitted contents which eliminates the requirements to inform the client which node is appropriate to cache the preferred content. Moreover, the number of nodes is reduced to be aware of the most essential nodes in terms of content quality and cache size. The main objective of CC is to extend the content validity, which is examined as the requested content is found inside the cache. The content material is considered valid if the lifetime within the publisher is higher than the lifetime of the version within the cache. While selecting the requested content, a validity test is performed to check the content inside the content publisher whenever a content material request arrives at the neighborhood-caching node. They show that the combination of the two proposed schemes results in a notable improvement in content validity at the expense of a certain degradation in both server-hit and hop reduction ratios. However, it requires a more extensive and in-depth analysis of the trade-offs and challenges associated with the proposed solutions. Authors in [25] presented a secure and energy-efficient framework using IoMT for E-healthcare (SEF-IoMT) and explored the growing popularity of IoT in the healthcare sector. The proposed framework explored the need for an improved framework to address issues related to energy consumption, communication costs, and data security. The proposed framework reduces energy consumption and communication overhead in IoMT. Comparative experimental results show the effectiveness of the proposed method in comparison to existing methods. However, it lacks a transparent evaluation of metrics and an in-depth analysis of its findings to validate its scientific contribution.

Another article [26] presents a secure and energy-efficient IoT model for e-health, focusing on the secure transmission and retrieval of biomedical images over IoT networks. The authors utilize compressive sensing and a five-dimensional hyper-chaotic map (FDHC) for image encryption, addressing the challenge of hyper-parameter tuning in FDHC. The encryption technique is image-sensitive, depending on the initial scrambled row and column for permutation and diffusion operations. Experimental results indicate that the proposed method outperforms existing image encryption techniques. Consequently, it is considered suitable for securing communication in energy-efficient IoT networks. However, a clear discussion of the hyper-parameter tuning problem, sensitivity to input images, and the practicality of the proposed method is missing. The study presented in [27] addresses a crucial issue in the field of healthcare IoT by emphasizing the need for data security and energy efficiency. This study addresses the critical challenges of storing sensitive medical data securely and preventing cyber threats in the rapidly growing IoMT network.

The integration of homomorphic secret sharing and artificial intelligence to enhance the maintainability of disease diagnosis systems and reinforce secure communication is highly important. However, the analysis of the results shows the proposed model suffers from an increased packet drop ratio and imbalanced energy consumption in the presence of high network load among IoT nodes. Age-based cooperative caching is presented in [28] for the efficient utilization of cache and routing in ICN. This framework is designed to minimize heavy computation and communication between routers. The algorithm dynamically updates the freshness of data in routers by pushing popular data over the network edges, while less popular data reside away from the edges. The method can be considered efficient for ICN, but these schemes work with special-purpose applications, i.e., P2P systems that mostly benefit web applications like web caching and distributed file systems. However, the evolving nature of ICN and how this approach might adapt to changing network conditions and requirements has not been considered for the evaluation and validation of the proposed model.

In another article, the authors [29] present a model that aims to balance between multi-hop communication costs and the freshness of transient data items. IoT devices create extra traffic load, and it is necessary to make caching decisions to minimize the traffic. The short lifetime of IoT data items leads to extra complexity in handling caching decisions. The cost function is used for data items generated at the source node to address the issue. The results analysis shows that this model effectively describes the effect of data transiency and accurately represents the benefits of a caching system, particularly in terms of reducing network load, and especially for substantially requested data items. However, practical implementation and scalability considerations, as well as addressing security and privacy concerns, are essential for the successful deployment of in-network caching solutions in IoT ecosystems, which has been not considered in this study. The article [30] discusses the role of TCP/IP in the Internet of Vehicles (IoV) and its limitations, such as weak scalability, low efficiency in dense environments, and unreliable addressing in high mobility situations. It highlights the potential of NDN technology in addressing these issues through content caching. It proposes a data caching scheme that considers the spatial-temporal characteristics of different message categories in VNDN, including emergency safety, traffic efficiency, and service messages. Experimental results show that this scheme outperforms existing data caching protocols, improving the average hit rate, hop count, and cache replacement times by approximately 50%. However, unreliable addressing in high mobility circumstances, especially considering vehicular networks, is challenging and involves vehicles moving at high speeds, which can lead to frequent changes in network topology. Moreover, Content-centric networking can raise concerns about data privacy and security. An analytical model is presented in [31] to balance the trade-off between multi-hop communication costs and data item freshness. Transient data caching decisions are used to ensure data freshness. This model is known as a pull-based caching scheme, as the scheme considers the rate of incoming requests from the content consumers and data item lifetime to cache data items. However, the paper does not delve into the technical challenges and complexities of implementing caching mechanisms for transient data. Practical issues related to cache management, cache replacement policies, and the scalability of such systems are not sufficiently discussed. Push-based caching is proposed in [32] to minimize the traffic load and explores the potential of named data networking (NDN) as a framework for facilitating IoT applications, particularly those requiring push-based communication. In the push-based caching scheme, data servers push the data items towards consumers with static networks. This scheme splits the IoT data traffic into four main parts, i.e., periodic data, event triggered data, command base, and data query supported by the NDN structure. However, reliability is still the issue in push models and, for instance, more comprehensive evaluation of the proposed model in practical IoT use cases is required, and lacks the consideration of potential challenges and drawbacks.

A caching scheme for the ICN-IoT data network is presented in [33]. The authors proposed a novel caching scheme based on the IoT data lifetime and user request rate, leveraging information ICN principles. This approach aims to reduce the energy consumption of IoT devices by intelligently caching data at various network nodes. The approach considers the incoming request rate of users to efficiently utilize the energy of IoT devices. The main idea is to keep the source node in sleep mode to save energy for the network. With the help of a cooperative caching scheme, the method also minimizes the traffic load and provides fresh data to the users. Data items are cached at intermediate nodes, i.e., between the content producers and content consumers, which leads to energy saving. The paper provides a valuable contribution to the field of IoT by addressing the crucial issue of energy efficiency through the innovative use of caching based on data lifetime and request rate. However, addressing scalability and security considerations is essential for the practical implementation of the proposed scheme in IoT networks. The paper [34] introduces the concept of a sensing cache, strategically positioned at a wireless gateway of IoT sensors to minimize energy consumption. A dynamic threshold adaptation algorithm allows the sensing cache to adjust its parameters in real time, maximizing the combined hit rate of the sensing service from multiple sensors. The sensor devices harvest energy from the environment to save energy for sending or receiving information. However, sensing errors may occur that can create problems with data freshness. The smart caching scheme for the IoT platform presented in [35] creates and maintains the balance between the freshness of data and energy of nodes. Freshness and energy consumption are the main parameters considered. When IoT data are combined over the Internet, issues like freshness loss, energy consumption, and increase in traffic load are raised. The freshness of data items and energy are considered parameters of the cache to provide fresh data to users while the energy of the sensors remains balanced. Data redundancy is also minimized in the smart cache scheme by dropping the data item from the cache nodes that are no longer needed. This leads to data freshness and less energy consumption. However, there are no details provided about the experiment, the methodology, or the results and it lacks critical analysis and empirical evidence. Freshness Aware Reverse Proxy (FARP) caching scheme is presented in [36]. The scheme is based on data item freshness and access performance. The caching scheme brings the advantage of access performance but when the data in the cache is not fresh and expires at the source node, the method creates additional burden. The cost function is used to balance the switching between used and unused data items present at the cache nodes. The method also adjusts the two parameters for the cache node to provide fresh data to the user. However, when the data expires, it may overlap with the latest published data at the source node. A probabilistic caching scheme for IoT networks is presented in [37]. This model considers the freshness of IoT data, energy, and storage as measurement parameters to optimize the retrieval of data. ICN content is fetched through unique name and location tags to ensure the freshness of data, data retrieval, ease of transmission cost, and sharing of data. In-network caching also helps to optimize data retrieval and manage network load. The proposed caching scheme also considers resource constraints, i.e., energy and space to provide fresh data to users. However, it does not provide a direct comparison with the state-of-the-art models. A coherent caching freshness scheme for IoT is proposed in [38]. The method increases the speed of data retrieval while minimizing the network load. The aim is to increase the cache hit rate and reduce network delays. The authors worked on on-path caching scheme to improve the energy of caching nodes. In this caching technique, the intermediate nodes not only resolve the problem of energy but also resolve the problems of data freshness and redundancy. The method uses the cost function for the energy-efficient utilization of caching nodes. However, several critical aspects have not been considered in this study including scalability, security, and interoperability.

All the cache frameworks discussed above have their advantages. Millions of IoT devices are connected to the Internet which causes huge traffic load and poses several challenges. The need for the efficient utilization of available resources to benefit from the

IoMT network is still challenging. The performance and lifetime of the sensors in IoMT are concerned with energy consumption, computation, and other parameters. The solutions designed for this purpose should consider the relation between related parameters to ensure that improving one parameter does not affect other parameters. Therefore, more research and robust cache frameworks are needed to achieve the maximum advantages of IoMT over ICN networks.

3. Materials and Methods

This section provides an overview of the proposed framework. Section 3.1 introduces the system model employed for experimental purposes. Section 3.2 elaborates on the Cache Routing and Replacement Policy utilized for the evaluation and analysis of this research study, and, finally, Section 3.3 outlines the operational aspects of the proposed framework.

3.1. System Model

The network system model comprises of N routers, i.e., set r_n is the number of each router, where $n = 1, 2, 3, \dots, N$ as shown in Figure 1. The r router is the nearest to the content consumer, while router r_N is the nearest to the content generator. The term c is used to represent the size of the cache on the router r_k . The Poisson distribution is used to follow the average number of requests of content service with the request rate l for class k using the Zipf–Mandelbrot distribution with type factor λ_k for class k . Network contents are partitioned into three different classes, i.e., class a , b , and c , based on characteristics as discussed in [39]. The content classification depends on the jitter, delay, content utilization, and the pattern of request rate. Class a requires low jitter and delay integrated with repeatedly requested contents. Class b is user-generated content with less jitter and a medium required delay. This class includes the content that has been generated and distributed by volunteer subscribers, like sites, web-based social media content, etc. Class c comprises video-on-demand content which is not the target of this research study. This research follows Poisson distribution to make several requests for content service classes. Due to content popularity, the Zipf–Mandelbrot distribution is used for request modeling, which comprises a flattened factor [40–42]. The content on top rankings has more chance of being requested as compared with lower rankings content.

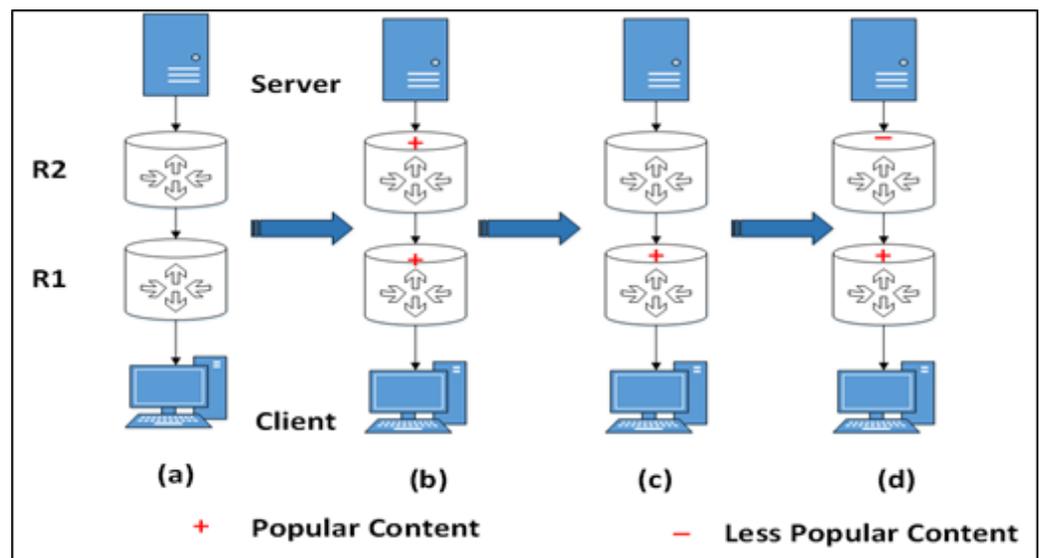


Figure 1. Workflow of popular content over the network. (a) Has no caching content (b) Both routers have the same content but the content of $R1$ is older than the content of $R2$. (c) After some time, the content of $R2$ expires. (d) Client needs less popular content, which is cached at $R2$. The age of the content is determined by the distance from the server or source node.

The cache receives request rate β , which is contingent upon the condition $\beta \geq 0$, denoting the request intensity. Let $P(i)$ be the probability of content requests; N is the total number of requests, q is the vector, and α is the exponential factor for content popularity. In this context, we can express the probability of accessing content i using Equation (1).

$$P(i) = \frac{(i + q)^{-\alpha}}{\sum_{i=1}^N (i + q)^{-\alpha}} \quad (1)$$

Each content may receive updates to replace the previous data item. It is assumed that the content n receives updates as per the Poisson distribution request model bounded with the refresh rate of $\lambda_n \geq 0$. As all the content of the data is subject to updates, then the same content item that resides in the local cache can be considered as an older version of the content. To measure the freshness of the content, the age Δ_n with respect to time t can be bounded as $\Delta_n(t) \in \{0, 1, 2, \dots, n\}$, where n represents the number of updates. The cost of the fetching content is subject to the condition, i.e., $c_f \geq 0$, in case of operations. For performance, the cost metric can be integrated with time as $c_a = \Delta_n(t)$ subject to $c_a \geq 0$, and linear growth rate.

3.2. Cache Routing and Replacement Policy

The caching technique characterizes the replication technique used to spread duplicates of the content in the cache. Replication techniques have two measurements. Content-based replication, which depends on attributes of the content and settles on caching choices, e.g., pick popular content to store and ignore others. Node-based replication depends on the qualities of the topology, e.g., picking more important nodes for storing the content. Both content-based and node-based replication are resource optimization approaches. Both techniques attempt to designate storage space to attain maximum advantage, e.g., increased cache-hit-ratio. The replication techniques explored in the context of ICN are all node-based since the content-based technique can work on a large Internet scale. There exists several caching approaches, i.e., Leave Copy Everywhere (LCE), Leave Copy Down (LCD), Bernoulli random caching, random choice caching, Probabilistic Caching (ProbCache), centrality-based caching, and hash-routing [40–43]. In the IoT over ICN network, caching nodes can be configured using one of the state-of-the-art cache eviction policies, i.e., Least Recently Used (LRU), Least Frequently Used (LFU), First in First Out (FIFO), and random. This research utilizes the LRU cache eviction policy because it is the most widely used cache replacement strategy. At an instance when new content is required to be pushed into the cache, it eliminates the least recently asserted content request. This replacement strategy is proficient for line speedup activities because both search and substitution replacement can be performed in consistent time, i.e., $O(1)$. Nonetheless, its efficiency degrades under the Independent Reference Model (IRM) supposition because of the likelihood that the current requested content does not rely upon previous content requests.

3.3. Proposed Framework

This article presents a Dynamic Caching Scheme (DCS) to improve the freshness of the cache contents and network performance. The proposed technique neither needs extensive computations nor tags, filters, metadata, and additional network communication that create extra overhead in IoT networks. Moreover, if the source node and user are at longer distances, it may extend the retrieval delay because of the smaller size of the cache nodes. In the IoMT network, each content has a lifetime and is expected to expire in some instance of time. The proposed framework is designed to dynamically push popular content towards edge nodes over the network by changing the data ages. Data ages are used to handle the lifetime of content over the network while removing the contents that are no longer needed. Each content in the cache has an age that decides the lifetime of the cached content. Popular content over the network has a longer age. First, at the router node, the content age is estimated in correlation with the freshness of the data item. If the age of the data item is not matched with the source node, it is removed from the cache. In the proposed

framework, an Interest Record Table (IRT) integrated with cache control is added at the router to improve the working of CS while avoiding traffic load at the source node. Figure 1 illustrates the workflow of popular content across the network. This network consists of a client and server, each connected to their respective routers, *R1* and *R2*. The server is responsible for generating two types of content: popular and less popular. Notably, both routers can store at least one content item at a time. In deciding whether to cache data or not, especially for transient small-sized data, a “period” field is introduced within the packet. This field supplies updated data directly from the source node. To accomplish this, the system incorporates the content’s age with the data item, with the first router node establishing the initial age baseline. The quantitative findings indicate that when data items are not cached at the nodes, this incurs a higher cost in the IoT network. Caching policies are typically implemented in intermediate nodes. Each node possesses specific cache threshold requirements and can dynamically adjust these thresholds based on the request rate. To minimize retrieval delays, edge nodes employ smaller caching thresholds, allowing them to cache more data items. In contrast, the root node has a larger caching threshold, enabling it to cache data items for a longer duration. The system carefully considers both data lifetime and request rates to reduce energy consumption, data retrieval delays, and the overall network traffic load. When the request rate reaches the specified threshold, it is recommended to cache the requested data in the edge node.

Figure 2 illustrates the architecture of the proposed model, showing the frequently requested content cached at the edge nodes, i.e., *R6* and *R9*. When the cache capacity of an edge node reduces, and the validity of the content persists, it is stored in the cache of an intermediate node. This strategy prevents excessive network load by accommodating incoming requests at intermediate nodes, avoiding the need to route them back to the source node. The Pending Interest Table (PIT) is involved in tracking unhandled interest packets and recording relevant data names, along with their incoming and outgoing edges. The Forwarding Information Base (FIB) aids in network bridging and routing by determining the appropriate output network edge controller. Content replacement across the network follows an LRU cache policy, primarily targeting content flows. Retrieving content directly from the source node can lead to delays in terms of cache hit ratios for IoT networks. The proposed framework utilizes diverse data types to optimize the bandwidth utilization and alleviate network congestion. When users *S1* and *S2* send interest packets for content *C1*, which is a popularly cached item at the edge node, the concept of content popularity becomes evident. If another request arises for content *C3* to be cached at the edge node, but the cache is full, *C3* is directed to an intermediate caching node. This approach minimizes content replication within the network, ensuring efficient memory utilization and a reduction in bandwidth consumption. The proposed framework enhances three key caching parameters: cache hit ratio, network stretch, and latency. Edge nodes copy content from the publisher, with content lifetime specified upon publishing data. IoT data adheres to specific lifetimes, and require termination at some point in the network. Consequently, the implementation of DCS enhances network performance and reduces the network traffic load on the source node. Algorithm 1 is used to deploy the routing policy for checking the age of the content at the cache of the edge node. If the content is expired and removed from the cache, then new content is added to the cache, and age is defined for this content. Upon receiving the age of the caching node, a message is sent to the router for an update, i.e., if it is replaced by the replica, then add a new age to the incoming contents until it expires and is removed from the cache node. If space is not available at the cache of the edge node, then the content is passed to the higher threshold value towards the central position. The workflow of the popular content over the network shows that it is periodic, as when users request come for content that is popular over the network, CS is first checked. If the requested content is available locally, then the same is provided. Otherwise, the procedure checks the cached content at CS, a call to IRT is performed, and the content is provided to the user. If the content is not available, then it is shifted to PIT to check whether it is the same content requested by any other node. In

case of the same content, the request with respect is removed. Another entry is made with respect to the current request, and data are provided to the desired request. In contrast, the FIB broadcasts the request for popular content.

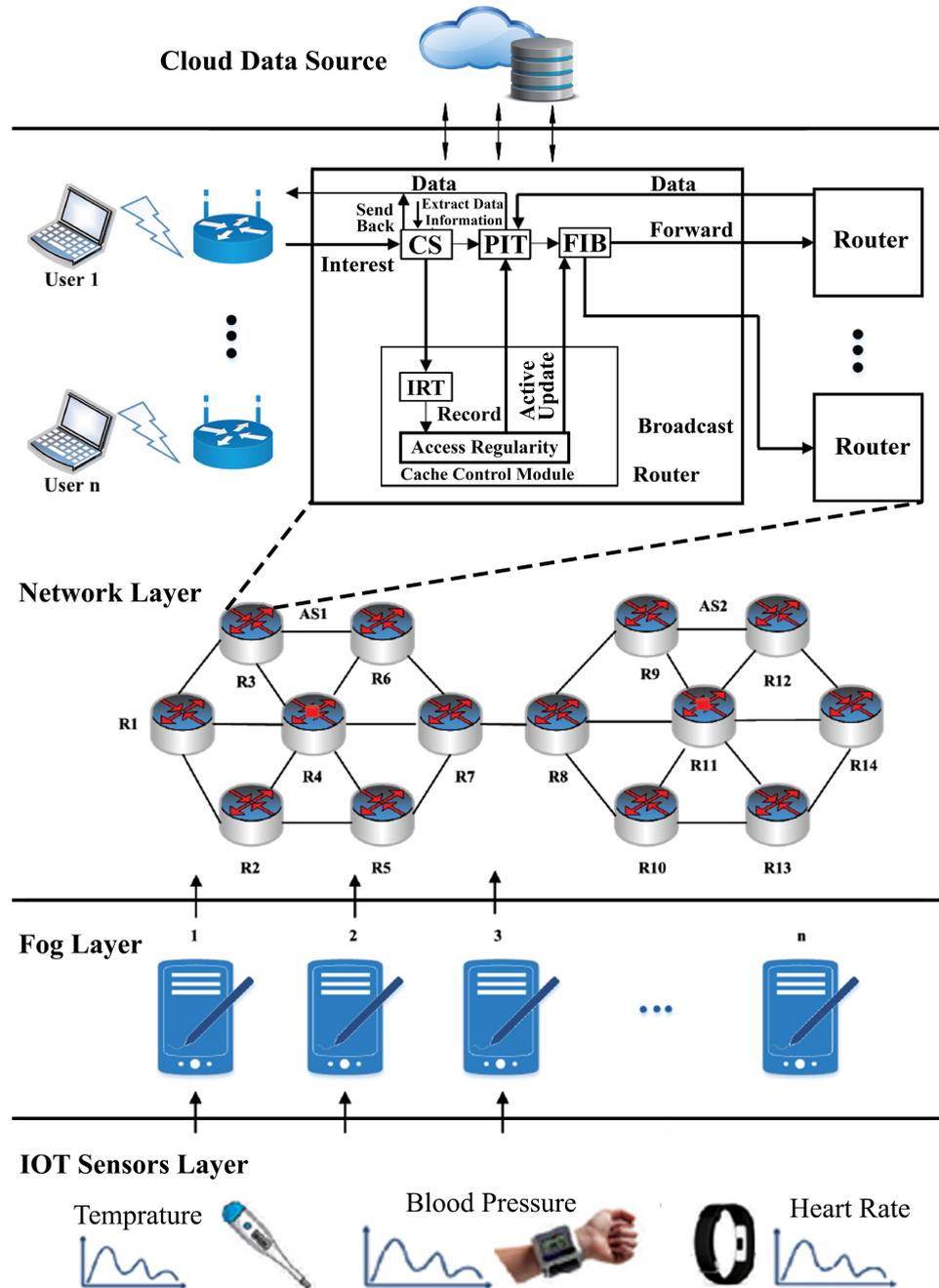


Figure 2. The architecture of the proposed system. The architecture presents the workflow of interest packets sent from biomedical sensor devices toward the cloud data source. Content producers and content consumers have caching nodes that cache the popular data for short lifetime. If the ordinary request comes over the network, it is forwarded to CS to check; if found, then it is sent to the consumer, or otherwise forwarded to PIT to check for any other entry in PIT. If it exists, then we discard the existing entry and add updated entry against this request; if data are not present, it is further sent to FIB to broadcast.

Algorithm 1: Receiving age of the IoMT data items at the caching nodes.

```

Input : Interest data item arrives at cache
Output: Age of cache content
1 Remove expired content from the cache of node.
2 data_item arrives at intermediate node i
3 if (data_item is in cache) then
4   if (freshness_of_data_item < freshness_of_cached_data_item) then
5     refresh (cache)
6     add to cache_edge_node (data_item)
7     data_item_age = new_age (data_item)
8     forward_the_data_item to the next_hop's_node
9   else
10    data_item_age = new_age (data_item)
11    forward_the_data_item to the next_hop's_node
12  end
13 else if (node_cache has more space) then
14   cache (data_item)
15   new_age(data_item)
16   forward_the_data_item to the next_hop's_node
17 else if (lifetime_data_item ≥ caching threshold) then
18   while (node_cache_has_no_space) do
19     call LRU replacement policy
20   end
21   cache (data_item)
22   forward_the_data_item to the next_hop's_node
23 else
24   forward_the_data_item to the next_hop's_node
25 end

```

4. Experimental Evaluation

This section presents an experimental evaluation of the proposed framework. First, the experimental setup is presented, including details of the parameters, the dataset used, the simulation environment, and the tools used for simulation. The experimental setup is followed by Section 4.2 that covers the detailed results with relevant discussion.

4.1. Experimental Setup

To assess the performance of the proposed model, a distributed network of virtual IoT devices is configured using open-source Icarus interlinked with the web-based Bevywise IoT simulator [44,45]. The WUSTL-EHMS-2020 dataset is used for simulation [46]. This dataset comprises 44 attributes, with 35 of them being network flow metrics, eight representing patients' biometric features, and one dedicated to the label. The configuration of essential parameters is presented in Table 1. Data freshness can be defined as the time elapsed between the creation of the IoT object and the retrieval of this object from the cache store [47]. The data age is the age of the data item, which is the time between the arrival at the router and the generation at the source node, while the period field is used to keep count of the age [48]. Three network topologies, namely Tree, Abilene, and GEANT, have been chosen for the comparative evaluation. To ensure a fair comparison, all algorithms were executed within the same environment. To demonstrate the validity and accuracy of

the results, each experiment was repeated 100 times, and the average value of each parameter was employed for a comparative analysis. The parameter used to measure network performance utilization is the cache hit ratio CH_{ratio} , calculated with Equation (2) [49].

$$CH_{ratio} = \frac{\sum_{k=1}^k Hit_k}{\sum_{k=1}^k (Hit + Miss)} \quad (2)$$

The content retrieval latency $Content_{Rlatency}$ parameter is used to monitor the traffic load over the network, i.e., the time taken by the interest packet to obtain content from the source node and send it back to the user. Let $|R|$ be the total number of requests sent by the user; then, the cache retrieval latency can be computed, as shown in Equation (3) [49].

$$Content_{Rlatency} = \frac{\sum_{i=1}^{|R|} latency R_i}{|R|} \quad (3)$$

The stretch shows the measurement of the distance from the content used to the content publisher. The stretch can be computed as shown in Equation (4) [49].

$$Stretch = \frac{\sum_{i=1}^R Hop - Traveled}{\sum_{i=1}^R Total - Hop'} \quad (4)$$

The nominator in Equation (4) shows the number of hops between the content user and the content publisher in terms of the cache hit occurrence. The denominator shows the total number of hops, i.e., hops between the content user and the source.

Memory consumption shows the measurement of the transmitted content that can be cached while downloading the data path for a time interval. Consumers can download the content from multiple routers. In ICN, memory consumption means the capacity that shows the volume utilized by interest and data contents, as shown in Equation (5).

$$memory_{consumption} = \frac{U_m}{T_m} \times 100 \quad (5)$$

where U_m shows the memory utilized by the cached content and T_m shows the cache storage (total memory) of the router and the data delivery path.

Table 1. Experimental setup simulation parameters.

Parameter Description	Parameter Value
Simulation time	48 h
Cache Size (elements)	500, 1000
Content size	10 MB
Catalog Size (elements)	10^6
Content Categories	File
Topology	Tree, Abilene, & GEANT
Zipf distribution (content popularity)	0.8 & 1.2
Replacement Policy	LRU

4.2. Results and Discussion

The proposed framework significantly improves network performance while reducing the traffic load directed at the source node. In contrast to DCS, the TCS employs tags that require separate computations and fail to accommodate popular content when it regains its popularity. Although the central caching node (CC) excels due to its strategic location, the limited number of such nodes results in inadequate content availability, characterized by a diminished cache-hit ratio, increased network stretch, and extended retrieval latency.

The proposed framework effectively leverages caching nodes to enhance the cache-hit ratio and minimize the network stretch. In the case of DCS, each node is equipped with

a table that calculates interests to identify the most frequently requested data across the network, taking into account data names, access frequency, and a specified threshold value. This threshold value helps us to recognize the most frequently requested content that needs to be cached at the edge node, considering the cache's service to multiple devices. The results from these devices are consolidated by a central caching node.

The main challenge lies in selecting an optimal threshold to maximize the cache-hit rate. The threshold adaptation algorithm outlined in [37] is utilized to enable the caching node to learn and implement an optimal threshold strategy. Figures 3–5 depict graphical comparisons of cache-hit ratios, latency, and stretch for CC, TCS, and the proposed framework across various network topologies, including Tree, Abilene, and GEANT. The proposed model is rigorously tested and validated under different cache size and content popularity configurations, specifically, 500 and 0.8, 1000 and 0.8, 500 and 1.2, 1000 and 1.2. In DCS, the selection of optimal caching nodes, along with the essential role played by edge nodes in maintaining data freshness, works to minimize the distance between the source node and the end nodes. Figure 3 shows the results of the cache hit ratio for different numbers of cache sizes and popularity. Results' analysis of Figure 3a shows that the proposed framework gained 105.5%, 109%, and 21.42% improvement in terms of cache-hit ratio over CC executed on Tree, Abilene, and GEANT network topology, respectively, for cache size of 500 and popularity of 0.8. The improvement of DCS over TCS with the same parameters and configuration is 72%, 76%, and 11.47%, executed on Tree, Abilene, and GEANT network topologies, respectively. Figure 3b shows that the proposed framework achieved 132.4%, 122.8%, and 55.3% improvement in terms of the cache-hit ratio over CC executed on Tree, Abilene, and GEANT network topology, respectively, on the cache size of 1000 and popularity of 0.8. The improvement of the proposed model over TCS with the same parameters and configuration is 59.2%, 65.9%, and 17.7% in the case of Tree, Abilene, and GEANT network topologies, respectively.

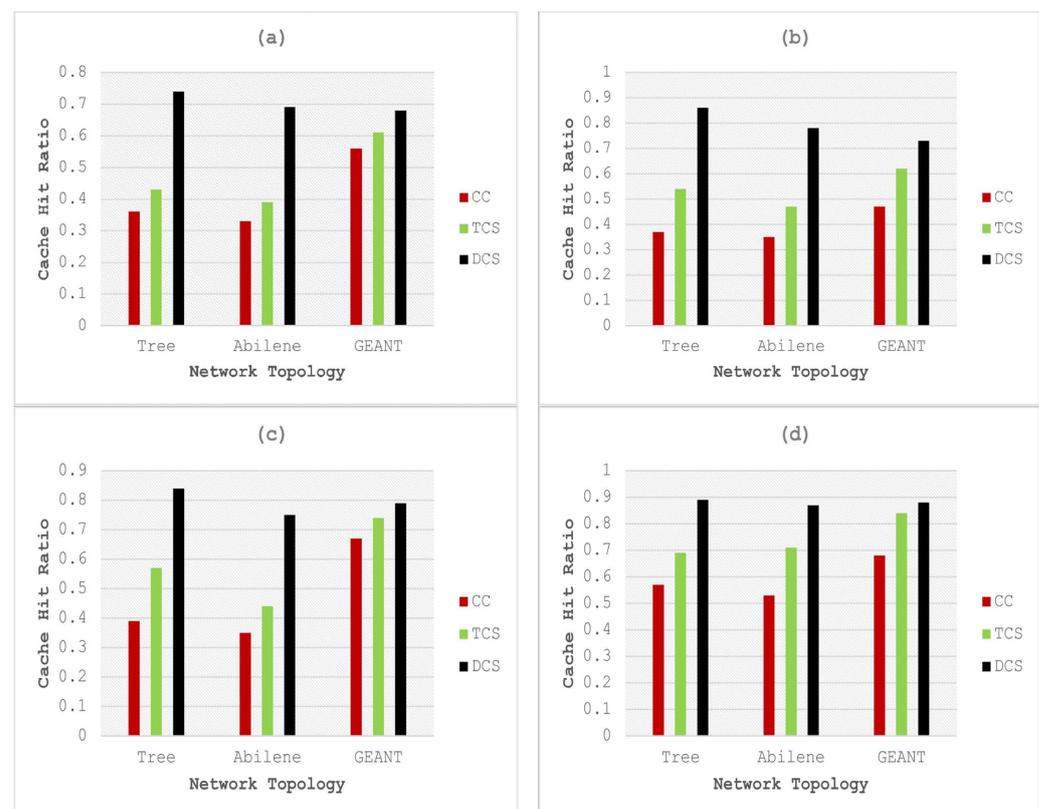


Figure 3. Comparative analysis of cache-hit ratio with cache sizes and popularity of (a) 500 and 0.8, (b) 1000 and 0.8, (c) 500 and 1.2, and (d) 1000 and 1.2.

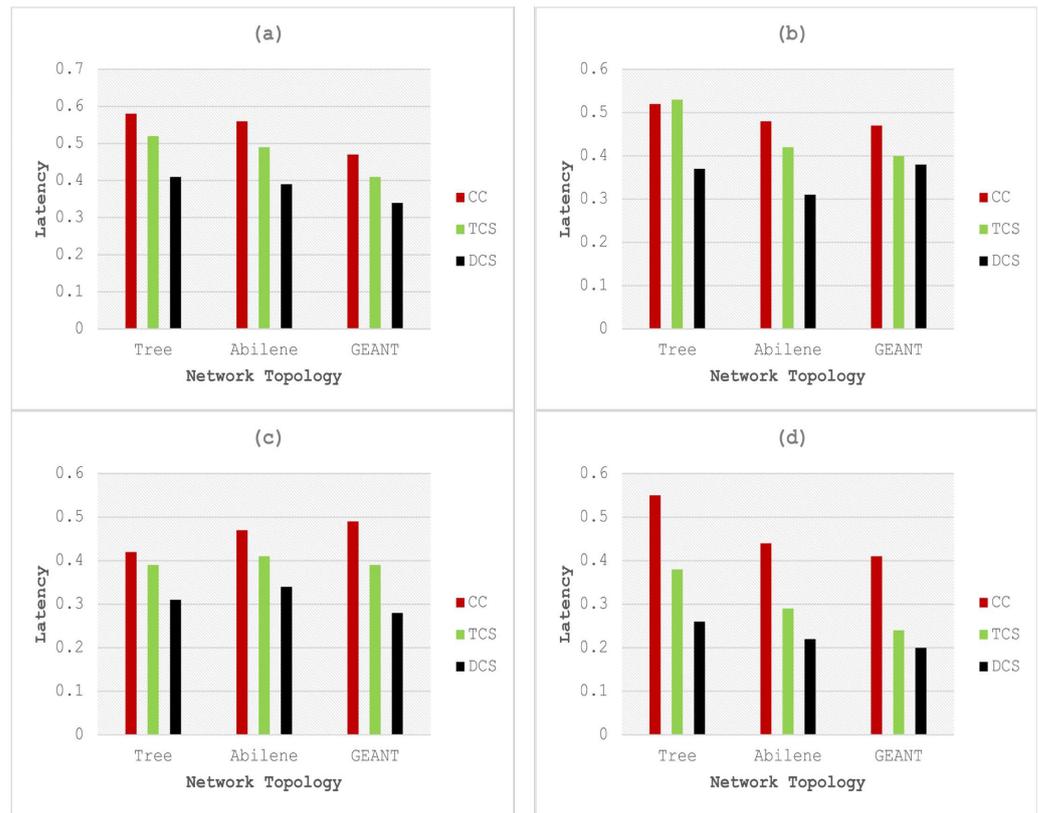


Figure 4. Comparative analysis of latency with cache sizes and popularity of (a) 500 and 0.8, (b) 1000 and 0.8, (c) 500 and 1.2, and (d) 1000 and 1.2.

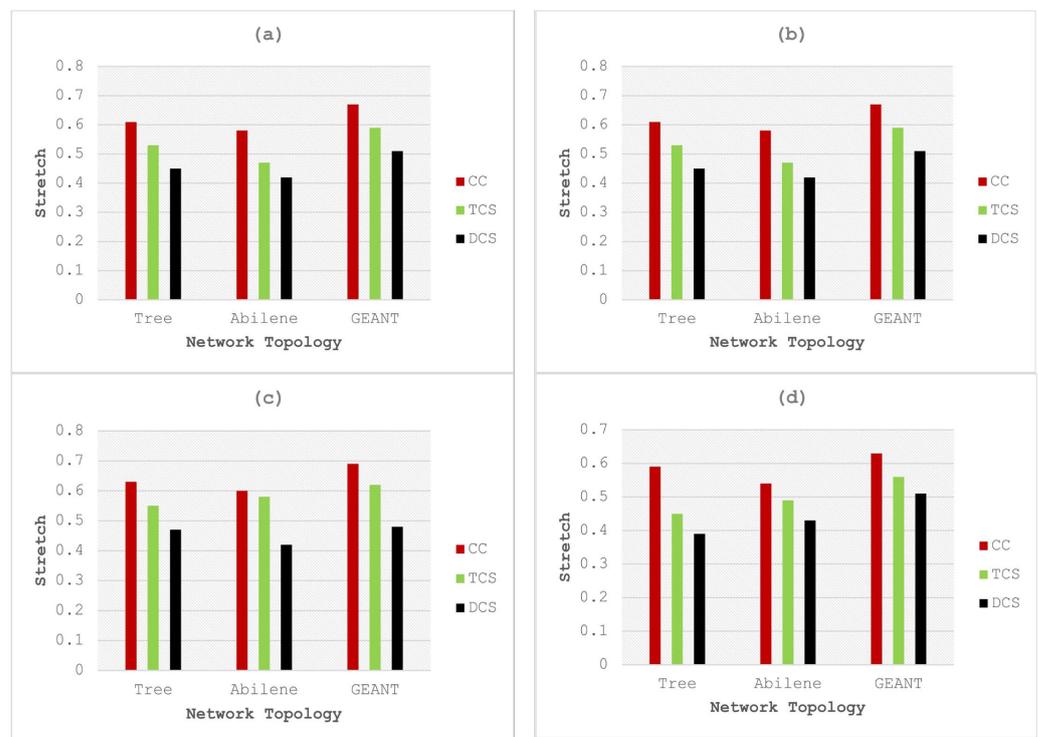


Figure 5. Comparative analysis of stretch with cache sizes and popularity of (a) 500 and 0.8, (b) 1000 and 0.8, (c) 500 and 1.2, and (d) 1000 and 1.2.

Figure 3c shows the comparative results of the cache-hit ratio of DCS with CC and TCS on the cache size of 500 and content popularity of 1.2. The results analysis shows that

the proposed model is 115.3%, 114.2%, and 17.9% better than CC executed on Tree, Abilene, and GEANT network topology, respectively. The improvement of the proposed model over TCS with the same parameters and configuration is 47.3%, 70.4%, and 6.75% tested on Tree, Abilene, and GEANT network topologies, respectively. Figure 3d presents the comparison of the cache-hit ratio of DCS with CC and TCS for a cache size of 1000 and popularity of 1.2. The results analysis shows 56.14%, 64.15%, and 29.4% improvement over CC in the case of Tree, Abilene, and GEANT network topology, respectively. The improvement of the proposed model over TCS with the same parameters and configuration is 28.98%, 22.5%, and 4.76% executed on Tree, Abilene, and GEANT network topologies, respectively.

Figure 4 shows the comparative results in terms of the latency of DCS with CC and TCS for different cache sizes and content popularity. Figure 4a shows the results of models with 500 elements for cache size, and the content popularity was set to 0.8. Three different network topologies, i.e., Tree, Abilene, and GEANT were used in experiments for testing and validation. The results analysis shows that the proposed model consumes less time, i.e., a 29.31%, 30.35%, and 27.65% decrease in cache latency as compared to CC on Tree, Abilene, and GEANT topologies, respectively. Meanwhile, DCS is 21.15%, 20.4%, and 17.07% more efficient than TCS executed with the same configuration of parameters and topologies, respectively.

Figure 4b shows the comparative results of DCS with CC, and TCS in terms of cache latency. For these experiments, 1000 elements of cache size and a 0.8 value of content popularity are considered and executed on Tree, Abilene, and GEANT network topologies. The results analysis shows that the proposed model is 28.8%, 35.41%, and 19.1% more efficient than CC in the case of Tree, Abilene, and GEANT topologies, respectively. DCS is 30.1%, 26.19%, and 5.0% efficient in cache latency, as compared to TCS executed on the same configuration of parameters and topologies. Figure 4c shows comparative results of DCS with CC, and TCS in terms of cache latency for the cache size of 500 and content popularity of 1.2. An analysis of the results shows that the proposed model is 26.1%, 27.6%, and 42.8% more efficient than CC in the case of Tree, Abilene, and GEANT topologies, respectively. DCS is 20.5%, 17.0%, and 28.2% efficient in cache latency as compared to TCS executed on the same configuration of parameters and topologies. Figure 4d shows the comparative results of DCS with CC, and TCS in terms of cache latency. For these experiments, 1000 elements of cache size and a 1.2 value of content popularity are considered and executed on Tree, Abilene, and GEANT network topologies. The analysis shows that the cache latency of the proposed model is 52.7%, 50.0%, and 51.2% better than CC and executed in Tree, Abilene, and GEANT topologies, respectively. DCS is 30.5%, 24.1%, and 16.6% more efficient than TCS executed on the same configuration of parameters and topologies. CC showed a higher latency rate because of the long distance between the source and the central position of the caching nodes. TCS increases the amount of similar content over the network with the help of tags and filters near the source location, due to which it does not select the central node for the caching; thus, latency increases, as shown in the results. The proposed model retrieves a copy of the popular data item first in the edge node, then increases the number of interest and sends it to the central position, which decreases the distance that leads to a decrease in the latency.

Figure 5 depicts the comparative results of DCS with CC, as well as TCS in terms of stretch. Figure 5a shows the results of 500 elements of cache size and a 0.8 content popularity. The analysis shows that the stretch of the proposed model is 26.2%, 27.5%, and 23.8% better than CC executed on Tree, Abilene, and GEANT topologies, respectively. DCS is 15.0%, 10.6%, and 13.5% more efficient than TCS executed on the same configuration of parameters and network topologies.

Figure 5b shows the comparative results of DCS with CC, as well as TCS in terms of stretch. For experiments, 1000 elements of cache size and a 0.8 value of content popularity are considered and executed on Tree, Abilene, and GEANT network topologies. Results analysis shows that the stretch of DCS is 26.22%, 27.5%, and 23.8% more efficient than CC executed on Tree, Abilene, and GEANT topology, respectively. The proposed model

achieved a percent improvement gain of 15.09%, 10.6%, and 13.5% as compared to TCS executed on the cache size of 1000 and content popularity rate of 0.8 on different network topologies. Figure 5c shows the comparative results of DCS with CC, as well as TCS in terms of stretch for the cache size of 500 with content popularity of 1.2 on Tree, Abilene, and GEANT network topologies. The analysis shows that the proposed model achieved a percent improvement gain of 25.3%, 30.0%, and 30.4% as compared to CC executed on Tree, Abilene, and GEANT topology, respectively. DCS is 14.5%, 27.5%, and 22.5% more efficient than TCS when executed on the same configuration of parameters and network topologies. Figure 5d shows the comparative results of DCS with CC, as well as TCS in terms of stretch. For these experiments, 1000 elements of cache size and a 1.2 value of content popularity are considered and executed on Tree, Abilene, and GEANT network topologies. The analysis shows that the stretch of the proposed model DCS is 33.9%, 20.3%, and 19.0% more efficient than CC executed on Tree, Abilene, and GEANT topology, respectively. DCS is 13.3%, 12.2%, and 8.9% more efficient than TCS when executed on the same configuration of parameters and network topologies. DCS moves the popular content near the user while the central position caching node is facilitated to provide the data item earlier than the source node because of the smaller stretch. Results clearly show that when the data item is placed near the user, it results in a decrease of the stretch.

5. Conclusions

IoMT is a significant and promising technology and provides an ease of access to real-world health data instantly. Currently, the number of IoMT devices is increasing exponentially, which intensifies the requirements. In the IoMT environment, the study of ICN caching policies is involved in terms of content placement strategies. IoMT contents are distributed in terms of scalability and cost-effectiveness. With the rapid growth of the IoMT network traffic, it is preemptory to be a suitable framework to address the challenges. This article introduces a dynamic caching strategy designed for the IoMT. By integrating IoMT with ICN, we propose a dynamic caching scheme aimed at reducing energy consumption within information-centric IoMT networks. This study considers key caching policy parameters, such as cache-hit ratio, stretch, and latency, to enhance the IoMT network performance. The proposed framework is evaluated on selected parameters through simulations while exploring various configurations involving network topology, cache size, and content popularity. An analysis of the results demonstrates the superior performance of the proposed caching framework, known as DCS, compared to others. The dynamic nature of the DCS approach yields significant results as compared to existing methods. In the future, it is intended to explore the pertinence of DCS in more scenarios and further optimize the data exchange processes. The proposed policy can be improved by including other related parameters including memory management of sensor nodes, security, etc.

Author Contributions: Conceptualization, M.S., A.A. and M.T.; methodology, M.S. and A.A.; software, M.T.; validation, M.T., M.S.K. and M.S.; formal analysis, M.S. and M.S.K.; investigation, M.T.; resources, A.A. and M.T.; data curation, M.S.K., M.T. and M.S.; writing—original draft preparation, M.T.; writing—review and editing, M.S., M.S.K. and A.A.; visualization, M.S.; supervision, M.T.; project administration, A.A. and M.S.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Qassim University under project (2023-FFT-2-HSRC-37648).

Data Availability Statement: The datasets used are available publicly.

Acknowledgments: Researchers would like to thank the Deanship of Scientific Research, Qassim University for funding publication of this project.

Conflicts of Interest: There is no conflict of interest to report.

References

1. Amadeo, M.; Campolo, C.; Quevedo, J.; Corujo, D.; Molinaro, A.; Iera, A.; Aguiar, R.L.; Vasilakos, A.V. Information-centric networking for the internet of things: Challenges and opportunities. *IEEE Netw.* **2016**, *30*, 92–100. [[CrossRef](#)]
2. Nour, B.; Sharif, K.; Li, F.; Biswas, S.; Mounghla, H.; Guizani, M.; Wang, Y. A survey of Internet of Things communication using ICN: A use case perspective. *Comput. Commun.* **2019**, *142*, 95–123. [[CrossRef](#)]
3. Quevedo, J.; Corujo, D.; Aguiar, R. Consumer driven information freshness approach for content centric networking. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 27 April–2 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 482–487.
4. Shajaiiah, H.; Sengupta, A.; Abdelhadi, A.; Clancy, C. Performance Trade-offs in IoT Uplink Networks under Secrecy Constraints. In Proceedings of the 2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), Istanbul, Turkey, 8 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
5. Abdullahi, I.; Arif, S.; Hassan, S. Survey on caching approaches in information centric networking. *J. Netw. Comput. Appl.* **2015**, *56*, 48–59. [[CrossRef](#)]
6. Din, I.U.; Hassan, S.; Khan, M.K.; Guizani, M.; Ghazali, O.; Habbal, A. Caching in information-centric networking: Strategies, challenges, and future research directions. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 1443–1474. [[CrossRef](#)]
7. Alkhazaleh, M.; Aljunid, S.; Sabri, N. A review of caching strategies and its categorizations in information centric network. *J. Theor. Appl. Inf. Technol.* **2019**, *97*, 19.
8. Ascigil, O.; Reñé, S.; Xylomenos, G.; Psaras, I.; Pavlou, G. A keyword-based ICN-IoT platform. In Proceedings of the 4th ACM Conference on Information-Centric Networking, Berlin, Germany, 26–28 September 2017; pp. 22–28.
9. Tarnoi, S.; Suksomboon, K.; Kumwilaisak, W.; Ji, Y. Performance of probabilistic caching and cache replacement policies for content-centric networks. In Proceedings of the 39th Annual IEEE Conference on Local Computer Networks, Edmonton, AB, Canada, 8–11 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 99–106.
10. Bedi, G.; Venayagamoorthy, G.K.; Singh, R.; Brooks, R.R.; Wang, K.C. Review of Internet of Things (IoT) in electric power and energy systems. *IEEE Internet Things J.* **2018**, *5*, 847–870. [[CrossRef](#)]
11. Wortmann, F.; Flüchter, K. Internet of things. *Bus. Inf. Syst. Eng.* **2015**, *57*, 221–224. [[CrossRef](#)]
12. Bosunia, M.R.; Hasan, K.; Nasir, N.A.; Kwon, S.; Jeong, S.H. Efficient data delivery based on content-centric networking for Internet of Things applications. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1550147716665518. [[CrossRef](#)]
13. Liu, X.; Ravindran, R.; Wang, G.Q. Information Centric Networking Based Service Centric Networking, 2019. U.S. Patent 10,194,414, 19 January, 2019.
14. Palattella, M.R.; Dohler, M.; Grieco, A.; Rizzo, G.; Torsner, J.; Engel, T.; Ladid, L. Internet of things in the 5G era: Enablers, architecture, and business models. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 510–527. [[CrossRef](#)]
15. Lindgren, A.; Abdesslem, F.B.; Ahlgren, B.; Schelén, O.; Malik, A.M. Design choices for the IoT in information-centric networks. In Proceedings of the 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), Vegas, NV, USA, 9–12 January 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 882–888.
16. Xu, C.; Wang, X. Transient content caching and updating with modified harmony search for Internet of Things. *Digit. Commun. Netw.* **2019**, *5*, 24–33. [[CrossRef](#)]
17. Arshad, S.; Azam, M.A.; Rehmani, M.H.; Loo, J. Recent advances in information-centric networking-based Internet of Things (ICN-IoT). *IEEE Internet Things J.* **2018**, *6*, 2128–2158. [[CrossRef](#)]
18. Siris, V.A.; Thomas, Y.; Polyzos, G.C. Supporting the IoT over integrated satellite-terrestrial networks using information-centric networking. In Proceedings of the 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Larnaca, Cyprus, 21–23 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5.
19. Malik, N.; Sardaraz, M.; Tahir, M.; Shah, B.; Ali, G.; Moreira, F. Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds. *Appl. Sci.* **2021**, *11*, 5849. [[CrossRef](#)]
20. Malik, S.; Tahir, M.; Sardaraz, M.; Alourani, A. A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques. *Appl. Sci.* **2022**, *12*, 2160. [[CrossRef](#)]
21. Mohiyuddin, A.; Javed, A.R.; Chakraborty, C.; Rizwan, M.; Shabbir, M.; Nebhen, J. Secure cloud storage for medical IoT data using adaptive neuro-fuzzy inference system. *Int. J. Fuzzy Syst.* **2022**, *24*, 1203–1215. [[CrossRef](#)]
22. Ahad, A.; Tahir, M.; Aman Sheikh, M.; Ahmed, K.I.; Mughees, A.; Numani, A. Technologies trend towards 5G network for smart health-care using IoT: A review. *Sensors* **2020**, *20*, 4047. [[CrossRef](#)]
23. Song, Y.; Ma, H.; Liu, L. TCCN: Tag-assisted content centric networking for Internet of Things. In Proceedings of the 2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 14–17 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–9.
24. Meddeb, M.; Dhraief, A.; Belghith, A.; Monteil, T.; Drira, K. Cache coherence in machine-to-machine information centric networks. In Proceedings of the 2015 IEEE 40th Conference on Local Computer Networks (LCN), Clearwater Beach, FL, USA, 26–29 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 430–433.
25. Saba, T.; Haseeb, K.; Ahmed, I.; Rehman, A. Secure and energy-efficient framework using Internet of Medical Things for e-healthcare. *J. Infect. Public Health* **2020**, *13*, 1567–1575. [[CrossRef](#)]
26. Kaur, M.; Singh, D.; Kumar, V.; Gupta, B.B.; Abd El-Latif, A.A. Secure and energy efficient-based E-health care framework for green internet of things. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1223–1231. [[CrossRef](#)]

27. Rehman, A.; Saba, T.; Haseeb, K.; Larabi Marie-Sainte, S.; Lloret, J. Energy-efficient IoT e-health using artificial intelligence model with homomorphic secret sharing. *Energies* **2021**, *14*, 6414. [CrossRef]
28. Ming, Z.; Xu, M.; Wang, D. Age-based cooperative caching in information-centric networking. In Proceedings of the 2014 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, China, 4–7 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–8.
29. Vural, S.; Navaratnam, P.; Wang, N.; Wang, C.; Dong, L.; Tafazolli, R. In-network caching of Internet-of-Things data. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 3185–3190.
30. Chen, C.; Jiang, J.; Fu, R.; Chen, L.; Li, C.; Wan, S. An intelligent caching strategy considering time-space characteristics in vehicular named data networks. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 19655–19667. [CrossRef]
31. Vural, S.; Wang, N.; Navaratnam, P.; Tafazolli, R. Caching transient data in internet content routers. *IEEE/ACM Trans. Netw.* **2016**, *25*, 1048–1061. [CrossRef]
32. Amadeo, M.; Campolo, C.; Molinaro, A. Internet of things via named data networking: The support of push traffic. In Proceedings of the 2014 International Conference and Workshop on the Network of the Future (NOF), Paris, France, 3–5 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–5.
33. Zhang, Z.; Lung, C.H.; Lambadaris, I.; St-Hilaire, M. IoT data lifetime-based cooperative caching scheme for ICN-IoT networks. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
34. Niyato, D.; Kim, D.I.; Wang, P.; Song, L. A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
35. Shrimali, R.; Shah, H.; Chauhan, R. Proposed caching scheme for optimizing trade-off between freshness and energy consumption in name data networking based IoT. *Adv. Internet Things* **2017**, *7*, 11. [CrossRef]
36. Takatsuka, Y.; Nagao, H.; Yaguchi, T.; Hanai, M.; Shudo, K. A caching mechanism based on data freshness. In Proceedings of the 2016 International Conference on Big Data and Smart Computing (BigComp), Hong Kong, China, 18–20 January 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 329–332.
37. Hail, M.A.; Amadeo, M.; Molinaro, A.; Fischer, S. Caching in named data networking for the wireless internet of things. In Proceedings of the 2015 International Conference on Recent Advances in Internet of Things (RIoT), Singapore, 7–9 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
38. Meddeb, M.; Dhraief, A.; Belghith, A.; Monteil, T.; Drira, K.; AlAhmadi, S. Cache freshness in named data networking for the internet of things. *Comput. J.* **2018**, *61*, 1496–1511. [CrossRef]
39. Sandvine Intelligent Broadband Networks. Identifying and Measuring Internet Traffic: Techniques and Considerations. An Industry. Whitepaper, Version 2.20. 2015. Available online: <https://www.sandvine.com/resources?filter=.whitepapers> (accessed on 26 October 2023).
40. Che, H.; Wang, Z.; Tung, Y. Analysis and design of hierarchical web caching systems. In Proceedings of the IEEE INFOCOM 2001, Conference on Computer Communications, Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), Anchorage, AK, USA, 22–26 April 2001; IEEE: Piscataway, NJ, USA, 2001; Volume 3, pp. 1416–1424.
41. Hefeeda, M.; Saleh, O. Traffic modeling and proportional partial caching for peer-to-peer systems. *Ieee/Acm Trans. Netw.* **2008**, *16*, 1447–1460. [CrossRef]
42. Koplenig, A. Using the parameters of the Zipf–Mandelbrot law to measure diachronic lexical, syntactical and stylistic changes—A large-scale corpus analysis. *Corpus Linguist. Linguist. Theory* **2018**, *14*, 1–34. [CrossRef]
43. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking named content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 1–4 December 2009; pp. 1–12.
44. Saino, L.; Psaras, I.; Pavlou, G. Icarus: A caching simulator for information centric networking (ICN). In Proceedings of the SIMUTools 2014: 7th International ICST Conference on Simulation Tools and Techniques, Lisbon, Portugal, 17–19 March 2014; ICST: Lisbon, Portugal, 2014; Volume 7, pp. 66–75.
45. Bevywise IoT Simulator, 2021. Available online: <https://www.bevywise.com/iot-simulator/help-document.html> (accessed on 19 July 2021).
46. Hady, A.A.; Ghubaish, A.; Salman, T.; Unal, D.; Jain, R. Intrusion detection system for healthcare systems using medical and network data: A comparison study. *IEEE Access* **2020**, *8*, 106576–106584. [CrossRef]
47. Duan, P.; Jia, Y.; Liang, L.; Rodriguez, J.; Huq, K.M.S.; Li, G. Space-reserved cooperative caching in 5G heterogeneous networks for industrial IoT. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2715–2724. [CrossRef]

48. Jin, H.; Xu, D.; Zhao, C.; Liang, D. Information-centric mobile caching network frameworks and caching optimization: A survey. *Eurasip J. Wirel. Commun. Netw.* **2017**, *2017*, 1–32. [[CrossRef](#)]
49. Naeem, M.A.; Ali, R.; Kim, B.S.; Nor, S.A.; Hassan, S. A periodic caching strategy solution for the smart city in information-centric Internet of Things. *Sustainability* **2018**, *10*, 2576. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.