EABC: Energy-aware Centrality-based Caching for Named Data Networking in the IoT

Xingyun He, Hu Liu, Wuyungerile Li**
Inner Mongolia Key Lab. of Wireless Networking & Mobile Computing
Engineering Research Center of Ecological Big Data, Ministry of Education
School of Computer Science, Inner Mongolia University
Hohhot, China
32109029@mail.imu.edu.cn, 575755734@qq.com, **gerile@imu.edu.cn

Alvin Valera, Winston K.G. Seah
Wireless Networks Research Group
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
{alvin.valera,winston.seah}@ecs.vuw.ac.nz

***Abstract*—Named Data Networking (NDN) is an information-centric internet architecture that delivers packets based on the name of the content in the packet. A key component of NDN is the caching strategy designed to reduce total network latency and load on content producers. To improve the speed and reliability of web content delivery, existing caching strategies typically cache content on a large number of intermediate nodes, which incur significant energy consumption and memory overhead. However, in Internet of Things (IoT) scenarios, memory and energy of nodes are scarce resources. Therefore, in NDN-based IoT applications, traditional caching strategies can cause node failures due to energy depletion, which can significantly reduce the network operational lifetime as well as create problems that caching is supposed to solve. In this paper, a caching strategy based on node centrality and energy availability, called Energy-aware Approximate Betweenness Centrality (EABC) is proposed for NDN-based IoT. EABC uses a topology-based heuristic to cache data content on nodes with high centrality and makes caching decisions based on the remaining energy of the nodes. We evaluate EABC using simulations based on ndnSIM in different topologies and compare it with several existing NDN caching strategies. The results show that EABC performs better in different types of network topologies, reduces the average transmission delay of data and balances the energy consumption of highly central nodes, thus extending the network lifetime.**

***Index Terms*—Cache Decision Strategy, NDN, IoT, Betweenness Centrality, Energy**

I. Introduction

In the current era of digitization, the exchange of information and the flow of data have become pivotal driving forces for social and economic development. The Internet of Things (IoT), through connecting various physical devices to the internet, has triggered an unprecedented surge in data. Information-Centric Networking (ICN), with its content-centric characteristics, emerges as an ideal choice for the Internet of Things (IoT), facilitating the effective management and distribution of large-scale IoT data. In this network paradigm, data revolves around content, completely abandoning dependence on specific address identifiers. It is against this backdrop that our attention is directed towards a specific ICN implementation—Named Data Networking (NDN). Given that IoT data are typically identified by content rather than addresses, NDN, as a concrete manifestation of information-centric networking, demonstrates unique advantages in addressing this challenge.

The NDN architecture adopts a content-centric paradigm, and caching is an important mechanism in NDN. The cache in an NDN node holds the requested data for faster response in subsequent requests, thus reducing network latency and congestion. It has also been proven that caching reduces the energy consumption of the entire network [1]. In the IoT, problems such as network latency and congestion are more prominent due to the huge number of devices and limited network bandwidth, and by applying NDN to the IoT, these problems can be effectively addressed and network efficiency can be improved. Therefore, NDN caching mechanism plays an important role in IoT and is expected to improve the performance and reliability of IoT applications.

Most studies on caching in NDN only consider features such as content or topology, and the proposed caching strategies are not applicable in the IoT because IoT devices are usually resource-constrained and have limitations in terms of processing power, memory, and energy compared to the traditional Internet. In IoT, node energy consumption is an important factor that cannot be ignored, and once a node is in a low or depleted energy state, its usability value is low as its remaining operational lifetime is limited [2]. Once its energy supply is depleted, data that were cached on the expired node will be lost and subsequent requests for the same data will have to be satisfied by other nodes that may have cached those data, or in the worst case, by the producers themselves. Besides link instability and even potential network failure if the node's expiration lead to the network becoming partitioned, interest packets will have to travel further in order for the requests to be fulfilled and likewise, data packets will traverse long distances to the consumers. Instead of achieving its key objectives of reducing network traffic and latency to fulfill requests, caching without nodes' energy consideration leads to opposite adverse consequences.

At the same time, topology and caching are closely linked,

**Corresponding author

and the IoT has multiple possible topologies, which leads to the possibility that a certain caching policy may be very effective in one type of topology and not work at all in another. In addition, if the network topology is dynamic, the ideal cache location may change over time. Therefore, the optimal caching policy should be able to be applied to all types of topologies. Considering the characteristics of IoT, e.g., constrained devices, limited energy, and variable topology, we propose a new caching policy, Energy-aware Approximate Betweenness Centrality (EABC), which aims to achieve a balance of content delivery latency and energy. From our performance evaluations, we demonstrate that EABC reduces content delivery latency, improves cache hit rate, balances network energy consumption, and extends network lifetime. At the same time, in order to cope with the changing topologies of IoT, EABC is also topology-agnostic, such that it can perform well in many different topologies.

The rest of this paper is organized as follows. Section II discusses related work on NDN caching strategies. In Section III, we describe our proposed NDN-IoT caching strategy, EABC, and provide a specific example. Section IV then presents simulation results for the energy consumption, cache hit rate, and delay of the EABC caching strategy in IoT topologies. To show the topology-agnostic feature of EABC, we evaluate its performance in both edge and core topologies, before we conclude in Section V.

II. Related Work

Caching, which is an integral part of the NDN architecture, determines which objects are placed on which cache nodes [3]. Whenever an NDN node receives a content block that is not yet available in its Content Store (CS), it needs to make a decision on whether to keep a copy, i.e. cache the content.

Cache Everything Everywhere (CEE) [3] is the simplest caching decision strategy, which is the default strategy used in traditional NDN. Whenever a piece of content passes through a node that does not already have a copy of it in its CS, that content will be cached by placing a copy in the CS. Leave Copy Down (LCD) [4], proposed by Laoutaris *et al.*, always caches the content in the next hop node where the cache hit occurs. However, in densely connected networks, CEE and LCD can lead to high redundancy, which is unsuitable for networks with severely limited caching capabilities.

Based on the need to increase caching diversity and reduce caching redundancy, Zhang *et al.* proposed Prob [5], which uses probability to determine whether to cache content. Hail *et al.* proposed pCASTING [6], which dynamically calculates the caching probability of each node and even each content block based on available node information such as content block freshness, node battery power, and cache occupancy. However, probability-based caching strategies have a degree of randomness as well as implementation constraints.

To address the challenges posed by large-scale IoT, such as real-time video, streaming video, and high-traffic data services, Hasan *et al.* [7] proposed an efficient cluster-based caching mechanism and content popularity method to improve content availability and reduce distribution time. Gupta *et al.* [8] designed a hierarchical network architecture based on ICN-IoT, which determines the content to cache on network nodes by comprehensively considering node centrality and implements near-path caching to reduce overall network traffic. Naeem *et al.* [9] proposed a hybrid caching strategy that selects content based on request frequency and efficiently caches it at the edge and leader nodes to improve the delivery efficiency of multimedia content. Additionally,in [10] introduced a caching strategy to make decisions based on the popularity and lifetime of IoT content, ensuring the delivery of fresh IoT data. However, while these caching methods have made significant progress in improving efficiency, it is worth noting that they have not yet fully accounted for the energy constraints of IoT devices.

Energy efficiency plays a pivotal role in the realm of the IoT [11]. In [12], authors proposed an energy-efficient approach based on the information-centric networking (ICN) paradigm using distributed caching of IoT content. Hahm *et al.* [13] exploited content names, utilizing smart interplay between cooperative caching and power-saving sleep capabilities on IoT devices. This enabled each device to automatically set names and adjust parameters to reduce energy consumption. Gupta *et al.* proposed an energy-efficient placement method for content chunks suitable for the vehicular environment [14]. The strategy considered the residual power of the current carrier, the local popularity of the content, and the cache gain to determine the placement of content chunks, which reduced content duplication. Caching content also expends the energy of an IoT node. Hence, Serhane *et al.* [15] proposed an Energy-aware caching placement scheme (EaCP) that aims to maximize the energy-saving by balancing content transmission energy and content caching energy. Similarly, co-operation between content caching and transmission has been proposed and validated using an experimental testbed [16].

As expected, various machine learning approaches have been proposed for energy-efficient caching in IoT networks. Deep Reinforcement Learning (DRL) approaches have been proposed for use in cache-enabled device-to-device (D2D) networks with considerations on device mobility and content popularity [17]. Similarly, a DRL-based caching scheme has been shown to improve the cache hit rate and reduces energy consumption of the IoT networks while taking data freshness and limited lifetime of IoT data into account [18]. It is unclear how such approaches can be implemented on resource-constrained IoT devices. Hence, DRL and other learning approaches are typically applied to caching of IoT services and data on edge servers [19] and not the IoT devices within the networks.

There is a class of NDN caching strategies that use betweenness centrality to determine where to cache content. Betweenness centrality has been found to be a measure of the importance of nodes in a network [20]. Caching on more "important" nodes will benefit the caching performance and thus can be applied to NDN caching. In NDN, it can be assumed without loss of generality, that the shortest path between i and j is used as the transmission path of the content. Therefore, the betweenness centrality of an arbitrary node v can then be defined as follows:

$$C_B(v) = \sum_{i \neq v \neq j \neq V} \sigma^{'}_{i,j}(v) \tag{1}$$

$$\sigma^{'}_{i,j}(v) = \begin{cases} 1, & if\ v\ on\ path\ (i,j) \\ 0, & otherwise \end{cases} \tag{2}$$

where V is the set of all nodes in the network, and $\sigma^{'}_{i,j}(v)$ indicates whether the shortest path from i to j passing through node v.

Chai *et al.* [21] proposed two methods to determine where to cache content based on the centrality of a node: *Betw* and *EgoBetw*. Betw is the most straightforward implementation of betweenness centrality. Before the nodes start exchanging interest and data packets, the network uses Eqn. (1) to assign the centrality values to all nodes. Due to the high complexity of Betw, a more lightweight alternative called EgoBetw was proposed that does not require global network information; instead, nodes only exchange information with their neighbor nodes in one adjacent hop to compute an approximation of their centrality value. However, if Betw and EgoBetw are applied to real scenarios, both need to repeatedly calculate the node centrality values when the topology changes, thus incurring additional communication and computational overheads, and changes in the topology can be very frequent depending on the deployment scenario.

In view of the difficulties in practical deployment of Betw/EgoBetw and resource contraints of IoT devices, Pfender *et al.* [22] proposed the Approximate Betweenness Centrality (ABC) caching strategy, which can approximate the node's centrality value based on network traffic without additional overhead and can easily adapt to dynamic topologies when calculating the centrality value. ABC embeds unique identifiers for producers and consumers in the interest packet, allowing each node processing the interest packet to determine its location on the path between the consumer and the producer. This enhances the node's knowledge about the delivery path it is on. Over time, by tracking the pairs of nodes it serves, each node can approximate its own numerical value, eventually converging to the value computed by Betw during the setup phase. The ABC caching strategy can easily adapt to the dynamic topology by processing the information stored on the record path. However, in IoT, nodes with higher centrality values will experience high load levels and have more traffic passing through them leading to greater battery consumption. When a node's energy is depleted, the node as well as the connectivity it provides fail. Dealing with nodes failures and providing energy-awareness when caching are key concerns of recent research [23].

As the centrality-based caching policy picks nodes with higher centrality values to cache, these "more important" nodes in the network incur higher energy consumption, while other nodes have a surplus of energy. With energy-depleted nodes in key locations, the network becomes disconnected and the consumers at the edge cannot get data from the producers resulting in performance degradation. Therefore, a caching policy needs to make energy-aware caching decisions to achieve a balance and optimization of content delivery latency and energy in order to extend the operational lifetime of the network.

III. EABC CACHING PLACEMENT STRATEGY

The Energy-aware Approximate Betweenness Centrality (EABC) caching policy aims to address the node energy consumption imbalance problem in ABC [21]. The policy achieves energy consumption optimization by offloading the content of nodes with high centrality values to nodes closer to the consumers, so that the energy of nodes with high centrality values is not easily and quickly depleted.

A. Main Idea

EABC uses energy as a metric for high centrality value nodes to influence caching decisions before energy depletion, and cache content in other nearby nodes. Initially, a consumer publishes interest packets into the network, which are forwarded to producers with content by using the corresponding forwarding policies. When the interest packet arrives at an intermediate node, it queries its CS table, the Pending Interest Table (PIT) and the Forwarding Information Base (FIB, or forwarding table). Where necessary, the node updates the centrality value in the packet so that the interest packet carries the largest centrality value in the path. When the interest packet is satisfied at the producer or an intermediate node with a copy of the content in its CS, a data packet is generated and sent back to the consumer along the reverse path of the interest packet. As the data packet travels back to the consumer, the decision to cache the content is made based on centrality as well as the node's remaining energy.

B. Packet Structure Design

In order to utilize centrality and energy metrics in the caching decision, it is necessary to extend the Interest and Data packets of NDN, as shown in Fig. 1. Firstly, a betweenness centrality field Betw is added to the Interest packet, which records the maximum node betweenness centrality value of the path to the producer. At the same time, the Data packet also adds a betweenness centrality field Betw, which is used to compare with the node's betweenness centrality value to determine whether the Data packet should be cached. In addition, the Data packet also includes an EnergyFlag field that records the remaining energy of the node, enabling load

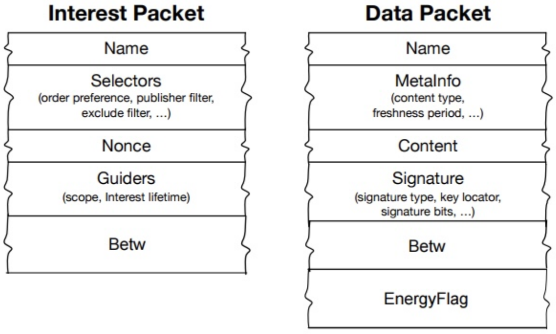


Fig. 1: Packet formats of EABC

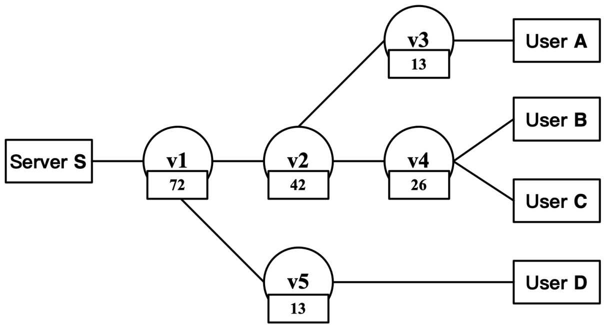


Fig. 2: Topological example of EABC approach

balancing caching decisions to be made when the remaining energy is insufficient.

C. EABC Caching Placement Strategy Design

The EABC caching decision policy is based on a combination of a node's centrality value and its remaining energy, as well as the packet's content. When the interest packet passes through a node, the node compares the centrality value in the packet's Betw field against its centrality value. If the node's centrality is higher, then it updates the packet's Betw field with its centrality value. When an interest packet hits a cache, an EnergyFlag is added to the resulting data packet and its initial value is set to 0. The significance of the flag is to determine whether the packet is a "new" packet or an "old" packet that was forwarded by due to low energy. When a node receives a data packet, it first reads the flag bit to determine if the packet is a "new" packet. When the remaining energy of the node is higher than a predefined threshold, m, it means that the current node satisfies the energy requirement for caching; then the node's centrality value is compared with the centrality value contained in the packet to decide whether to cache the data. When the energy of the node is lower than this threshold, the packet is forwarded to the next hop node closer to the consumer.

When a data packet is not cached at a node due to insufficient energy, the EnergyFlag in the packet will be set to 1, indicating that the packet is "old" and, subsequently, centrality will not be used as a caching criterion when passing through a node with sufficient energy. Additionally, the strategy considers a scenario where the entire path that the packet passes through is low on energy. According to the caching decision strategy, data packets cannot be cached in this situation, which is detrimental to network performance. To address this issue, the strategy increases the EnergyFlag by 1 each time a packet cannot be cached due to insufficient energy at a node. When the EnergyFlag value exceeds 2, the data will be cached unconditionally. This caching decision strategy, without increasing cache redundancy, not only allows the node with the highest centrality to have a lower load, which balances the load of relay nodes and prolongs the network lifetime; it also caches the content to the edge location closer to the consumers, and therefore reduces the average content access latency. The pseudocode for the strategy is shown in Algorithm 1.

Algorithm 1 Energy-aware Approximate Betweenness Centrality

```
function HANDLE-DATA(Data)
    if Data.Energyflag = 0 then
        if myCentrality > Data.Centrality then
            if myRemainingEnergy > m then
                cache(Data)
            else
                Data.EnergyFlag ← 1
            end if
            forward(Data)
        end if
    else
        if myRemainingEnergy > m then
            cache(Data)
            Data.EnergyFlag ← 0
        else
            Data.EnergyFlag ← Data.EnergyFlag + 1
            if Data.Energyflag > 2 then
                cache(Data)
                Data.EnergyFlag ← 0
            end if
        end if
    end if
end function
```

Since the betweenness caching strategy has a high correlation with the topology, a topology example is used to demonstrate the EABC approach. As shown in Fig. 2, there is one producer S and four consumers A, B, C, and D. **v1**-**v5** are all relay nodes, and the node subscripts are their respective centrality values. When the network is running for a period of time, the ABC policy can approximate the centrality values of each node. As time passes, **v1** becomes the node with the highest centrality value among the three paths S-A, S-B/C, and S-D. Accordingly, the content will be cached in **v1**, so the load on **v1** will be high and its energy will be depleted quickly. When the energy of node **v1** drops below the cache energy threshold, m, **v1** stops caching new content; instead, its next hops **v2** and **v5** nearer the consumers will cache the

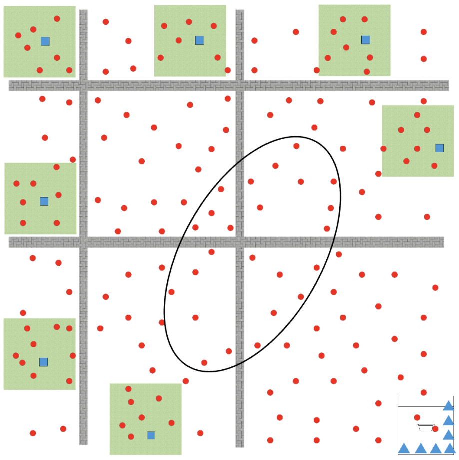


Fig. 3: Smart agricultural topology

TABLE I: Simulation Parameters

Parameter	Value
Content catalog size	1000 contents
Content size	1200*8 bytes
Consumer request rate	1 content/s
Node cache size	5
Simulation time	500s
Network topology	240m×240m random (Uniform)
Transmission Technology	IEEE802.11g
Transmission rate	6 Mbps
Delay model	Constant Speed Propagation Delay Model
Loss model	Log Distance Propagation Loss Model
Transmission range	15m
Number of node	175
Number of producers	7
Number of consumers	7
Link energy density	$1.5*10^{-9}$ J/bit
Router energy density	$2*10^{-8}$ J/bit
Cache energy consumption	$1*10^{-9}$ W/bit

content. Subsequent interest packets sent by consumers will be satisfied at **v2** and **v5**, so that the caching decision not only balances the load of the high centrality node **v1**, extending its lifetime, but also caches the content closer to the consumers, thus reducing content delivery latency.

IV. PERFORMANCE EVALUATION

This section discusses validation of the EABC strategy and performance evaluation using simulation. The study assumes an IoT application in smart agriculture, and the performance based on energy, cache hit rate, and latency are analyzed and evaluated. Finally, the performance of the strategy in edge and core topologies is compared to validate the topology-agnostic property of the strategy.

A. Simulation Setup

This study used the ndnSIM network simulation platform to verify the performance of EABC. The ndnSIM version is 2.7, and the operating system is Ubuntu 16.04 LTS 64bit. The simulation experiment mainly targets the smart agriculture topology, with a total of 175 nodes, including 161 relay nodes, randomly distributed uniformly in a 240x240m rectangular pasture, as shown in Fig. 3. Among them, seven producers are sensor nodes, shown as squares in the figure, used to detect environmental data around the pasture. Seven consumers, shown as triangles in the figure, request data from producers in the data receiving area at a rate of one interest packet per second. Different producers and different consumers have five matching prefixes, enabling different consumers to collect environment data packets sent by corresponding producers.

The forwarding strategy of the network uses the Content Connectivity and Location Aware Forwarding (CCLF) forwarding strategy [24], which can easily adapt to wireless multihop networks while better addressing the adverse effects of network storms in communication. The cache replacement policy uses the default Least Recently Used (LRU) [3]. In the IoT, there are numerous ways in which a node consumes energy, including the energy consumed in standby, the energy consumed by transmitting packets, the energy consumed by caching and replacing packets, etc. The energy consumption is mainly attributed to content caching and data transmission [25]. We use the energy model in ns-3 to simulate the energy consumption of wireless devices in a real scenario. The model allows to specify the initial energy of each device and during the simulation it decreases this value according to the activity of the nodes. The simulation time is 500 seconds. The settings of the simulation parameters are shown in Table. I.

We also implemented four commonly used cache placement algorithms in NDN as comparative algorithms, namely, CEE [3], Prob [5], pCASTING [6], and ABC [22].

B. Establishing the Energy Threshold m

When making caching decisions, the policy determines whether the percentage of energy remaining in the node is above the energy threshold m, and if so, the node caches new data (cf: Step 4 in Algorithm 1.) When m is low, the node with the highest centrality value will experience high load for longer periods and run out of energy quickly. When the value of m is high, the nodes with high centrality values will be passed over prematurely, thus increasing the content delivery delay. Hence, through simulation experiments, we assessed the impact of various m values on content delivery delay and node energy consumption, ultimately identifying 0.5 as the optimal m value, as depicted in Table. II.

When the energy threshold is set to 0.5, it means that EABC will make cache decisions when the node's energy level drops to 50% of its initial energy. This threshold value provides the optimal network lifetime and latency. When the energy threshold is set below 0.5, the network lifetime will decrease significantly because the high load of nodes with high betweenness centrality will not be balanced until later. EABC caches content closer to consumers, thus reducing latency. When the energy threshold is set above 0.5, the strategy will abandon nodes with high betweenness centrality earlier, forcing content retrieval from farther producers, resulting in

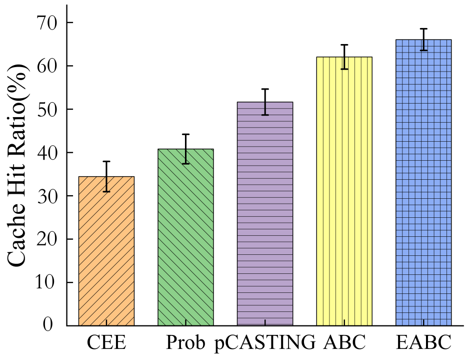


Fig. 4: Cache Hit Ratio

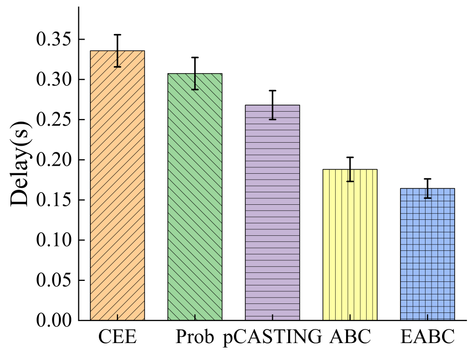


Fig. 5: Delay

TABLE II: Energy threshold evaluation

Energy Threshold m	0.3	0.4	0.5	0.6	0.7
Network Lifetime (s)	413	428	438	426	410
Average Delay (s)	0.1703	0.1662	0.1643	0.1668	0.1709

decreased network lifetime and increased average latency. The experiments in this paper are conducted with an energy threshold of 0.5.

C. Simulation Results

1) Cache Hit Ratio: Fig. 4 shows the cache hit ratio of each strategy. Due to the small cache space set for each node, which is in line with IoT scenarios, CEE, Prob, and pCASTING may have difficulty obtaining content from nodes other than producers and have lower cache hit rates as a result. On the other hand, both ABC and EABC cache content in the nodes with the highest betweenness centrality, and data packets frequently pass through these nodes, resulting in higher cache hit rates. Since EABC caches content closer to the consumer location after the node's energy falls below the energy threshold, it increases the diversity of network content and improves cache hit rates. Compared with the ABC strategy, EABC's cache hit rate is improved by about 6.44%.

2) Delay: The delay performance of the different caching strategies is shown in Fig. 5. The low cache hit rate of CEE, Prob, and pCASTING also means that they are not obtaining cached content from intermediate nodes and have to obtain content from producers, resulting in a higher delay. On the other hand, ABC and EABC cache content at centrally located nodes, making it easier for consumers to obtain the required content, thus data packets travel shorter distances and have lower delays. As EABC caches content closer to the consumer when the node energy is below the energy threshold, the delay is further reduced. Compared to ABC, EABC improves the delay performance by about 12.6%.

3) Node Energy: In the simulation scenario, the energy model in ndnSIM is used to calculate the energy consumption of wireless devices. The model allows to specify the initial energy of each device and during the simulation it decreases this value according to the activity of the nodes [6].

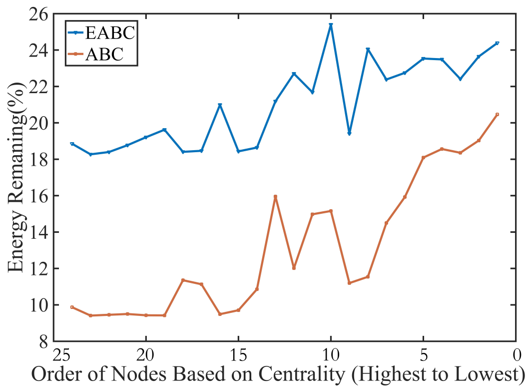


Fig. 6: Remaining energy of high-centrality nodes

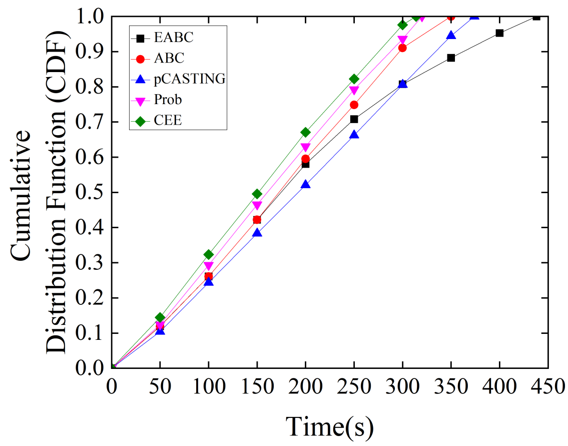


Fig. 7: CDF of energy over time

Fig. 6 illustrates the remaining energy of high-centrality nodes under the EABC and ABC caching strategies at the same time before the simulation ends. Nodes with high-centrality are crucial in the network, and when these nodes' energy is depleted, the network's lifetime comes to an end.

Therefore, monitoring the remaining energy of these high-centrality nodes can gauge the network's lifetime.

The nodes are sorted in descending order based on their betweenness centrality values. Since betweenness centrality is calculated by counting the number of data packets passing through a node, higher values indicate greater data traffic through that node, providing an approximate measure of the node's load. From the graph, it is evident that the betweenness centrality values of nodes are roughly inversely proportional to their remaining energy.

Nodes using the EABC strategy have higher remaining energy than those using the ABC strategy. EABC balances the energy of high-centrality nodes closer to the consumers, reducing their load. As a result, these nodes have more remaining energy over the same working time, and the remaining energy values among nodes are more balanced. Meanwhile, the ABC strategy shows a large variance in remaining energy, indicating that nodes with high-centrality experience high loads, while other relay nodes' energy remains underutilized, leading to resource wastage. Under the EABC strategy, the remaining energy variance of high-centrality nodes is smaller, demonstrating that the strategy reduces the energy consumption of these nodes, balances the load of relay nodes, and prolongs the network's lifetime.

When a high-centrality node lacks sufficient energy, the EABC caches data at the next-hop node meeting the energy requirement, bringing the data closer to the consumer. This reduces the burden on high-centrality nodes without increasing the number of network caches, allowing them to retain more energy after the same work period. As a result, a more balanced level of residual energy is achieved for each node. In contrast, the ABC exhibits significant variance in residual energy, signifying that high-centrality nodes bear a heavy load, while other relay nodes' energy remains unused, resulting in resource wastage. Conversely, the EABC reduces energy consumption for high-centrality nodes and balances the load of relay nodes, thereby extending the network's lifetime.

By plotting the cumulative distribution function (CDF) of energy and time, it is possible to visualize the network lifetime of each algorithm under the same conditions. It is also possible to determine the remaining energy of each router with different algorithms and, at the same time, to derive the energy consumption distribution of the whole network. The CDF curves for energy discharge time of the five caching strategies are shown in Fig. 7. The closer the value on the y-axis is to 1, the closer the node is to running out of energy, while the x-axis represents the network's lifespan. It can be observed that the energy consumption rate of the EABC algorithm is initially similar to that of ABC, but when a node's energy drops to the energy threshold, the energy consumption rate slows down, and the network's lifetime is greatly improved compared to the ABC caching strategy, with an average improvement of about 25.1%.

TABLE III: Additional Simulation Parameters for Topology-Agnostic Validation

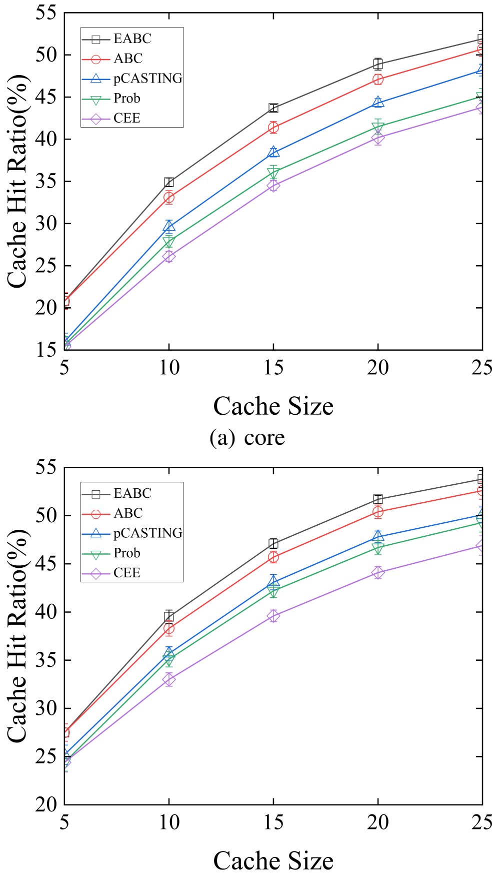
Parameter	Value
Node cache size	5/10/15/20/25
Transmission range	25m
Number of node	52/59
Number of producers	1
Number of consumers	18/30

Compared to the dynamic probability caching of pCASTING, since EABC uses the nodes' remaining energy for caching decisions from the beginning of the network operation, EABC's performance is not as good as pCASTING at the beginning of the experiment. However, as time progresses, the remaining energy of nodes decreases, and the remaining space of nodes becomes smaller, resulting in a decrease in the dynamic caching probability of pCASTING, making it difficult to cache content closer to the consumers, and the need to obtain content from farther nodes.

D. Topology-agnostic Verification

In NDN, caching policies are generally affected by topology, and different caching policies will have different performance in different topologies. The types of topologies are divided into two types when considering extreme cases: *edge* topologies and *core/centre* topologies, which are at the two extreme variants of all topologies. IoT topology typically falls between core and edge topologies. If the EABC strategy demonstrates good performance under both core and edge topologies, it can be inferred that it is also suitable for other topological structures. To verify the topology-agnostic property of the EABC caching policy, this experiment evaluates the cache hit rate, average content retrieval delay, and node energy in these two types of topologies, respectively. The specific experimental parameters are configured as shown in Table. III. Simulation results are averaged over 10 independent runs and reported with the 95% confidence intervals.

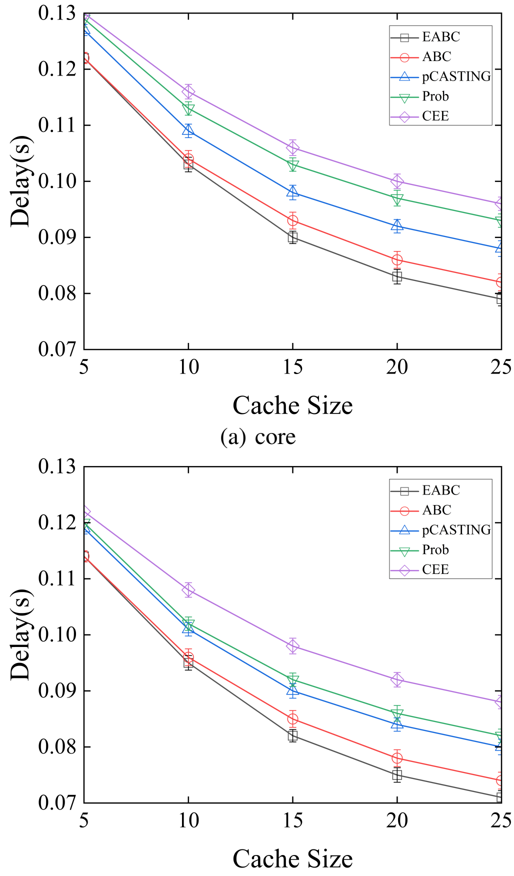
1) Cache Hit Ratio: The five different caching strategies with varying cache capacities were evaluated in both core and edge topologies, and results are shown in Fig. 8. Due to the limitations of IoT scenarios, the node cache space is set to a very small size. This makes it more challenging for the CEE to fetch content from the intermediate router and, instead, fetches it directly from the producer, resulting in a low cache hit ratio. However, Prob and pCASTING cache different content on various nodes, representing a certain improvement in cache hit ratio compared to CEE. Nonetheless, this improvement is still constrained by cache capacity. On the other hand, both ABC and EABC cache content at the node with the highest betweenness centrality, and interest packets and packets frequently pass through this node, so they have a high cache hit rate. When the cache capacity is 10 content blocks, the cache hit ratio of EABC in the core topology improves by 5.43% compared to ABC. The cache hit ratio of EABC in the edge topology improves by 3.13% compared to



(a) core

(b) edge

Fig. 8: Cache Hit Ratio



(a) core

(b) edge

Fig. 9: Delay

ABC. Therefore, it can be seen that EABC performs better than other algorithms in both edge and core topologies in improving the network-wide cache hit ratio, proving that it is topology-agnostic.

2) Delay: The trend of the average transmission delay of five caching algorithms with the change of cache capacity is shown in Fig. 9. It can be seen that the average latency of EABC and ABC in both topologies is close to each other, but the latency advantage is greater than that of CEE, Prob and pCASTING. When the cache capacity is 10 content blocks, the latency of EABC in the core topology improves by 0.96% compared to ABC, and the latency of EABC in the edge topology improves by 1.04% compared to ABC. Therefore, it can be seen that EABC has better performance in reducing the average transmission delay in both edge topology and core topology compared to other algorithms.

3) Node Energy: Since the other caching strategies do not consider the IoT application scenarios, this experiment only analyzes the two caching strategies EABC and ABC. Nodes with lower residual energy represent higher load and are colored darker in the figure, while nodes with higher residual energy represent lower load and are colored lighter. Fig. 10(a) and Fig. 10(b) show the energy hotspot distribution of ABC and EABC in the core topology, the ABC strategy results in high-centrality nodes experiencing a high level of load, causing their remaining energy to deplete rapidly compared to edge routers and consumers. As a consequence, before high-centrality nodes exhaust their energy, edge nodes still have a surplus of energy. By this stage, the network ceases to function, resulting in the unused energy of nodes with abundant power being wasted. On the other hand, EABC unloads the content cached by high-centrality nodes to locations closer to consumers. This prevents the rapid depletion of energy in high-load nodes, thus avoiding the scenario where the exhaustion of energy in crucial nodes leads to network paralysis. Fig. 11(a) and Fig. 11(b) show the energy hotspot distribution of ABC and EABC in the edge topology, from which we can see that the nodes with high energy consumption across the network are almost all concentrated in the centre, but EABC offloads the content closer to the consumers to improve the cache utilization and balance the load. Therefore, it can be seen that EABC can play a role in balancing the network energy in both edge topology and core topology compared to ABC, which makes the energy consumption of the nodes with high load in the network slower and thus prolongs the network lifetime.

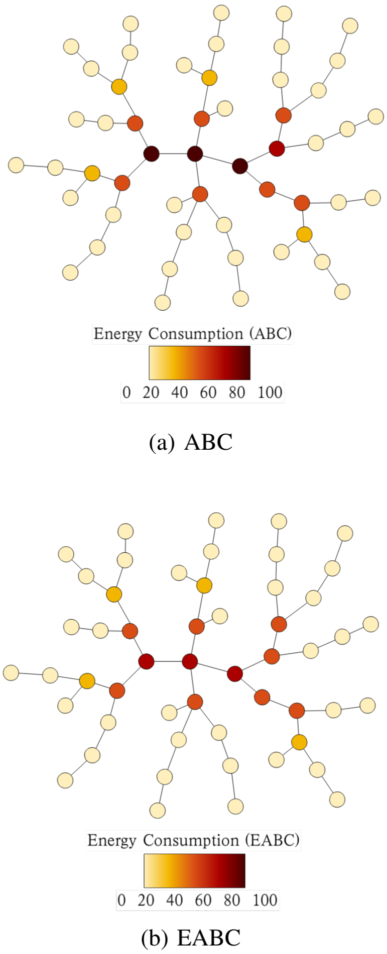


Fig. 10: Distribution of energy hotspots in core topology

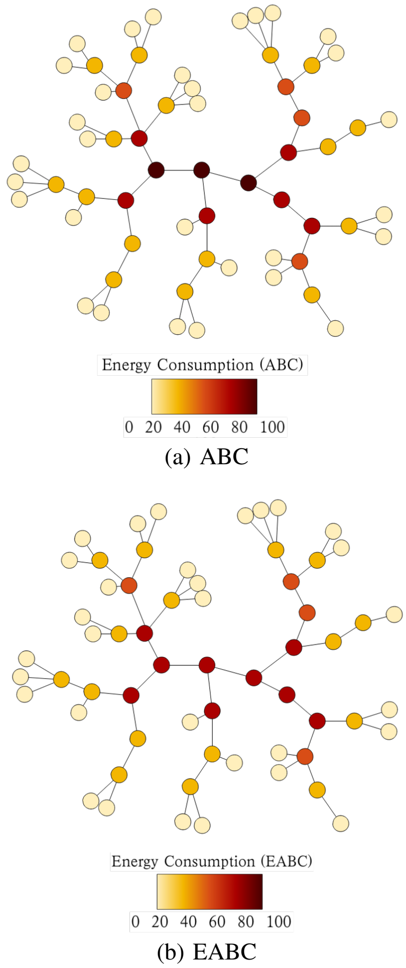


Fig. 11: Distribution of energy hotspots in edge topology

V. Conclusions

This paper investigates caching decision strategies in NDN networks, taking into account the constrained resources in IoT scenarios. To address the uneven energy consumption of high-load nodes in energy-limited networks, an NDN-IoT caching strategy based on node centrality and energy is proposed. During the interest request process, a field is added to the interest packet to record the betweenness centrality of all nodes in the network. By considering the betweenness centrality and the remaining energy of nodes, content is cached in important nodes in the network to balance energy consumption of high-load nodes, reduce content retrieval latency, and improve caching hit rates. Finally, simulation experiments are conducted using core and edge topologies, and the performance is evaluated in terms of caching hit rates, average transmission delay, network lifetime, etc., demonstrating the topology independence of the proposed strategy.

Acknowledgement

This paper was supported by the Natural Science Foundation of Inner Mongolia Autonomous Region 2021MS06003.

References

[1] J. Takemasa, Y. Koizumi, and T. Hasegawa, "Is caching a key to energy reduction of ndn networks?" *IEICE Transactions on Communications*, vol. 99, no. 12, pp. 2489–2497, 2016.

[2] H. Al-Ward, C. K. Tan, and W. H. Lim, "Caching transient data in Information-Centric Internet-of-Things (IC-IoT) networks: A survey," *Journal of Network and Computer Applications*, vol. 206, p. 103491, 2022.

[3] V. Jacobson *et al.*, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2009, pp. 1–12.

[4] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, vol. 63, no. 7, pp. 609–634, 2006.

[5] T. Zhang, X. Xu, L. Zhou, X. Jiang, and J. Loo, "Cache space efficient caching scheme for content-centric mobile ad hoc networks," *IEEE Systems Journal*, vol. 13, no. 1, pp. 530–541, 2018.

[6] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *Proceedings of the International Conference on Recent Advances in Internet of Things (RIoT)*, 2015, pp. 1–6.
[7] K. Hasan and S.-H. Jeong, "Efficient caching for data-driven iot applications and fast content delivery with low latency in icn," *Applied Sciences*, vol. 9, no. 22, p. 4730, 2019.
[8] D. Gupta, S. Rani, and S. H. Ahmed, "ICN-edge caching scheme for handling multimedia big data traffic in smart cities," *Multimedia Tools and Applications*, pp. 1–21, 2022.
[9] M. A. Naeem, T. N. Nguyen, R. Ali, K. Cengiz, Y. Meng, and T. Khurshaid, "Hybrid cache management in iot-based named data networking," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7140–7150, 2021.
[10] M. Amadeo, G. Ruggeri, C. Campolo, A. Molinaro, and G. Mangiullo, "Caching popular and fresh iot contents at the edge via named data networking," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 610–615.
[11] S. Rahel, A. Jamali, and S. El Kafhali, "Energy-efficient on caching in named data networking: A survey," in *Proceedings of the 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*. IEEE, 2017, pp. 1–8.
[12] O. Hahm, E. Baccelli, T. C. Schmidt, M. Wahlisch, and C. Adjih, "A named data network approach to energy efficiency in IoT," in *Proceedings of the IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 4-8 December 2016, pp. 1–6.
[13] O. Hahm, E. Baccelli, T. C. Schmidt, M. Wählisch, C. Adjih, and L. Massoulié, "Low-power internet of things with ndn & cooperative caching," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, 2017, pp. 98–108.
[14] D. Gupta, S. Rani, A. Singh, and J. J. Rodrigues, "ICN based efficient content caching scheme for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
[15] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla, "Energy-aware Cache Placement Scheme for IoT-based ICN Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2021, pp. 1–6.
[16] S. Mori, "Energy-Efficient Cooperative Caching Scheme for Green ICWSN: Preliminary Analysis and Testbed Development," in *Proceedings of the 2nd Workshop on Networked Sensing Systems for a Sustainable Society*, ser. NET4us '23. Madrid, Spain: Association for Computing Machinery, 2023, p. 207–212. [Online]. Available: https://doi.org/10.1145/3615991.3616406
[17] L. Li, Y. Xu, J. Yin, W. Liang, X. Li, W. Chen, and Z. Han, "Deep Reinforcement Learning Approaches for Content Caching in Cache-Enabled D2D Networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 544–557, 2020.
[18] H. Wu, A. Nasehzadeh, and P. Wang, "A deep reinforcement learning-based caching strategy for iot networks with transient data," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13 310–13 319, 2022.
[19] M. Alipio and M. Bures, "Deep Reinforcement Learning Perspectives on Improving Reliable Transmissions in IoT Networks: Problem Formulation, Parameter Choices, Challenges, and Future Directions," *Internet of Things*, vol. 23, p. 100846, 2023.
[20] H. Wang, J. M. Hernandez, and P. Van Mieghem, "Betweenness centrality in a weighted network," *Physical Review E*, vol. 77, no. 4, 2008.
[21] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proceedings of the 11th International Networking Conference (NETWORKING)*, 2012, pp. 27–40.
[22] J. Pfender, A. Valera, and W. K. G. Seah, "Easy as ABC: A lightweight centrality-based caching strategy for information-centric IoT," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, 2019, pp. 100–111.
[23] N. Baltagiannis, I. A. Kapetanidou, and V. Tsaoussidis, "EFPCaching: Energy-aware and Fault-tolerant Probabilistic Caching of popular IoT content in ICN," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Miami, FL, USA, 8-12 May 2023, pp. 1–4.
[24] M. Chowdhury, J. A. Khan, and L. Wang, "Leveraging content connectivity and location awareness for adaptive forwarding in ndn-based mobile ad hoc networks," in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, 2020, pp. 59–69.
[25] J. Li, B. Liu, and H. Wu, "Energy-efficient in-network caching for content-centric networking," *IEEE Communications Letters*, vol. 17, no. 4, pp. 797–800, 2013.