NCP: A Near ICN Cache Placement Scheme for IoT-based Traffic Class

Boubakr Nour*, Kashif Sharif*, Fan Li*, Hassine Moungla‡§, Ahmed E. Kamal,¶, and Hossam Afifi§
*School of Computer Science, Beijing Institute of Technology, Beijing, China
‡LIPADE, University of Paris Descartes, Sorbonne Paris Cite, Paris, France
§UMR 5157, CNRS, Institute Mines Telecom, Telcom SudParis, Nano-Innov CEA Saclay, France
¶Electrical and Computer Engineering, Iowa State University, Ames, USA
Email: {n.boubakr,kashif,fli}@bit.edu.cn, hassine.moungla@parisdescartes.fr,
kamal@iastate.edu, hossam.afifi@telecom-sudparis.eu

***Abstract*—Information-Centric Networking is considered as one of the most promising architecture for IoT. The use of content-centric approach may improve the content access & dissemination, reduce the content retrieval latency, and enhance the network performance. The use of in-network caching in ICN enhances the data availability in the network, overcome the issue of single-point failure, and improve IoT devices power efficiency. In this paper, we present a Near-ICN Cache Placement (NCP) scheme for IoT taking traffic class into consideration. NCP is designed to select the optimal replica cache by minimizing: the cost of moving the data from content producer to replica nodes, cost of caching the content in the replica, and the cost of delivery the content to consumers. Hence, we presented a multi-objective optimization problem, with a heuristic caching selection algorithm. We evaluated NCP with various performance metrics against different caching schemes. The obtained results show improvement in the cache utilization, with fast data retrieval, and enhancement in the network cache distribution & diversity.**

I. Introduction

The current Internet model has been designed to route all requests for the same content toward the original content provider. This model lacks of data dissemination support and fast content retrieval, that by consequence increases the network load, content retrieval delay, and consume more bandwidth. The original content provider is required to be connected all time to fulfill all requests, with a huge insufficiency of content/service availability and an issue of single point of failure. This issues enforce researchers to use in-network caching concept. Hence, Content Delivery Networks (CDN) [1] has been introduced that consists of deploying an overlay web-caching at the application layer on top of the current Internet architecture. However, deploying an addons solution such CDN on top of IP, model that has many other security and mobility patches that make it more complex [2]. Further, there is no standardized protocol for CDN where different companies may develop it based on their demands, regardless of the implementation and deployment cost.

Consequently, the in-network caching has been considered as a fundamental design concept for the future Internet architectures, where various solutions have been proposed. Information-Centric Networking (ICN) [3] is one of the promising paradigms that may replace the current Host-Centric Networks. ICN implements in-network caching in the network layer rather than application layer, in a distributed and standardized manner. The network infrastructure caches the content, and responses for different requests instead of forwarding them to the original content provider, that improves the overall network performance by facilitating the content retrieval, reduce the network delay, and improve the energy consumption.

All these advantages go back to the use of content name instead of IP address. ICN decouples the content from its location, by giving each content an unique, location-independent name. Also, all security mechanisms are applied to the content itself regardless of the communication or the used channel. As the content is self-consistent and independent from its original location. Any ICN node can cache the content and serve it for the future requests.

Furthermore, billions of new devices, mobiles, and smart sensors are connected to the Internet under the concept of Internet of Things (IoT) [4]. These devices can sense, collaborate, and interchange data between them and the Internet. As the complex design of IP model can not handle such interconnection and data exchange, ICN meets the IoT requirements. Thanks to the abstraction of content and its location, heterogeneous devices may connect without the need of middlewares. Also, the simple, and wide space content naming scheme facilitate the content, services, and devices naming with a clean discovery and forwarding design. Moreover, the seamless mobility support, trust models, and distributed caching make ICN an ideal candidate for IoT [5]. In-network caching is extremely required in IoT environment for fast content dissemination with multiple devices in a cost-efficient way. IoT applications may solicitude content based on some properties, such critical/emergency content, monitoring, event traffic or query-based traffic.

The motivation behinds this work is to propose a near cache placement selection for IoT based on traffic class. Benefiting of ICN model, we select the optimal node to cache the content with the minimum cost of moving data from the original content provider to the replica node, the minimum caching cost at the intermediate cache store, and the minimum cost of delivering the content from the replica-node to the content consumer, we took different IoT traffic class (pull, periodic-push, and event-push traffic) into consideration.

To summarize, in this paper we provide the following contributions: modeling multi-objective minimization problem for caching the content more closer to IoT consumer, taking into consideration: cost of moving data from original producer to the cache store, cost of caching the data in the cache, the cost of moving the cached content to consumers, and the IoT traffic class, proposing an heuristic algorithm to select the near cache placement in an optimal manner.

The rest of the paper is organized as follows: Section II highlights ICN-IoT efforts and their caching solutions. Section III presents our proposed solution for ICN-IoT networks. The evaluation performance over large-scale topology, and result discussion are presented in section IV. Finally, we conclude the paper in section V.

II. Related Works

Information-Centric Networking [3] has been proposed as a promising paradigm for the future Internet, to overcome various issues and challenges in the current host-centric model [6]. ICN aims to integrate all network functionalities around the name of content instead of using host addresses. In the following, we review the existing, related works on ICN and IoT, focusing on in-network caching research relevant.

ICN for IoT Research Efforts: In this paper, we recommend leveraging ICN as a forwarding plane for IoT environment. The re-design from connectivity towards content-oriented paradigm puts ICN as one of the best candidates for IoT. It is notable that IoT application pattern follows content-oriented fashion, where sensors and actuators do not need to communicate with a specific *things*, they are more interested in the offered data regardless of its location. Also, the seamless mobility management, in-network caching, and content-based security make ICN more appropriate and suitable for IoT environment.

ICN opens new opportunities to implement a native view of IoT. In such context, various solutions have been proposed. Amadeo et al. [7] focused their efforts in smart home, by proposing an ICN framework based on the use of hierarchical names, support of push and pull traffic, and propose a multi-party forwarding strategy to allow data retrieval from multi-producers. While work in [8] addresses Healthcare applications by proposing a distributed ICN architecture that deals with communication models, publish-subscribe, and mobility issues. Whereas Bouk et al. [9] discussed Intelligent Transportation System from smart cities perspective to provide a secure and a reliable communication by taking ICN features.

ICN In-network Caching: Due to the fact that content names are independent from the original provider location, and each data packet is self-consistent. ICN can provide in-network caching feature [10], with potential that each intermediate node in the communication path can cache the content and serve it for future requests. Hence, the overall network performance will me improved by facilitating content retrieval, and reduce the communication delay. However, deciding what content should be cached and on which device requires ICN to involves different metrics such content popularity and freshness as well as device properties.

Work in [11] focuses on the ubiquitous in-network ICN caching to improve adaptive video streaming, authors suggested the use of bit-rates and content size for best cache utilization, by proposing a rate-selective caching scheme that maximizes the overall throughput and improves QoS. Abani et al. [12] proposed an entropy-based proactive strategy to measure the mobility prediction using Markov-based predictors. The proposed caching strategy fetches the content and caches it in the network, then locates the best node to retrieve the content that may reduce the latency of retrieving predictable content requests, decrease the server load, and cache redundancy, and handle mobility hand-overs. Araldo et al. [13] studied ICN caching placement from ISP perspectives, where they proposed a cost-aware greedy algorithm taking the content placement and its size into consideration, in order to minimize the overall costs or maximize the hit-ratio. Authors defined the ISP's cost of content retrieval by the cost associated to external bandwidth needed to retrieve the requested contents.

ICN in-network Caching for IoT: From the other hand, as IoT devices and traffic have different characteristics in compared to the regular Internet traffic and devices, both of these properties (e.g., content popularity and freshness, nodes energy level and distance from original content producer and data consumers) should be taken into consideration when designing a caching placement scheme.

Work in [14] focuses on the content freshness metric, and proposes a freshness-based caching scheme. This scheme consists of adding *Content Freshness Value* in *Cache Store Table* and checks the requested content from consumer with the freshness value before serving that request. Vural et al. [15] discussed ICN caching from IoT perspective. Because of the nature of IoT devices and data, the caching strategies should not be applied in a similar way to multimedia data. Hence, authors considered different metrics (data property and popularity) to decide if an IoT content should be cached or not. Different metrics are used in the study such as content lifetime, time range of incoming requests, and hop distance to the content source and requesters. A distributed probabilistic caching strategy namely *pCASTING* has been proposed in [16], by considering a multi-hop wireless IoT system, taking the data freshness parameter, node characteristics such as energy level and storage capabilities into consideration to adapt a distributed caching probability without the need for any additional signaling information. *pCASTING* aims to increase the energy usage with low content retrieval delays in compared to other NDN strategies. Seetharam et al. [17] proposed a simple greedy caching algorithm to determine which content should be cached in the network. The caching scheme is based on the content popularity metrics, through calculating the total of incoming requests for content and the relative popularity of each content chunk. Work in [18] analyses a cooperative caching scheme and power-saving in low-power IoT environment, and proposes a *Cooperative Caching Side-Protocol* that aims to maximize sleeping cycles, minimize nodes energy consumption, and increase the content availability.

Broadly, one of the most used caching strategies [19] are: Leave Copy Everywhere (LCE), Leave Copy Down (LCD), Edge Caching (EC), and Consumer Cache (CC). LCE consists of keeping a copy of the content in all the intermediate nodes along the content delivery path. While LCD aims to keep a copy only in the gateway down-stream during the reverse path towards the consumer. Whereas EC caches the content on edge node from the consumer point of view. Similarly, CC keeps a copy of the content one hop after the consumer regardless if its an edge or node.

It is worth-note here to highlight that all the previous works take only one objective in their study, and did not focus on IoT traffic class. Thus, the primary motivation of this work is to design a near cache placement selection scheme for IoT, with multi-objective minimization problem, as well as taking IoT characteristics and traffic classes into consideration.

III. NCP: Near-ICN Cache Placement for IoT

ICN-based caching is highly required for IoT applications to disseminate data in a fast manner, from sensor producers toward edge nodes and consumers in a cost-efficient way. Also, it aims to improve the energy consuming and mobility handover. In the following, we present a *Near-ICN Cache Placement for IoT* scheme.

A. System Model

In this section, we describe the used system model. A comprehensive introduction to the most used notations in the approach can be found in Table I.

TABLE I
Summary of the notation used in this paper.

Parameters of the Models	
N	Set of nodes
Q	$Q \subset N$ Set of requesters
P	$P \subset N$ Set of original providers
L	$L = \{R \cup ET\}$ Set of replica-nodes
$FS(i)$	Set of forward arcs $(i, j) \in A$ for node $i \in N$
$BS(i)$	Set of backward arcs $(i, j) \in A$ for node $i \in N$
D	Set of data
$B_{i,j}$	Link capacity between nodes i and j
S_l	Total cache size of node $l \in L$
x_q^d	Demand for data $d \in D$ from node $q \in Q$
r_p^d	0-1 Data reachability: $r_p^d = 1$ if producer $p \in P$ can serve object $d \in D$
k_l^d	0-1 Cache storage reachability: $k_l^d = 1$ if replica-node $l \in L$ can serve object $d \in D$
$C_{i,j}^d$	Cost of moving $d \in D$ from node i to node j
$\overline{C_l^d}$	Cost of cache $d \in D$ in replica-node $l \in L$
C_l^d	Cost of cache $d \in D$ in replica-node $l \in L$
β_i	Traffic Class (Pull, Periodic-Push, or Event-Push)
Decision Variables of the Models	
$a_{i,j}$	1-0 Node assignment: $a_{i,j} = 1$ if node $i \in L$ is assigned to $j \in ET$
$y_{i,j}^{d,q}$	Flow arc $(i, j) \in A$ for data $d \in D$ requested by $q \in Q$
w_i^d	Flow served for data $d \in D$ by producer or replica-node $i \in \{P \cup L\}$
$z_{i,j}^q$	1-0 Forwarding variable: $z_{i,j}^q = 1$ the arc $(i, j) \in A$ is used to route request $q \in Q$

ICN-IoT network is represented as a graph $G = (N, A)$, where N is a set of nodes contains a collection of access things AT, edge things ET, and intermediate routers R with caching capabilities. We define $Q \subset N$ as a set of requesters asking for data, $P \subset N$ as a set of original data providers, and $L = \{R \cup ET\}$ as a set of nodes who can offer the data. Each node $q \in Q$ generates a traffic demand x_q^d, asking for data $d \in D$, this request might be satisfied by a replica-node $k_l^d = 1$ where $l \in L$, or retrieving the data directly from content provider $r_p^d = 1$ where $p \in P$. Each node $i \in N$ might be assigned at most to one ET.

B. Problem Formulation

We consider the global objective from ICN-IoT perspectives as follows: move the data requested closer to requesters, with 1) the minimum cost from original provider to replica-node, 2) minimum cost of caching the data in the replica-node, and 3) the minimum cost to deliver the data from the replica-node to the requester. Further, each type of IoT traffic should be treated and cached separately than others.

Hence, we divide and formulate our main objective into four sub-problems:

Objective (1) Move the requested data closer to requesters, with the minimum cost from original data provider to the replica-node: The cost of moving the data $C_{p,l}^d$, $d \in D$ from the provider $p \in P$ to the replica-node $l \in L$ is depending on the data itself. The objective can be formulated as follows:

$$\min \sum_{d \in D} \sum_{\substack{(i,j) \in N \\ i \in P, j \in L}} C_{i,j}^d y_{i,j}^{d,q} \tag{1}$$

Objective (2) Minimum cost of caching the data in the replica-node: This objective is defined on each replica-node of the graph as follows: if a data is stored in a replica node, then the associated cost of caching/storage $\overline{C_l^d}$ has to be minimized. This can be formulated as follows:

$$\min \sum_{d \in D} \sum_{\substack{i \in L \\ (j,i) \in BS(i)}} \overline{C_i^d} y_{j,i}^{d,q} \tag{2}$$

Objective (3) The minimum cost to deliver the data from the replica-node to the requester: This objective is similar to the objective (1), but the traffic is only from the replica-node to the requester rather than the original provider to the replica-node:

$$\min \sum_{d \in D} \sum_{\substack{\forall (i,j) \in N \\ i \in Q, j \in L}} C_{i,j}^d y_{i,j}^{d,q} \tag{3}$$

The Global Objective Function: Minimize the overall caching per class of IoT traffic: This objective (4) aims to select the most prioritized traffic and cache it closer to the requester based on the overall optimization defined in objectives (1), (2), and (3), and β_i is the traffic class (Pull

Traffic, Periodic-Push, and Event-Push) with different weighting parameter (14). The more weighting is the more class to prioritize.

$$\min \sum_{d\in D}\sum_{i=1}^{k}\beta_i\Big(\sum_{(p,l)\in N} C^d_{p,l}y^{d,q}_{p,l} + \sum_{(p,l)\in BS(l)} \overline{C^d_l}y^{d,q}_{p,l} + \sum_{(q,l)\in N} C^d_{q,l}y^{d,q}_{q,l}\Big) \quad \forall (q,l,p)\in N, q\in q, l\in L, p\in P, \tag{4}$$

Subject to:

$$\sum_{(j,r)\in BS(r)} y^{d,q}_{j,r} - \sum_{(r,j)\in FS(r)} y^{d,q}_{r,j} = 0, \quad \forall d\in D, \forall q\in Q, \forall r\in L \tag{5}$$

$$\sum_{(j,i)\in BS(i)} y^{d,q}_{j,i} = x^d_i, \qquad \forall d\in D, \forall q\in Q \tag{6}$$

$$\sum_{q\in Q}\sum_{(p,j)\in FS(p)} y^{d,q}_{p,j} = w^d_p, \qquad \forall d\in D, \forall p\in P \tag{7}$$

$$w^d_p \le \sum_{q\in Q} r^d_p x^d_q, \forall d\in D, \forall p\in P, \forall l\in L \tag{8}$$

$$w^d_l \le \sum_{q\in Q} k^d_l x^d_q, \forall d\in D, \forall p\in P, \forall l\in L \tag{9}$$

$$\sum_{q\in Q} x^d_q = \sum_{i\in\{P\cup L\}} w^d_i, \qquad \forall d\in D, \forall i\in\{P\cup L\} \tag{10}$$

$$\sum_{d\in D}\sum_{q\in Q} y^{d,q}_{i,j} \le B_{i,j}, \qquad \forall (i,j)\in A \tag{11}$$

$$\sum_{d\in D} C^d_l \sum_{q\in Q}\sum_{(i,l)\in BS(l)} y^{dq}_{il} \le S_l, \qquad \forall l\in L \tag{12}$$

$$\sum_{d\in D} y^{d,q}_{j,i} \le B_{i,j} z^q_{i,j}, \forall i\in N\setminus Q, \forall (i,j)\in FS(i), \forall q\in Q \tag{13}$$

$$\sum_{i=1}^{k}\beta_i = 1 \tag{14}$$

$$\sum_{j\in ET} a_{i,j} \le 1, \qquad \forall i\in N\setminus ET \tag{15}$$

$$a_{i,j}\in\{0,1\}, \qquad \forall i\in N\setminus ET, \forall j\in N\setminus ET \tag{16}$$

$$k^d_l\in\{0,1\}, \qquad \forall d\in D, \forall l\in L \tag{17}$$

$$w^d_p \ge 0, \qquad \forall p\in P, \forall d\in D \tag{18}$$

$$r^d_p\in\{0,1\}, \qquad \forall d\in D, \forall p\in P \tag{19}$$

$$y^{d,q}_{i,j} \ge 0, \qquad \forall d\in D, \forall q\in Q, \forall (i,j)\in A, \tag{20}$$

$$z^q_{i,j}\in\{0,1\}, \qquad \forall q\in Q, \forall (i,j)\in A \tag{21}$$

The objective function (4) minimizes the overall cost from original provider and replica-node, by minimizing the cost of caching the data in the replica-node, and minimizing the minimum cost to deliver the data from the replica-node to the requester, taking the traffic class into consideration.

The flow balance at every intermediate node and requester node are imposed by (5) and (6), respectively. The flow balance at producer nodes depends on the requested flow (7) which is regulated by (8), (9), and (10). These constraints consider the fact that only original producers or replica-nodes can serve the requests, and the overall traffic served equals overall demands by consumers.

Link capacity constraints are enforced in (11), where all demand for all data over link have not to exceed the capacity of link. While caching capacity constraints are imposed in (12), the cost of caching data does not have to exceed the caching capacity.

In particular, the constraint (13) makes sure that ICN/NDN routing rule is respected, where data delivery uses the same path of request in reverse. Constraint (14) represents the weighting per each IoT traffic class (Pull, Periodic-Push, or Event-Push). Where the constraint (15) enforces that each node has to be assigned at most one ET.

Finally, non negativity on flow variables and binary condition are imposed in (16)-(21).

Algorithm 1: Highest-First, Farthest-Later Algorithm

```
Input: N: Graph, T Traffic Class, C^d_{i,j}, overline{C^d_l}
Output: ReplicaNodes : List of selected replica nodes
Phase 0: Initialization
SortedIntermediateNodes := {};
ReplicaNodes := {};
NotReplicaNodes := {};
Phase 1: Highest replica nodes
SortedIntermediateNodes := sort intermediate nodes
  based on Traffic Class, Degrees, and Free Cache
  Memory;
Phase 2: Farthest replica nodes
for (node in SortedIntermediateNodes) do
    if (not isAdjacent(node, ReplicaNodes)) then
        ReplicaNodes.append(node);
    else
        NotReplicaNodes.append(node);
    end
end
```

C. Heuristic Scheme: Highest-First, Farthest-Later

In the following, we discuss the proposed algorithm that aims to select the optimal cache placement based on IoT traffic class. Algorithm 1 presents a pseudo-code of the proposed solution, that is divided into two phases:

Phase 1 - Highest-First: In the first phase, we sort the list of candidate intermediate nodes, based on the highest received demands, and the free cache memory (Algorithm 1, Line 4), the sort is done for each IoT traffic class.

Phase 2 - Farthest-Later: The second step aims to place the content cache on the nodes that have the highest demands as well as they are far away (Algorithm 1, Line 6). By this selection, we ensure only the intermediates nodes that receive many demands and no immediate neighbors are selected (Algorithm 1, Line 10).

IV. Performance & Evaluation

This section presents and details the performance evaluation of our proposed caching selection algorithm for Information-Centric IoT networks.

To evaluate our solution, we propose scale-free network topology, shown in Figure 1, which consists of a distributed complex graph along with various network hierarchical layers (core network, distributed, aggregation, and access), and different IoT gateways to collect data from IoT sensors and actuators. Further, we used a scale-free network based on Barabasi-Albert model [20].

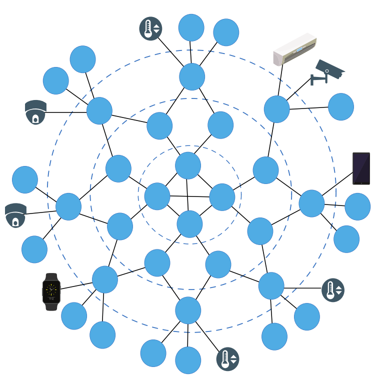

Fig. 1. ICN-IoT Distribution Network

A. Performance Metrics

In the performance evaluation part, we measure network delay, hop reduction ratio, number of selected replica nodes, and cache utilization.

Network Delay: We denote $T_{q,d}$ the time duration to satisfy all demands d for a requester q. In the simulation, we calculate the average network delay τ shown in Eq. 22 for Q requester sending D demands.

$$\tau = \frac{\sum_{q=1}^{Q} \frac{\sum_{d=1}^{D} T_{q,d}}{D}}{Q} \tag{22}$$

Hop Reduction Ratio: It represents the number of hops that can be traversed to fetch the data from the cache store than the original content producer, and represented by Eq. 23.

$$\delta = 1 - \frac{\sum_{q=1}^{Q} \frac{\sum_{d=1}^{D} \frac{h_{q,r}}{h_{q,p}}}{D}}{Q} \tag{23}$$

For each requester q, it sends D demands. For each demands d from requester q, the hop reduction ration is calculated based on the path length $h_{q,r}$ from the requester q and the cache store $r \in R$, where r satisfied the request d, over the path length $h_{q,p}$ from the requester q to the original content producer $p \in P$. In the simulation, the hop reduction ratio is calculated as the average over the Q requesters of average of D demands.

Cache Utilization: We denote $C_{q,d}$ the number of cached packet in the whole network for the demands d issued by the requester q. In the simulation, we calculate the average cache utilization κ shown in Eq. 24 for Q requester sending D demands.

$$\kappa = \frac{\sum_{q=1}^{Q} \frac{\sum_{d=1}^{D} C_{q,d}}{D}}{Q} \tag{24}$$

B. Simulation Results

In the following, we discuss the numerical results obtained by performing an extensive analysis. We benchmarked NCP against different caching placement strategies including LCE, LCD, EC, and CC; assuming that all contents have the same size. Also, as IoT devices generate small value, we do not consider the size in the study. Further, we highlight here that EC and CC strategies both produce the same results due to the fact that all consumers in the generated topologies are one-hop far from the edge node. Hence, the term EC expresses both strategies.

Figures 2, 3, and 4 present the evaluation performance for: network delay, hop reduction ration, and cache utilization respectively.

LCE always follows cache concept by caching data on all nodes, whereas EC selects only consumer edge nodes as replica. Hence, the network delay for EC and LCE is the same and overlapping, due to the fact that the first hop from consumers is the edge which is selected by EC. On the other hand, LCD selects nodes one-hop from the producer, which means that requests need to be forwarded so close to producer, that by consequence produces large network delay. However, NCP selects cache nodes based on demands, the selection is done by the network perspective neither close to producer nor consumers (Objectives (1) and (2)), and memory usage (Objective (3)). Thus, the network delay may be in an average.

In the other side, NCP outperforms the other strategies in terms of hop reduction (Figure 3), through the selection of the optimal near-cache placement, by eliminating the need to forward requests to the original content producer, and allows a fast data retrieval.

Finally, the cache utilization is shown in Figure 4, we can notice that LCE utilizes the whole cache utilization by

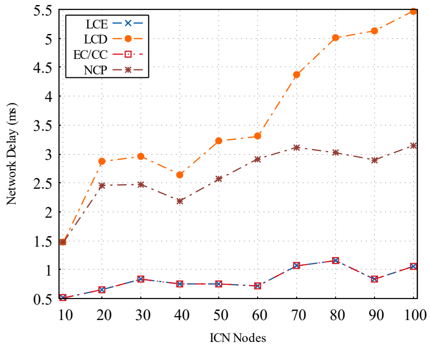

Fig. 2. Network Delay

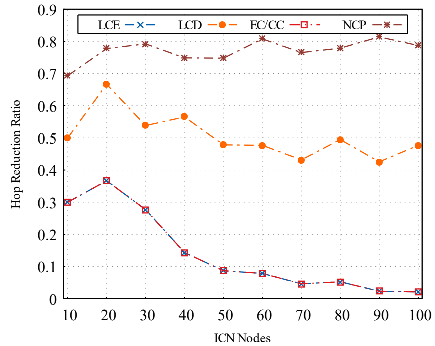

Fig. 3. Hop Reduction Ratio

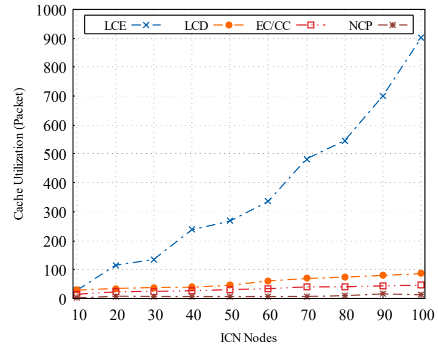

Fig. 4. Cache Utilization

caching replica content in the network level, and decreases the chance for other content which means it decreases the caching distribution and diversity. LCE and EC select small set of cache replica either producer neighbors or consumers' edge respectively that reduces the cache utilization. While NCP selects only the replica with the highest demands/capabilities per class, more free cache space, and that are far away. Hence, it minimizes the whole cache utilization per class and increases the cache distribution in the network.

V. Conclusion

In-network caching is one of the fundamental features of ICN. This work presented a multi-objectives function, and proposed a new caching strategy that aims to minimize the cost of selecting the optimal cache placement in IoT based on traffic classes, taking the cost of data movement from producer to replica nodes, cost of caching in the replica, and the cost of moving the content from replica to consumers into consideration. NCP strategy outperforms other existing strategies in term of cache utilization, and hop reduction by moving the content more closer to consumer regarding the network constraints. Hence, it increases the overall caching distribution and diversity in the network.

References

[1] M. A. Salahuddin, J. Sahoo, R. Glitho, H. Elbiaze, and W. Ajib, "A Survey on Content Placement Algorithms for Cloud-based Content Delivery Networks," *IEEE Access*, 2017.

[2] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," *Technical Report NDN-0038. NDN Project*, 2016.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, " A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[5] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[6] D. P. Arjunwadkar, "Introduction of NDN with comparison to current internet architecture based on TCP/IP," *International Journal of Computer Applications*, vol. 105, no. 5, 2014.

[7] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Information Centric Networking in IoT scenarios: The case of a smart home," in *IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 648–653.

[8] B. Nour, K. Sharif, F. Li, and H. Moungla, "A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study," in *IEEE Global Communications Conference (IEEE GLOBECOM)*. IEEE, 2017.

[9] S. H. Bouk, S. H. Ahmed, D. Kim, and H. Song, "Named-data-networking-based ITS for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 105–111, 2017.

[10] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.

[11] W. Li, S. M. Oteafy, and H. S. Hassanein, "Rate-selective caching for adaptive streaming over information-centric networks," *IEEE Transactions on Computers*, vol. 66, no. 9, pp. 1613–1628, 2017.

[12] N. Abani, T. Braun, and M. Gerla, "Proactive caching with mobility prediction under uncertainty in information-centric networks," in *ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 88–97.

[13] A. Araldo, M. Mangili, F. Martignon, and D. Rossi, "Cost-aware caching: optimizing cache provisioning and object placement in ICN," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2014, pp. 1108–1113.

[14] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *IEEE Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 482–487.

[15] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of internet-of-things data," in *IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3185–3190.

[16] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *International Conference on Recent Advances in Internet of Things (RIoT)*. IEEE, 2015, pp. 1–6.

[17] A. Seetharam and C. Tellambura, "Greedy Caching : A Latency-aware Caching Strategy for Information-centric Networks," in *IFIP Networking Conference*, 2017.

[18] O. Hahm, E. Baccelli, T. Schmidt, M. Wählisch, C. Adjih, and L. Massoulié, "Low-power internet of things with NDN & cooperative caching," in *ACM Conference on Information-Centric Networking (ICN)*, 2017.

[19] I. Abdullahi, S. Arif, and S. Hassan, "Survey on caching approaches in Information Centric Networking," *Journal of Network and Computer Applications*, vol. 56, pp. 48–59, 2015.

[20] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.