

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

Boubakr Nour\*, Kashif Sharif\*, Fan Li\*, Hassine Moun gla<sup>‡§</sup>, Ahmed E. Kamal<sup>¶</sup>, and Hossam Afifi<sup>§</sup>

\*School of Computer Science, Beijing Institute of Technology, Beijing, China

<sup>‡</sup>LIPADE, University of Paris Descartes, Sorbonne Paris Cite, Paris, France

<sup>§</sup>UMR 5157, CNRS, Institute Mines Telecom, Telecom SudParis, Nano-Innov CEA Saclay, France

<sup>¶</sup>Electrical and Computer Engineering, Iowa State University, Ames, USA

Email: {n.boubakr,kashif,flf}@bit.edu.cn, hassine.moun gla@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, overcomes the issue of single-point failure, and improves 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, the 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 data dissemination and fast content retrieval support, that in consequence increases the network load, content retrieval delay, and consumes more bandwidth. The original content provider is expected to be connected all time to satisfy all requests, with a considerable insufficiency of content/service availability and an issue of single point of failure. This issues motivated researchers to use in-network caching concept. Hence, Content Delivery Networks (CDNs) have been introduced, that consist of deploying an overlay web-caching at the application layer on top of the current Internet architecture. However, deploying addons solutions such as CDN on top of IP model, that has many other security and mobility patches, makes it more complex [1]. Besides, 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. Information-Centric Networking (ICN) [2] 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 the 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. Thus, ICN may improve the overall network performance by facilitating the content retrieval, reduce the network delay, and improve the energy consumption.

All of these benefits and improvements due 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. Additionally, all security mechanisms are applied to the content itself regardless of the used communication channel. Finally, 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 demands.

Furthermore, billions of new devices, mobiles, and smart sensors are connected to the Internet under the concept of the Internet of Things (IoT) [3]. These devices are able to sense, collaborate, and interchange data between each other and the Internet [4]. As the complex design of IP model cannot handle such interconnection and data exchange [1], ICN meets the IoT requirements. Thanks to the abstraction of content and its location, heterogeneous devices may connect without the need of middlewares. Additionally, the wide space content naming scheme [5], [6] 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 [7]. In-network caching is hugely 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 as critical/emergency content, monitoring, event traffic, or query-based traffic.

The motivation behinds this work is to propose a near cache placement selection scheme for IoT applications based on traffic class. Benefiting of ICN model, we select the optimal node to cache the content. We propose a multi-objective minimization problem aiming to minimize the cost of moving data from the original content provider to the replica node, minimize the caching cost at the intermediate cache store, and minimize the cost of delivering the content from the replica-

node to the content consumers, taking different IoT traffic class (pull, periodic-push, and event-push traffic) into consideration.

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 [2] has been proposed as a promising paradigm for the future Internet, that aims to overcome various issues and challenges in the current host-centric model. ICN intends to integrate all network functionalities around the content name 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 the IoT environment. The re-design from connectivity towards content-oriented paradigm makes ICN as one of the best candidates for IoT. It is notable that the 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. Furthermore, 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 a context, various solutions have been proposed. Amadeo *et al.* [8] focus 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 [9] addresses healthcare applications, and proposes a distributed ICN architecture that deals with communication models, publish-subscribe, and mobility issues. Whereas Bouk *et al.* [10] discuss Intelligent Transportation System from smart cities perspective to provide a secure and a reliable communication on top ICN model.

*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 [11], with the possibility that each intermediate node, in the communication path, can cache the content and serve it for future requests. Therefore, the overall network performance will be improved by facilitating content retrieval, and reduce the communication delay. However, deciding what content should be cached and on which device require ICN to involves different metrics such as content popularity, freshness, as well as device properties.

Work in [12] focuses on ubiquitous in-network ICN caching to improve adaptive video streaming. The authors suggest the use of bit-rates and content size for best cache utilization; they propose a rate-selective caching scheme that maximizes the overall throughput, and improves QoS. Abani *et al.* [13]

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.* [14] study ICN caching placement from ISP perspective. They propose a cost-aware greedy algorithm to minimize the overall costs or maximize the hit-ratio, taking the content placement and its size into consideration. The authors define the ISP's cost of content retrieval by the cost associated with the external bandwidth needed to retrieve the requested contents.

*ICN In-network Caching for IoT:* On the other hand, IoT devices and traffic have different characteristics in compared to the regular Internet, 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.

Quevedo *et al.* [15] focus on the content freshness metric, and propose a freshness-based caching scheme. This scheme consists of adding *Content Freshness Value* in *Cache Store Table* and keeps checking the requested content from a consumer with the freshness value before serving that request. Vural *et al.* [16] discuss ICN caching from IoT perspective, and recommend that due to the nature of IoT devices and data, the caching strategies should not be applied in a similar way to multimedia data. Hence, the authors consider different metrics (e.g., 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 from the content source to requesters. A distributed probabilistic caching strategy namely *pCASTING* has been proposed in [17], by considering a multi-hop wireless IoT system. *pCASTING* takes the data freshness parameter, node characteristics (i.e., energy level and storage capabilities) into consideration adopting a distributed caching probability without the need for any additional signaling information. The strategy aims to increase the energy usage with low content retrieval delays compared to other NDN caching schemes. Seetharam *et al.* [18] propose 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, by calculating the total of incoming requests for the content and the relative popularity of each content chunk. Hahm *et al.* [19] analyze the cooperative caching scheme and power-saving in low-power IoT environment, and propose a *Cooperative Caching Side-Protocol* that aims to maximize the sleeping cycles, minimize the nodes energy consumption, and increase the content availability.

Although the schemes mentioned above, the most used caching strategies [11] 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 downstream during the reverse path towards the consumer. Whereas EC caches the content on the 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. Finally, 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

Finally, it is worth-note to highlight that all the aforementioned 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 model a multi-objective minimization problem, and design a near cache placement selection scheme for IoT.

#### 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.

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 as follows: move the requested data close to the consumers with: (1) the minimum delivery cost from the original provider to the replica-node, (2) the 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 consumers. Further, each type of IoT traffic should be treated and cached separately than others, in which quality of service policies can be applied. Hence, we divide and formulate our main objective into four sub-problems:

**Objective (1):** *The minimum delivery 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:

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

**Objective (2):** *The minimum cost of caching the data in the replica-node:* this objective is defined on each replica-node as follows: if a data is stored in a replica node, then the associated

TABLE I  
SUMMARY OF THE MOST USED NOTATION IN THIS PAPER.

<i>Parameters of the Models</i>	
$N$	Set of nodes
$Q$	$Q \subset N$ Set of requesters
$P$	$P \subset N$ Set of original providers
$R$	Set of routers with caching capabilities
$AT$	Set of Access Things
$ET$	Set of Edge Things
$L$	$L = \{R \cup ET\}$ 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 $l \in L$
$\beta_i$	Traffic Class (Pull, Periodic-Push, or Event-Push)
<i>Decision Variables of the Models</i>	
$a_{i,j}$	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 $q \in Q$

cost of caching/storage  $\overline{C}_l^d$  has to be minimized. This can be formulated as follows:

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

**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 flow is only from the replica-node to the requester rather than the original provider.

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

**The Global Objective Function:** *Minimize the overall delivery and caching per IoT traffic class:* this objective (4) aims to select the most prioritized traffic and cache it close to the requesters based on the overall optimization defined in objectives (1), (2), and (3), and  $\beta_i$  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.

$$\min \sum_{d \in D} \sum_{i=1}^k \beta_i \left( \sum_{(p,l) \in N} C_{p,l}^d y_{p,l}^{d,q} + \sum_{(p,l) \in BS(l)} \overline{C}_l^d y_{p,l}^{d,q} + \sum_{(q,l) \in N} C_{q,l}^d y_{q,l}^{d,q} \right) \quad \forall (q, l, p) \in N, q \in Q, l \in L, p \in P, \quad (4)$$

Subject to:

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

$$\sum_{(j,i) \in BS(i)} y_{j,i}^{d,q} = x_i^d, \quad \forall d \in D, \forall q \in Q \quad (6)$$

$$\sum_{q \in Q} \sum_{(p,j) \in FS(p)} y_{p,j}^{d,q} = w_p^d, \quad \forall d \in D, \forall p \in P \quad (7)$$

$$w_p^d \leq \sum_{q \in Q} r_p^d x_q^d, \quad \forall d \in D, \forall p \in P, \forall l \in L \quad (8)$$

$$w_l^d \leq \sum_{q \in Q} k_l^d x_q^d, \quad \forall d \in D, \forall p \in P, \forall l \in L \quad (9)$$

$$\sum_{q \in Q} x_q^d = \sum_{i \in \{P \cup L\}} w_i^d, \quad \forall d \in D, \forall i \in \{P \cup L\} \quad (10)$$

$$\sum_{d \in D} \sum_{q \in Q} y_{i,j}^{d,q} \leq B_{i,j}, \quad \forall (i,j) \in A \quad (11)$$

$$\sum_{d \in D} C_l^d \sum_{q \in Q} \sum_{(i,l) \in BS(l)} y_{il}^{dq} \leq S_l, \quad \forall l \in L \quad (12)$$

$$\sum_{d \in D} y_{j,i}^{d,q} \leq B_{i,j} z_{i,j}^q, \quad \forall i \in N \setminus Q, \forall (i,j) \in FS(i), \forall q \in Q \quad (13)$$

$$\sum_{i=1}^k \beta_i = 1 \quad (14)$$

$$\sum_{j \in ET} a_{i,j} \leq 1, \quad \forall i \in N \setminus ET \quad (15)$$

$$a_{i,j} \in \{0, 1\}, \quad \forall i \in N \setminus ET, \forall j \in N \setminus ET \quad (16)$$

$$k_l^d \in \{0, 1\}, \quad \forall d \in D, \forall l \in L \quad (17)$$

$$w_p^d \geq 0, \quad \forall p \in P, \forall d \in D \quad (18)$$

$$r_p^d \in \{0, 1\}, \quad \forall d \in D, \forall p \in P \quad (19)$$

$$y_{i,j}^{d,q} \geq 0, \quad \forall d \in D, \forall q \in Q, \forall (i,j) \in A, \quad (20)$$

$$z_{i,j}^q \in \{0, 1\}, \quad \forall q \in Q, \forall (i,j) \in A \quad (21)$$

The objective function (4) minimizes the overall delivery and caching cost from the original provider and replica-node, by minimizing the cost of caching the data in the replica-node, and minimizing the data delivery cost from the provider to replica-node to the requesters, 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 constraint is enforced in (11), where all demands for all data over link have not to exceed the capacity of link. While caching capacity constraint is 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).

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

In the following, we present and 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 scheme. The proposed algorithm is divided into two phases, and can be executed at the *ET* of an IoT environment.

---

#### Algorithm 1: Highest-First, Farthest-Later Algorithm

---

**Input:**  $N$ : Graph,  $T$  Traffic Class,  $C_{i,j}^d, \overline{C}_l^d$

**Output:** ReplicaNodes : List of selected replica nodes

*Phase 0: Initialization*

- 1 SortedIntermediateNodes := {};
- 2 ReplicaNodes := {};
- 3 NotReplicaNodes := {};

*Phase 1: Highest replica nodes*

- 4 SortedIntermediateNodes := sort intermediate nodes based on *Traffic Class, Degrees, and Free Cache Memory*;

*Phase 2: Farthest replica nodes*

- 5 **for** (node in SortedIntermediateNodes) **do**
  - 6     **if** (not isAdjacent(node, ReplicaNodes)) **then**
  - 7         ReplicaNodes.append(node);
  - 8     **else**
  - 9         NotReplicaNodes.append(node);
  - 10    **end**
  - 11 **end**
- 

*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 describes the performance evaluation of our proposed caching selection algorithm for Information-Centric

IoT networks. To evaluate our solution, we propose a scale-free network topology based on Barabasi-Albert model [20], as 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.

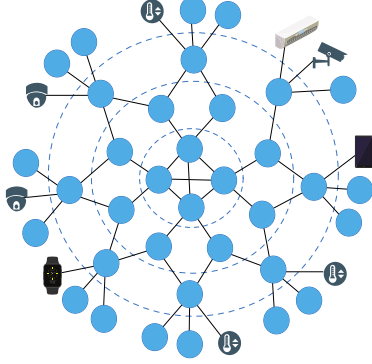


Fig. 1. ICN-IoT Distribution Network

#### 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  $\tau$  shown in Eq. 22 for  $Q$  requester sending  $D$  demands.

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

**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.

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

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  $\kappa$  shown in Eq. 24 for  $Q$  requester sending  $D$  demands.

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

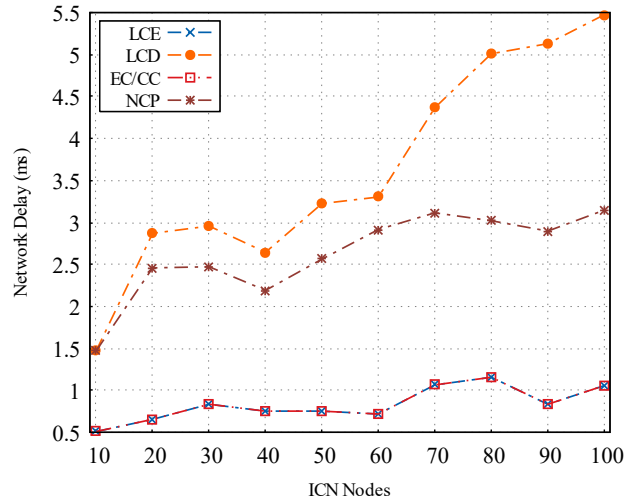


Fig. 2. Network Delay

#### 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 both EC and CC strategies produce the same results since 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 ratio, and cache utilization respectively.

LCE always follows cache concept by caching data on all nodes, whereas EC selects only consumer edge nodes as a replica. Hence, the network delay for EC and LCE is the same

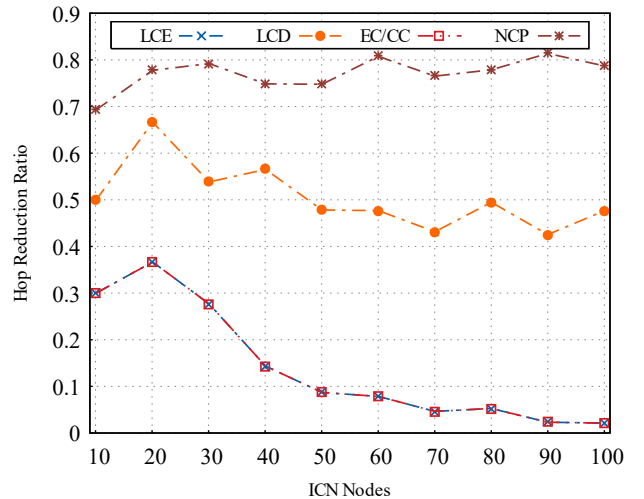


Fig. 3. Hop Reduction Ratio

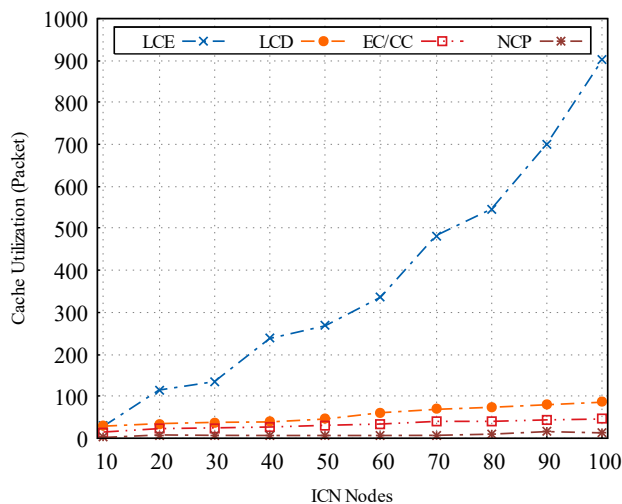


Fig. 4. Cache Utilization

and overlapping, since 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 the producer, that by consequence produces considerable network delay. However, NCP selects cache nodes based on demands, the selection is made 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 total cache memory by 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-objective 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 the consumer regarding the network constraints. Hence, it increases the overall caching distribution and diversity in the network.

## ACKNOWLEDGMENTS

The work of F. Li was supported by the National Natural Science Foundation of China (NSFC) under Grant 61772077, and Grant 61370192. Dr. Li is corresponding author.

## REFERENCES

- [1] W. Shang, Y. Yu, R. Droms, and L. Zhang, "Challenges in IoT networking via TCP/IP architecture," *Tech. Rep. NDN-0038*, 2016.
- [2] A. Seetharam, "On Caching and Routing in Information-Centric Networks," *IEEE Communications Magazine*, no. 3, pp. 204–209, 2018.
- [3] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, 2017.
- [4] H. Mounsla, A. Jarray, A. Karmouch, and A. Mehaoua, "Cost-effective reliability and energy-based intra-WBAN interference mitigation," in *IEEE GLOBECOM Conference*, 2014, pp. 2399–2404.
- [5] B. Nour, K. Sharif, F. Li, H. Mounsla, and Y. Liu, "M2HAV: A Standardized ICN Naming Scheme for Wireless Devices in Internet of Things," in *WASA Conference*, 2017, pp. 289–301.
- [6] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam, "Hierarchical and Flat-Based Hybrid Naming Scheme in Content-Centric Networks of Things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1070–1080, 2018.
- [7] 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.
- [8] M. Amadeo, C. Campolo, J. Quevedo, A. Iera, and A. Molinaro, "Information Centric Networking in IoT scenarios: The case of a smart home," in *IEEE ICC Conference*, 2015, pp. 648–653.
- [9] B. Nour, K. Sharif, F. Li, and H. Mounsla, "A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study," in *IEEE GLOBECOM Conference*, 2017.
- [10] 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.
- [11] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2018.
- [12] 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.
- [13] N. Abani, T. Braun, and M. Gerla, "Proactive caching with mobility prediction under uncertainty in information-centric networks," in *ACM ICN Conference*, 2017, pp. 88–97.
- [14] A. Araldo, M. Mangili, F. Martignon, and D. Rossi, "Cost-aware caching: optimizing cache provisioning and object placement in ICN," in *IEEE Globecom Conference*, 2014, pp. 1108–1113.
- [15] J. Quevedo, D. Corujo, and R. Aguiar, "Consumer driven information freshness approach for content centric networking," in *IEEE INFOCOM WKSPS*, 2014, pp. 482–487.
- [16] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of internet-of-things data," in *IEEE ICC Conference*, 2014, pp. 3185–3190.
- [17] M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in *IEEE RIoT Conference*, 2015, pp. 1–6.
- [18] A. Seetharam and C. Tellambura, "Greedy Caching : A Latency-aware Caching Strategy for Information-centric Networks," in *IFIP Networking Conference*, 2017.
- [19] O. Hahm, E. Baccelli, T. Schmidt, M. Wählisch, C. Adjih, and L. Massoulié, "Low-power Internet of Things with NDN & cooperative caching," in *ACM ICN Conference*, 2017.
- [20] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.