International Journal of Innovative Computing, Information and Control
Volume **15**, Number **5**, October **2019**

ICIC International ©2019 ISSN 1349-4198
pp. 1865–1880

A DETERMINISTIC LATENCY NAME RESOLUTION FRAMEWORK USING NETWORK PARTITIONING FOR 5G-ICN INTEGRATION

Yi Liao[1,2,*], Yiqiang Sheng[1,2,*] and Jinlin Wang[1,2]

[1]National Network New Media Engineering Research Center
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
[*]Corresponding authors: { liaoy; shengyq }@dsp.ac.cn

[2]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

Received January 2019; revised May 2019

Abstract. *Achieving low latency is one of the goals of 5G networks and beyond. We consider Information-Centric Networking (ICN) as a candidate network architecture to realize 5G objectives. The trend of the future network is transformed from best-effort to deterministic transmission. In this paper, we introduce the concept of deterministic latency into the Name Resolution System (NRS) of ICN and propose a Deterministic Latency Name Resolution (DLNR) framework to provide name resolution service with deterministic low latency to satisfy the delay sensitive requirement of NRS. As one of the key techniques, firstly, a Latency-aware Hierarchical elastic area Partitioning (LHP) algorithm is designed to solve the problem of resolver placement with bounded and deterministic transmission latency conditions. Secondly, we design corresponding latency demand-aware name registration and resolution schemes so that constant forwarding hops can be realized. Simulation results show that deterministic low latency name resolution service is achieved by Enhanced NRS (ENRS), which is a system developed to realize the framework of DLNR. Furthermore, the average response latency of ENRS outperforms the Resolution Handler (RH) of DONA 25.2% at best. The query overhead of ENRS is stable due to the deterministic request-forwarding, and the single query overhead of ENRS is reduced by up to 30 times compared with KNN-DNRS.*
Keywords: 5G, ICN, Deterministic low latency, Name resolution, Network partitioning

1. **Introduction.** The missions of 5G put forward harsh demands on the performance of networks. Different application scenarios of 5G have different requirements on the end-to-end latency. For example, the end-to-end latency requirement of factory automation is between 0.25ms to 10ms; the end-to-end latency requirement of AR/VR is less than 25ms and the latency requirement of automatic driving is from 10ms to 100ms [1]. However, current existing IP-based network architectures have some limitations that would not be capable of meeting the requirements of 5G since current IP-based network architectures have a) weak mobility support, b) lacking uniform network layer identifier and c) difficulty adapting to the highly dynamic environments.

Information-Centric Networking (ICN) is an emerging network architecture for the future network, which is support for the identifier (ID) and locator separation. ICN shifts the communication model from a host-centric paradigm that focuses on interconnection to an information-centric paradigm that focuses on information [2].

DOI: 10.24507/ijicic.15.05.1865

ICN splits the ID and locator into different naming spaces, contents are identified by location-independent IDs instead of locators; thus, an infrastructure which maps and stores the mappings of IDs and locators is needed, which is named as Name Resolution System (NRS). In existing ICN architectures, there are mainly two kinds of name resolution approaches, Name-Based Routing approach (NBR) and a Standalone Name Resolution approach (SNR) [3]. NBR approach usually uses hierarchical and aggregative names, and the name resolution process is coupled with message routing, such as NDN [4], and CCN [5]. While in SNR approach the two functions are decoupled, and it usually uses flat names to look up content's locators (e.g., the IP), and then the content is routed by the locators, for example, DONA [6], PURSUIT [7], SAIL [8], and MobilityFirst [9]. NRS is an essential component of the ICN infrastructure. The delivery of data or content can be realized only when the name resolution process is completed.

In this paper, we divide the design requirements of NRS in 5G-ICN integrated networks into two aspects: 1) name resolution accuracy and 2) delay sensitive. Name resolution accuracy requires the NRS to provide accurate and up-to-date name information. When considering the delay sensitive requirement, the name resolution process provided by the NRS must be completed within a minimum delay. For some time-sensitive applications in the 5G network, less than 1ms end-to-end latency is needed, for example, industry control, remoting health-care and other reliable communication with ultra-low latency. If the name resolution process takes too long, then the content request packet may get dropped because of the timeout, or it will yield high content retrieval time for the content requestor. Therefore, the delay sensitive requirement of NRS should be guaranteed priority for the applications which are sensitive to latency in 5G-ICN networks.

The trend of the future network is transformed from best-effort to deterministic data transmission. In industrial environments or some other vertical fields (e.g., vehicle networking), the application services demand deterministic transmission under complicated conditions. The concept of deterministic latency is derived from Time-Sensitive Network (TSN) [10]. It means that the end-to-end latency is bounded or constant and the jitter is controllable. In recent years, Deterministic Network (DetNet) has been raised by IETF. The data paths defined by DetNet standards are capable of supporting bounded characteristics for the paths, such as bounded end-to-end latency, packet loss, packet delay variation (i.e., jitter), and high reliability [11]. In this paper, we are trying to introduce the concept of deterministic latency into NRS and motivate the deterministic low-latency name resolution service to satisfy the delay-sensitive requirement of NRS in 5G-ICN networks. To achieve this goal, we propose a Deterministic Latency Name Resolution (DLNR) framework using network partitioning for 5G-ICN integration. SNR approach is employed in DLNR because of the feasibility of deployment and compatibility with current IP devices. The basic ideas of achieving DLNR can be divided into two steps: a) guaranteeing the maximum deterministic transmission delay for each forwarding hop; b) making deterministic forwarding paths of registration and query requests according to different response time demands. The main contributions of this paper are as follows.

1) We propose a DLNR framework to provide deterministic low-latency name resolution service to satisfy the delay sensitive requirement of NRS by introducing deterministic latency into the NRS.

2) We design a Latency-aware Hierarchical elastic area Partitioning (LHP) algorithm, which is a network partitioning based heuristic algorithm to solve the resolver placement problem. In each segment, a suitable place is selected to deploy one or more resolvers to guarantee the bounded and deterministic maximal transmission latency between the users and its served resolver. Topological information is embedded in the deployment of resolvers.

3) We design corresponding latency demand-aware name registration and name resolution mechanisms to select suitable hierarchical resolver to register or to query based on response time requirements to realize deterministic forwarding of requests.

The remaining sections of this article are as follows. Section 2 provides a brief introduction of ICN name resolution. Section 3 mainly describes the system architecture of our proposed DLNR and its corresponding name registration and resolution schemes in detail. Then in Section 4, we evaluate and analyze the performance of ENRS compared with state-of-the-art NRS. In Section 5 we discuss some challenges the ENRS is facing such as mobility and security and give our considerations on them. We present our conclusion in Section 6.

2. **Related Work.** The current Internet is facing so-called identity-location conflation problem as the IP address plays both the roles of identifier and locator. A number of designs have addressed this problem, and they have tried to decouple the content names from host addresses and proposed ID-locator splitting architecture to translate a self-certifying host identifier to locator, e.g., HIP [12], XIA [13], and LISP [14], which bring great benefit to support mobility and in-network caching.

ICN adopts the ID-locator splitting architecture. In the past decade, several ICN architectures have been proposed for the future networks, e.g., DONA [6], NetInf [15], PSIRP [16], MobilityFirst [9], CONET [17], CONVERGENCE [18], and NDN [4]. The way of name resolution requests routing in the resolvers has a great effect on the performance of NRS, especially the lookup latency. There are three popular classes of NRS based on request routing methods: a) flooding-based approach, e.g., CCN/NDN [4,5]; b) Distributed Hash Table (DHT)-based approach, e.g., MDHT [2], H-Pastry [19], GNRS [9], and αRoute [20]; c) tree-based approach, e.g., Griffin [21], and Ftree [22].

In CCN and NDN the name resolution and data routing are coupled, and flooding-based methods were employed to propagate the routing information [23]. The flooding-based approach has a prominent shortcoming that it would produce a large scale of messages and the lookup latency is uncertain.

In MDHT [2], H-Pastry [19], etc., hierarchical DHTs were utilized by dividing the network topology into hierarchical resolution domains. However, the DHT-based name resolution approaches are overlay designs which need overlay-to-underlay mapping schemes, the routing path may mismatch with the underlay routing path, resulting in long resolution path and latency so that the lookup latency may be jitter, and the upper bound of lookup latency cannot be guaranteed.

In [22], a tree-based ICN name resolution system Ftree was proposed. In Ftree, name records were stored only at the leaf nodes so that object requestors can get resolution response nearby by generating a set of keys using multiple hash functions for a given object name. The limitation of tree-based routing scheme is that the lookup latency is deeply influenced by the depth of the tree. In [24] the Federated Extensible Resolution of Names (FERN) was proposed, which organizes nodes into hierarchical name resolution groups to support deterministic request-forwarding. However, FERN had not made constraints on the latency of single forwarding hop. Thus, the lookup latency cannot be guaranteed, either.

Scalability is a notable problem for flat-name based NRS since flat names are hard to aggregate. In [25] a structure of container where contents or information objects reside was employed for scalable routing. They constructed the containers hierarchically instead of constructing a hierarchy in naming. The structure of a container was organized in recursive way so they can utilize it for their better management. A container may include nested containers and may be involved in a larger container. The containers can be extended into realistic networks such as local, regional or provider networks. MDHT [2], H-pastry [19] system also provided nested, hierarchical DHT architectures for scalable distributed name resolution of flat IDs. The hierarchical model can improve the scalability of the system and the nested structure is benefit to system management. However, seldom of them describe the methods of hierarchical and nested service area partitioning.

Network partitioning is an effective way to solve the problem of area partitioning. Network partitioning is also described as graph partitioning. Graph partitioning is to divide a graph into two or more parts based on certain condition. Graphs are partitioned based on the number of vertices and edges [26]. Graph partitioning has turned out an NP-complete problem [27], so good heuristic and other approximation algorithms for partitioning graph are needed to find solutions to get optimal partition in reasonable amount of time.

Different graph partitioning methods are designed to achieve different goals. For example, in [28] topology-aware partitioning method was used to solve the problem of static energy management for a supercomputer. In [29] the authors proposed a cooperative game theory based network partitioning method to solve the problem of controller placement in SDN-based large networks. It divided the wide-area network into different domains and deployed a dedicated controller in each of the subnetworks at a location that minimized the average switch to controller latency. In [30] Zone Unit (ZU) deployed based on the 5G wireless multi-zone was used as a Device-to-Device (D2D) proxy caching server that exchanges media data information with nearby D2D UEs to provide seamlessly streaming media data service. Some other heuristic partitioning algorithm [31,32] and multi-level partitioning algorithm [31-33] were proposed to realize balanced partitioning or hierarchical partitioning. Inspired by this, we are considering using network partitioning approach to divide intra-domain into nested hierarchical areas to decide the distribution of resolvers so that users can get service nearby as well as guaranteeing the distinguished maximal latency bounds.

3. Framework.

3.1. Overview of naming and name bindings.

Naming. There are two kinds of names in our NRS, i.e., Entity Identifier (EID) and Network Address (NA). The naming schemes of EID and NA are as follows.

a) EID. We adopt 128 bits length flat and global unique EID which is support for self-certifying as the name used in the network layer. The naming scheme of EID is called P:L naming rule, where P represents principle and L means label [6]. In our system, each entity has an EID, including device, content and service, etc. All the copies of the same content share the same EID.

b) NA. We take NA as the locator of entity. Considering the compatibility with current IP-based networks, we use IP (IPv4 or IPv6) address as the NA. The data's NA is the same as the device's NA where the data resides.

Name Binding. Name binding is the mapping pairs of two names. There are two kinds of name bindings in our system, including a) EID-NA, b) EID-EID. We call the first one direct binding and the remaining one intermediary binding. EID-NA is in a one-to-one or one-to-many mapping relationship, e.g., in a multiple access scenario, the EID of an entity can be bounded with one or multiple NAs. The intermediary binding of EID-EID can be used to provide useful capabilities, such as redirecting traffic towards application servers, e.g., firewalls and accounting servers [2]. The bindings of names are dynamic due to the dynamic of the network.

3.2. **Architectural framework.** Figure 1 demonstrates the framework of DLNR with Global Name Resolution (GNR) in 3GPP's 5G architecture. The DLNR is a locally enhanced name resolution mechanism to provide one-to-many relationship between an identifier and locators with the constraints of scopes or distances to achieve deterministic low-latencies in a limited domain by accelerating the name resolution process. GNR is responsible for inter-domain name resolution which keeps the global information of names to guarantee that any EID registered in GNR is accessible and addressable. GNR is recommended to deploy in the 5G Core (5GC) network to help for name information exchanging between different domains.

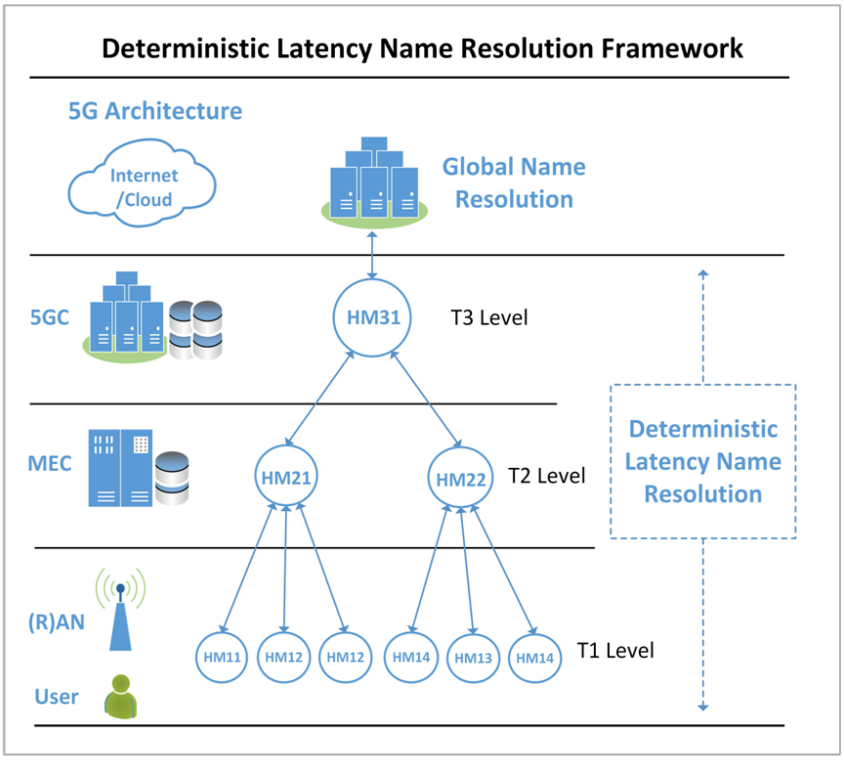

FIGURE 1. The framework of Deterministic Latency Name Resolution (DLNR)

DLNR divides the spatial extent into nested Hierarchical Elastic Area (HEA) based on different deterministic latency constraints, which partition the service areas in finer granularity so that users can get services nearby to realize requests locality. The topology of the nested HEAs is a tree with bidirectional communication. The relationships of parents and children are determined by the hierarchical information. Resolvers in the highest level are the roots of the tree and lowest level resolvers are the leaves. The upper-level HEA is consist of several underlying HEAs. Any two of the HEAs at the same hierarchy are non-overlapping. The HEAs in the same hierarchy are with the same latency bounds and different hierarchies have different latency bounds. The higher the level of HEA, the greater the latency bound.

We call the user node as Local Node (LN), which refers to any network element with an independent address, including end system, edge node, and the intermediate node. i.e., RAN, WiFi-AP, router or User Equipment (UE). LN is a basic unit in the DLNR. An HEA is consist of several LNs, and a dedicated resolver named as HEA Manager (HM) is assigned to each HEA at a location that satisfies constraint condition between the user node to the HM. In Figure 1, we use HMij to represent the HEA it served and show the tree structure of DLNR. Name bindings of EIDs and NAs are stored in HM. Each HM provides query and update functions for LNs in the same HEA. Name registration and query requests will be forwarded to proper hierarchical HM according to different message types and service parameters. The HMs deployed at different levels can be virtualized nodes and they can be deployed on the same physical locations. ICN enables sessionless transport through per-hop name resolution. Our system supports for late-binding technology [34] that any equipment or device could initialize a name resolution request to the NRS in the hop-by-hop forwarding process of packet if the transmission is interrupted.

Except for name bindings, HM also maintains the nested structure of HEA in its administrative domain. There is an HEA Management System (HMS) which is a central management system. The main functions of HMS are 1) collecting and maintaining the global topology information of LNs, 2) implementation of LHP algorithm based on partitioning latency parameters, 3) maintenance and configuration of the nested structure of HEAs in all hierarchies with HEA-related parameters, 4) node management, including new LN join, quit or fail methods to keep the stability of the system.

More details about the key techniques of partitioning LNs into HEAs and the name registration and query strategies are described in Section 3.3 and Section 3.4.

3.3. **Latency-aware hierarchical elastic area partitioning algorithm.** In this section we mainly introduce the LHP algorithm, which is a heuristic graph partitioning algorithm, to solve the issue of HMs' distribution. The design principle of LHP algorithm is to partition the network into nested HEAs and select a location to deploy an HM in each HEA at the position that satisfies the given deterministic one-way latency bound T_i of the i-th level to ensure that the maximal transmission delays from the user nodes to the HM are less than T_i. The range and number of LNs selected in a single i-th level HEA are determined by T_i. The HEA in the same level uses the same T_i as partitioning parameter. The bound latency parameters T_i of the i-th level HEA satisfies $T_i < T_{i+1}$, $1 \le i \le L-1$. L is the highest level of HEA to partition.

The physical network is denoted with the undirected graph $G(V,E,W)$, where $V = \{v_0, v_1, v_2, \dots, v_{N-1}\}$ is the node set, E is the set of edges and W is the set of the edges' weights. In G, if v_p is connected to v_q, then $e_{pq} = 1$, else $e_{pq} = 0$, $0 \le p, q \le N-1$. w_{pq} represents the latency between v_p and v_q. We denote all pairs of shortest path latency matrix with D, where the entry $d(v_a, v_b)$ represents the shortest path latency between v_a and v_b. We assume that $G(V,E,W)$ is divided into M i-th level subgraphs $G_i^m(V_i^m, E_i^m, W_i^m)$, $m = 1, 2, \dots, M$. All the G_i^m are subject to the following:

$$\bigcup_{m=1}^{M} V_i^m = V, \quad \bigcup_{m=1}^{M} E_i^m = E, \quad \bigcup_{m=1}^{M} W_i^m = W \tag{1}$$

$$V_i^m \cap V_i^n = \varnothing, \quad \forall m, n \in \{0, 1, 2, \dots, M\}, \quad m \ne n \tag{2}$$

Equation (1) specifies that all the subnetworks together should cover the entire network. Equation (2) indicates that subnetworks are non-overlapping.

LHP algorithm takes $G(V,E,W)$ as an input set of potential locations for deploying resolvers, as well as latency matrix D, threshold parameter set $\{T_i\}$ and hierarchical parameter L, and returns the HM's location set and HEA node sets as outputs. For a given $G(V,E,W)$ and matrix D, the steps of LHP algorithm to divide the graph into L hierarchies with a set of $\{T_i\}$ by using top-down partitioning method are as follows. The symbols used in the LHP algorithm are shown in Table 1. The pseudocode of the LHP algorithm is illustrated as Algorithm 1.

Table 1. The meaning of symbols

Symbol	Meaning	Symbol	Meaning
i	The hierarchical number of HEA	j	The count of selected resolver location
k	The count of partitioned i-th level HEA	V_i^m	The node set of G_i^m
T_i	The partitioning latency bound of the i-th level HEA	N_i^m	The node count of G_i^m
HM_{ik}	The k-th resolver of the i-th level HEA	N	The node count of G
A_i^j	The candidated resolver location of the j-th HEA in the i-th level	C_i	The count of the i-th level HEA
R_i	The resolver location set of all the i-th level HEA	H_i^k	The set of nodes belonging to the k-th HEA in the i-th level
S_i	The node set of the i-th level input graph	–	–

Step 1: Initialization: $i = L$, $j = 0$, $k = 0$, $S_i = \{v_p\}$.

Step 2: Read $G(V, E, W)$ and get the partitioning latency bound T_i for the i-th level HEAs.

Step 3: Cut the edges in G whose weight is larger than T_i. In this way, G may be divided into one or more connected subgraphs $\{G_i^m(V_i^m, E_i^m, W_i^m)\}$, $1 \leq m \leq M$. Sort $\{G_i^m\}$ by descending order according to N_i^m.

Step 4: For each G_i^m, if $N_i^m = 1$, then set the isolated node as A_i^j of the i-th level HEA, $HM_{ik} = A_i^j$. Then add A_i^j into R_i and H_i^k. $j = j + 1$, $k = k + 1$. End of G_i^m partitioning, go to Step 9. Else go to Step 5.

Step 5: Else if $N_i^m > 1$, firstly, select the v_p from V_i^m whose weighted sum of bandwidth, CPU, storage capacity is the best as A_i^j. Set $HM_{ik} = A_i^j$ and add A_i^j into R_i. Next, go to Step 6.

Step 6: Set A_i^j as root, use Breath-First Search (BFS) algorithm to find the satisfied v_p on the path to A_i^j whose latency is less than T_i and add them into H_i^k. $j = j + 1$, $k = k + 1$.

Step 7: Delete A_i^j and all the selected v_p from V_i^m. If $i - 1 > 0$, then add A_i^j and v_p into S_{i-1}, which is the set of all the nodes to form the $(i-1)$-th level HEA inside the G_i^m.

Step 8: Check whether V_i^m is empty. If $V_i^m = \varnothing$, then go to Step 9. Else go back to Step 5, repeating the process of selecting new anchor nodes and using BFS to find satisfied v_p to form new i-th level HEAs until V_i^m is empty.

Step 9: Check if all of the G_i^m have been partitioned. If satisfied, then go to Step 10. Else return to Step 4;

Step 10: If $i - 1 > 0$, then set $i = i - 1$.

Step 11: Get the latency information of nodes in S_{i-1} from G and D to generate a new graph $G'(V', E', W')$.

Step 12: Then set $G'(V', E', W')$ as the input graph of the $(i-1)$-th level HEA partitioning. That is $G = G'(V', E', W')$. Next return to Step 2 until $i = 1$.

In LHP algorithm, the number of i-th level HEAs and the range of LNs included in a single HEA are determined by T_i and L. HEAs are elastic by adjusting T_i and L according to different deployment requirements and application scenarios. The main difference of LHP algorithm comparing with state-of-art partitioning algorithm is that our partitioning

Algorithm 1. Latency-aware HEA partitioning

Input: $G(V, E, W)$, $\{T_i\}$, L, D

Output: $\{HM_{ik}\}$, $\{H_i^k\}$, $1 \le i \le L$, $1 \le k \le K_i$

Parameters: HM_i^k, A_i^j, $G_i^m(V_i^m, E_i^m, W_i^m)$, V_i^m, N_i^m

1: Initialization: $i = L$, $j = 0$, $k = 0$, $S_i = \{v_p\}$
2: While $(i \ge 1)$ do:
3:     Get T_i;
4:     Delete edges in G where $w_{pq} \ge T_i$ to cut off G;
5:     Assume that we get M counts maximum connected component $\{G_i^m(V_i^m, E_i^m, W_i^m)\}$, $1 \le m \le M$;
6:     Sort all of the $G_i^m\,(V_i^m, E_i^m, W_i^m)$ by descending order according to N_i^m;
7:     for m from 1 to M do:
8:         Get $G_i^m(V_i^m, E_i^m, W_i^m)$;
9:         if $N_i^m = 1$ then:
10:             Set the isolated node as A_i^j of i-th level HEA; $HM_{ik} = A_i^j$
11:             Add the node into H_i^k;
12:             $k = k + 1$;
13:             $j = j + 1$;
14:         end if
15:         else:
16:             Select an anchor node A_i^j from V_i^m whose weighted sum of bandwidth, CPU, storage capacity is the best as HM_{ik}; $HM_{ik} = A_i^j$;
17:             set A_i^j as root, then use BFS to find those v_p on the path to A_i^j whose latency are less than or equal to T_i and add them into H_i^k;
18:             Delete A_i^j and all the selected v_p from V_i^m;
19:             if $i - 1 > 0$:
20:                 Add HM_i^k and selected v_p into S_{i-1};
21:             end if
22:             $k = k + 1$;
23:             $j = j + 1$;
24:             if $V_i^m = \varnothing$ then:
25:                 finish processing G_i^m;
26:             end if
27:             else:
28:                 return to 16;
29:             end if
30:         end if
31:     end for
32:     $C_i = k$
33:     if $i - 1 > 0$:
34:         $i = i - 1$
35:         get the connection information of nodes in S_{i-1} from G to generate a new graph $G'(V', E', W')$;
36:         set G as the input graph of the $(i-1)$-th level HEA partitioning; $G = G'(V', E', W')$;
37:         return to 2;
38:     end if
39: end while

algorithm utilizes deterministic latency as partitioning constraints, while the traditional partitioning algorithms are aimed at minimizing or maximum edge cutting.

3.4. **Name registration and resolution strategies.** In Section 3.3, we have described the partitioning algorithm to decide the distribution of HMs based on transmission latencies. In this section, we mainly describe the strategies to select suitable HM to register or to query by comparing the service-related response time requirement parameter T_d with T_i to make deterministic request-forwarding to satisfy different low-latency requirements of different scenarios. In DNLR, we define three types of massages, i.e., *REGISTER*, *QUERY* and *UPDATE*. T_d, EID or NA are carried in the requests, and T_n is the latency threshold of DLNR. Fine-grained name registration and name resolution requests forwarding schemes are described as follows.

a) Name Register/Update

Case 1: if $0 < T_d/2 < T_2$, the *REGISTER* request will be forwarded to the HM of bottom-level HEA that the requestor belongs to, e.g., HM11. Then HM11 will check whether there exists EID in it. If not, EID will be registered into its BF and the EID-NA binding will be stored as a tuple <EID, NA, T_d, Timestamp>. Each name registration message will be acknowledged with a response message.

Case 2: if $T_i \leq T_d/2 < T_{i+1}$, $2 \leq i \leq L-1$, *REGISTER* request will be forwarded to the HM of the i-th level HEA the requestor belongs to, e.g., HM21. Then the EID will be written into the BF and the binding of EID-NA will be written into the database of HM.

Case 3: if $T_L \leq T_d/2 \leq T_n$, *REGISTER* request will be forwarded to the HM of top-level HEA, e.g., HM31. Then the EID will be written into BF and the EID-NA binding will be written into the DB of HM.

By utilizing the proposed name registration scheme, each HM in different hierarchies stores different name records in a distributed way, thus improving the resource utilization rate of HM and the system is less vulnerable for single point failure. The procedure of updating EID-NA binding is similar to registration. If EID already exists, then the EID-NA binding will be updated if the NA and the timestamp are fresher than the old one.

b) Name Resolution

For a given EID with T_d, the workflow of resolving the EID-NA binding in ENRS is illustrated as follows.

Case 1: if $0 < T_d/2 < T_2$, the *QUERY* request will be forwarded to the HM of bottom-level HEA that the requestor belongs to. The HM will first check whether EID exists in the BF. If EID exists, then HM will query its local DB. Then HM will send a response which carried the EID with one or more NAs to the requestor. Otherwise, if EID does not exist in the BF, then a query failure response will be returned to the requestor. In Case 1, query request will not disseminate to HM's neighbors. Though it can improve the hit ratio while the response time may beyond the requirements of ultra-low latency applications.

Case 2: if $T_i \leq T_d/2 < T_{i+1}$, $2 \leq i \leq L-1$. The *QUERY* request will be forwarded to the i-th level HM that the UE belongs to. Then the HM will check its BF and DB, and the processes of query BF and DB are the same as Case 1. However, different from Case 1, whether to forward the query request to the neighbors of HM depends on the distance between HM and its neighbors.

Case 3: if $T_L \leq T_d/2 \leq T_n$, the *QUERY* request will be forwarded to the HM of top-level HEA that the UE belongs to. Then the HM will check its BF and DB, the processes of query BF and DB are the same as Case 2.

The name query schemes mentioned above do a trade-off between resolution accuracy and resolution latency, we can get resolution response in constant hops so that the query response is within expectable time. In addition, if the two *LN*s are in the same HEA, the resolution path can be contained in the same to achieve local resolution and forwarding locality [2]. As a consequence, the inter-domain traffic would be minimized. Clock synchronization is required in *LN*s and *HM*s.

4. **Evaluation.** We developed an Enhanced Name Resolution System (ENRS) to realize the framework of DLNR. We delegated the GNR to an external cloud platform Global Name Resolution Cloud (GNRC). We employed the Resolution Handlers (RHs) of DONA [6] as GNRC deployed in AS level. We evaluated the performance of ENRS compared with GNRC, Random-NRS, Distributed Name and Resolution System (DNRS) [35,36]. Random-NRS randomly selects K resolvers to register or query names, and we set K = 1 in Random-NRS for the fairness of comparison. DNRS adopted the K-Nearest Neighbor (KNN) algorithm by considering the time of publishing the names in the NRS to choose registered and queried resolvers so that IDs are publishing in the nearest NRS. The time of publishing a name in DNRS was calculated by $T_a - T_b$, where T_a represented the time that the request was received by DNRS and T_b represented the time the request was sent.

We generated five different scales of mesh graphs: $\{100, 200, 400, 800, 1600\}$ as the input graphs of LHP algorithm by BA model of BRITE [37] topological generator. ENRS was constructed based on the outputs of LHP algorithm. According to [11], the end-to-end latency of the current Internet was in the magnitude of ten milliseconds. Thus, we set the scope of edge weight of the input graph from 0 to 10ms. Considering the different delay requirement indicators of the three major application scenarios of 5G, as we mentioned in Section 1, for the industrial applications, the end-to-end latencies should be on the order of a few milliseconds [38], for the Tactile Internet [39] the end-to-end latencies should be around 1 millisecond, and for the one-way front-haul in wireless cellular networks is on the order of 100 microseconds. Thus, we classify the end-to-end latency of 5G typical scenarios into three orders of magnitude: 0∼1ms, 1∼10ms, 10∼100ms and set $L = 3$ to be consistent with the three demand ranges. In addition, the end-to-end latency requirement is described as round trip time while the transmission latency is the time cost in one-way. Therefore, the one-way deterministic partitioning latency is set as $T_1 = 0.5$ms, $T_2 = 5$ms, $T_3 = 25$ms, $T_n = 50$ms, which are approximately half of the round-trip time response time requirements. The false rate of BF is related with the length of BF. We set the BF as 1MB to balance the storage cost and the false rate.

The experiment environment has been created in Python 3.7 running on Ubuntu 14.04 LTS with 32GB RAM. The experimental setup and simulation configuration are shown in Table 2.

We conducted the simulation in 15 rounds for each graph with LHP algorithm and chose different numbers of requested nodes to query different scales of EIDs to evaluate the impacts of response time requirement, request location distribution and request number. In each graph, all the nodes were chosen as EID publishers to register for 10000 individual EID and NA bindings. Then different proportion of nodes were chosen as EID requestors with different response time requirements to query different number of EIDs respectively. In the simulation, the requested nodes and requested EIDs were randomly selected, the distributions of T_d for EIDs were uniform in each quantized interval of T_1, T_2, T_3, and thus the lookup requests to be forwarded to each level HM were equiprobable.

KNN-DNRS and Random-NRS were run under the same conditions of resolvers' distribution, the same numbers of query nodes and the same numbers of EIDs to lookup. We got the results and analyzed the performance of ENRS into serval aspects as follows.

TABLE 2. Experimental setup and simulation configuration

Parameter	Description
L	3
Number of nodes ($node\_count$)	100, 200, 400, 800, 1600
BF length	1MB
m degree	2
Latency scope (ms)	$(0, 10]$
Deterministic partitioning latency (ms)	$T_1 = 0.5$, $T_2 = 5$, $T_3 = 25$
Number of EID published	10000
Number of EID queried	1000, 2000, 4000, 8000
Number of publishers	$node\_count$
Number of subscribers	$0.25 * node\_count$, $0.50 * node\_count$, $0.75 * node\_count$

4.1. **Latency characteristics of HEA.** As shown in Table 3 and Figure 2, for the given parameters L and $\{T_i\}$, we calculated the latency bounds of every HEA produced by LHP algorithm to evaluate the performance of HEA's partitioning, including the count of HMs deployed in each hierarchy, the maximal latency and average latency between LNs to their HMs in each level. In Table 3 and Figure 2, L_i refers to the i-th level HEA.

TABLE 3. The latency characteristics of HEAs produced by LHP algorithm

Node size	HM count	Maximal latency (ms)			Average latency (ms)		
		L_1	L_2	L_3	L_1	L_2	L_3
100	9	0.48	4.95	24.81	0.25	2.52	17.61
200	16	0.49	4.92	24.91	0.26	2.50	17.29
400	25	0.49	4.97	24.96	0.25	2.33	17.47
800	36	0.49	4.99	24.98	0.24	2.42	17.53
1600	68	0.49	4.99	24.98	0.26	2.50	17.53

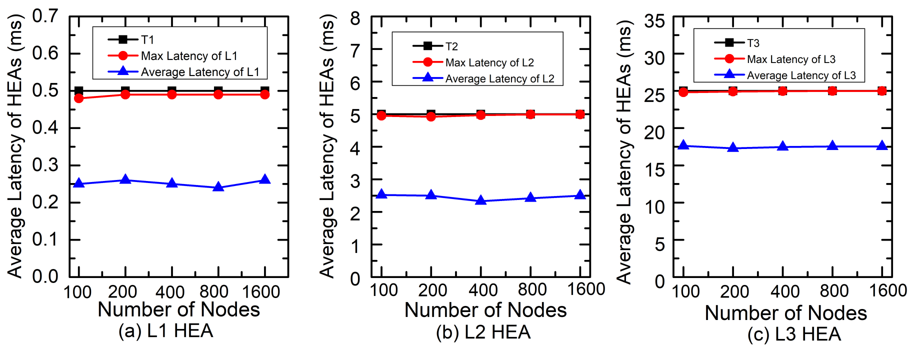

FIGURE 2. The average latency of HEAs in different levels

Table 3 and Figure 2 show that HM's count grows along with the increase of node size. The maximal latencies and the average latencies within different hierarchical HEAs are less than the partitioning threshold T_i for each level HEA, which was consistent with the design principles of LHP algorithm. Thus, we can get a conclusion that the maximal deterministic transmission latency of HEAs in each level can be satisfied by LHP algorithm.

4.2. **Query response latency.** In this section, we mainly analyze the performance of name lookup latency. We use the round-trip time as the response latency. The average latency of single lookup in the local database in our simulation environment was about 7µs, so we mainly conducted the transmission delay as the query latency. The results of ENRS's query response latency compared with T_d, KNN-DNRS, Random-NRS and GNRC are shown in Figures 3 and 4. Figure 3 demonstrates the performance of the average query latency versus the condition of different node sizes. Figure 4 shows the variation of the average query latency with different distributions of lookup queries.

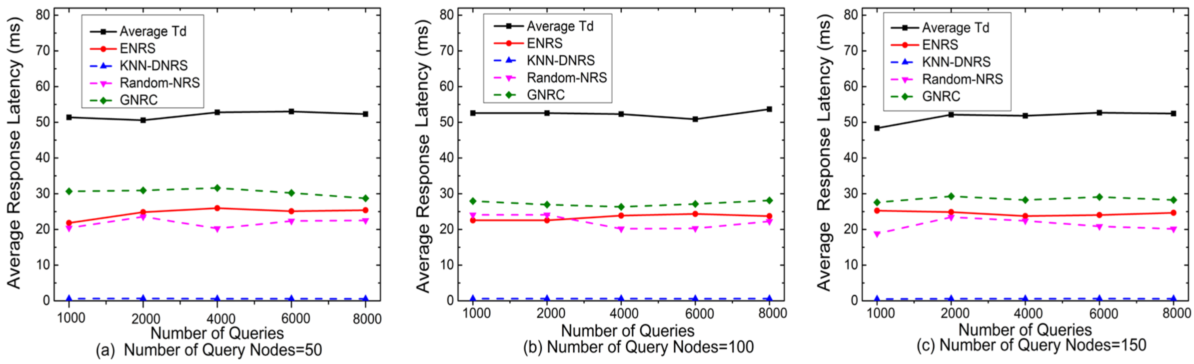

FIGURE 3. Number of queries vs. average response latency (ms)

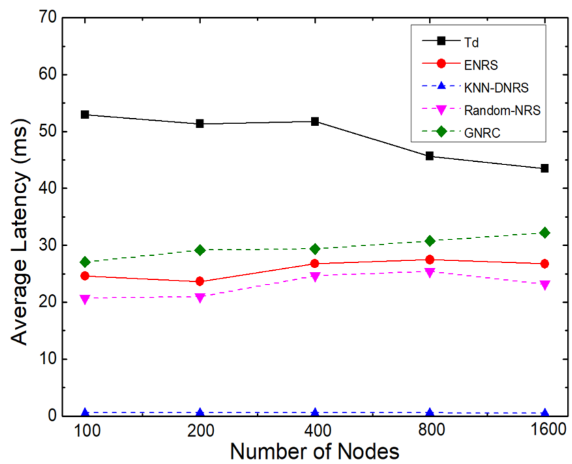

FIGURE 4. Number of nodes vs. average response latency (ms) in each hierarchical HM

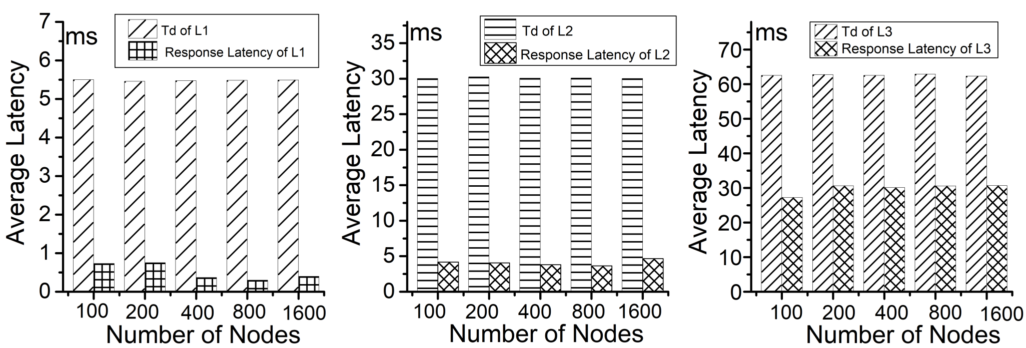

FIGURE 5. Number of nodes vs. average response latency (ms)

From Figures 3 and 4, we can learn that the average response latencies of ENRS are smaller than the average T_d with different numbers of node sizes and different distributions of queries. Further, we analyze the distribution of ENRS's average T_d and average lookup response latency in each level versus different numbers of nodes in Figure 5. From Figure 3, Figure 4, and Figure 5 we can know the query response latencies are smaller than requirements; thus, we can get a conclusion that the different deterministic latency requirements are met in ENRS.

From Figure 4 we can learn that the average response latency of ENRS is smaller than GNRC while larger than KNN-DNRS and Random-NRS with different node sizes. The average response latency of ENRS is reduced by at least 17.2% compared to GNRC, and the ratio can be up to 25.2%. In addition, both of Figure 3 and Figure 4 show that ENRS's response latency is relatively stable compared with Random-NRS even the query conditions changed. Since ENRS is deployed closer to users, the average response latency is smaller than GNRC. ENRS is support for constant hop resolution and deterministic request forwarding based on response time requirement, thus the query latency can be guaranteed and the jitter is small. While in Random-NRS the position of the resolver is randomly chosen, thus the response latency is fluctuant with the variation of query nodes' distributions and queries' distributions, the query latency is indeterminated. KNN-DNRS always chooses the nearest resolvers to register or query; thus, the response latency of KNN-DNRS is the smallest. However, "closest" is also in a state of uncertainty with distance. The response latency would alter when the query positions change. Thus, Random-NRS and KNN-DNRS are not suitable for those applications which require reliable communication such as industry control.

Above all, the simulation results illustrate that differential deterministic low response latency can achieve in ENRS. What is more, the average response latency of ENRS is reduced by 17.2% compared to GNRC, and the proportion of reduction can be up to 25.2%.

4.3. **Overhead analysis.** We are using the traffic of name query as the signaling overhead. The traffic of name resolution can be calculated as *the count of query messages* $*$ *the length of messages*. As shown in Figure 6, the single query traffic overhead of ENRS is stable, and it is the smallest among ENRS, KNN-DNRS and Random-NRS. The single query traffic overhead of ENRS is 4.4 times and 1.3 times smaller than KNN-DNRS and

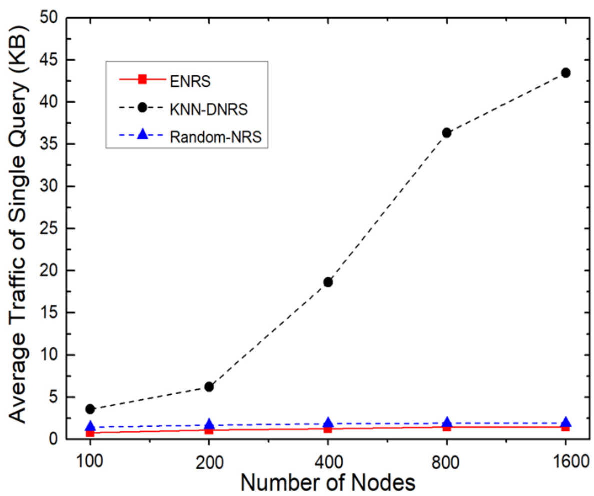

FIGURE 6. Number of nodes vs. average traffic of single query

Random-NRS respectively, and the multiple can be expanded to 30 times and 2 times when the node sizes grow.

In ENRS, we assign constant resolvers for every node to satisfy different response requirements and make constraints on the forwarding hops of queries. Thus, the paths of lookups are deterministic and the resolution hops are constant. Locality resolution and forwarding locality are achieved in ENRS, which makes the lookup traffic overhead relatively stable and controllable. With the same number of resolvers available, KNN-DNRS has to calculate the end-to-end latency with every resolver to select the nearest one to register or query, so the query traffic is increasing sharply with the increase of node sizes and the numbers of query. In Random-NRS, requested nodes randomly select one resolver to register and query. If the resolver does not have the requested name, it will forward the request to its neighbor resolvers by flooding, so the lookup traffic overhead is larger than ENRS and is increasing with the growth of resolvers.

5. **Discussion.** In the 5G-ICN network, the name resolution system will become one of the most important infrastructures. As for ENRS, except for guaranteeing the core functions of name registration and resolution, there still exist some other important aspects we should think about.

Mobility. Mobility has become a basic requirement of communication networks because of the explosive growth of wireless/mobile devices. Mobility objectives can be divided into two types in ICN: producer mobility and consumer mobility. In our system, both the producer and consumer mobility can be primarily handled by the ENRS through late-binding technology [34]. If an entity (e.g., UE, IoT device, and data entity) is moved, it has to be re-registered in the NRS with its new NAs and must be updated in time for mobility support to make itself accessible for other entities. Producer mobility is a big challenge for ICN. A briefly introduction of the Proof-of-Concept proposed by Huawei to provide seamless producer mobility was made in [40] to provide seamless mobility of producer. In our system, the producer mobility issue is also worth to be concerned.

Security. Since ENRS stores all the locators of data objects, the security of ENRS is extremely important. Users who register themselves in ENRS require the authentication to ensure identifying authenticity. Access control policies should be made to name bindings so that the name record would not be revealed to unauthorized users. Some sensitive or private data should not be leaked. In our ENRS, the authentication and accessibility of each name can be ensured by using self-verified names and by constraint of the propagate scope of name binding records.

6. **Conclusion.** In this paper, we propose a Deterministic Latency Name Resolution (DLNR) framework to provide name resolution services with deterministic low latency to satisfy the delay sensitive requirement of NRS in 5G-ICN integration network. To achieve this goal, firstly, we introduce a Latency-aware HEA Partitioning (LHP) algorithm to partition the network into nested areas and deploy a resolver in each area at suitable place to guarantee the maximal latency bound on transmission between users to the HMs in the same HEA. Then we design corresponding name registration and name resolution schemes based on response time requirements to help choosing different serviced hierarchical HMs to make deterministic forwarding of requests.

Simulation results demonstrate that differential deterministic low latencies name resolution requirements are realized under the combination of deterministic transmission latency of HEAs and deterministic request-forwarding. Furthermore, the average response latency of ENRS is reduced by up to 25.2% compared to GNRC. The overhead of a single query of ENRS is stable due to the deterministic forwarding of query request, and the average query overhead of ENRS is extended to 30 times smaller than KNN-DNRS.

Acknowledgments. This work is supported by Special Fund for Strategic Pilot Technology of Chinese Academy of Sciences under Grant. XDC02010701 and Chinese Special Fund of National Major Science and Technology No. 2017ZX03001019. We want to convey our gratitude to Yiran Hao, Jiaqi Li, Jun Yang, Yuanhang Li for giving meaningful support on this research.

REFERENCES

[1] I. Parvez, A. Rahmati, I. Guvenc et al., A survey on low latency towards 5G: RAN, core network and caching solutions, *IEEE Communications Surveys & Tutorials*, vol.20, no.4, 2018.
[2] M. D'Ambrosio, C. Dannewitz, H. Karl et al., MDHT: A hierarchical name resolution service for information-centric networks, *Proc. of the ACM SIGCOMM Workshop on Information-Centric Networking*, Toronto, Canada, pp.7-12, 2011.
[3] L. Dong and G. Wang, A hybrid approach for name resolution and producer selection in information centric network, *Proc. of the 2018 International Conference on Computing, Networking and Communications (ICNC)*, Maui, United States, pp.574-580, 2018.
[4] L. Zhang, A. Afanasyev, J. Burke et al., Named data networking, *ACM SIGCOMM Computer Communication Review*, vol.44, no.3, pp.66-73, 2014.
[5] V. Jacobson, D. K. Smetters, J. D. Thornton et al., Networking named content, *Proc. of the 5th International Conference on Emerging Networking Experiments and Technologies*, Rome, Italy, pp.1-12, 2009.
[6] T. Koponen, M. Chawla, B.-G. Chun et al., A data-oriented (and beyond) network architecture, *Proc. of the ACM SIGCOMM Computer Communication Review*, pp.181-192, 2007.
[7] N. Fotiou, P. Nikander, D. Trossen et al., Developing information networking further: From PSIRP to PURSUIT, *Proc. of the International Conference on Broadband Communications, Networks and Systems*, Athens, Greece, pp.1-13, 2010.
[8] T. Edwall, *Scalable and Adaptive Internet Solutions (Sail)*, 2011.
[9] A. Venkataramani, J. F. Kurose, D. Raychaudhuri et al., MobilityFirst: A mobility-centric and trustworthy Internet architecture, *ACM SIGCOMM Computer Communication Review*, vol.44, no.3, pp.74-80, 2014.
[10] *Time-Sensitive Networking*, http://www.ieee802.org/1/pages/tsn.html.
[11] A. Nasrallah, A. Thyagaturu, Z. Alharbi et al., Ultra-low latency (ULL) networks: A comprehensive survey covering the IEEE TSN standard and related ULL research, *arXiv preprint arXiv: 1803.07673*, 2018.
[12] P. Nikander, J. Ylitalo and J. Wall, Integrating security, mobility and multi-homing in a HIP way, *Proc. of the NDSS*, pp.6-7, 2003.
[13] D. Han, A. Anand, F. Dogar et al., XIA: Efficient support for evolvable internetworking, *Proc. of the 9th USENIX Conference on Networked Systems Design and Implementation*, 2012.
[14] D. Meyer and D. Lewis, *The Locator/ID Separation Protocol (LISP)*, IETF RFC 6830, 2013.
[15] C. Dannewitz, D. Kutscher, B. Ohlman et al., Network of information (NetInf) – An information-centric networking architecture, *Computer Communications*, vol.36, no.7, pp.721-735, 2013.
[16] D. Lagutin, K. Visala and S. Tarkoma, Publish/subscribe for Internet: PSIRP perspective, in *Towards the Future Internet*, G. Tselentis et al. (eds.), 2010.
[17] N. B. Melazzi, M. Cancellieri, A. Detti et al., *The CONET Solution for Information Centric Networking*, Technical Report, 2012.
[18] S. Salsano, M. Cancellieri and A. Detti, *ICTP – Information Centric Transport Protocol for CONET ICN, draft-salsanoictp-00*, Work in Progress, 2013.
[19] N. Fotiou, K. V. Katsaros, G. Xylomenos et al., H-pastry: An inter-domain topology aware overlay for the support of name-resolution services in the future Internet, *Computer Communications*, vol.62, pp.13-22, 2015.
[20] R. Ahmed, M. F. Bari, S. R. Chowdhury et al., αRoute: A name based routing scheme for information centric networks, *Proc. of the INFOCOM*, pp.90-94, 2013.
[21] Y. Sun, Y. Zhang, H. Zhang et al., Geometric routing on flat names for ICN, *Proc. of the 2015 IEEE Global Communications Conference (GLOBECOM)*, San Diego, United States, pp.1-6, 2015.

[22] W. Louati, W. Ben-Ameur and D. Zeghlache, A bottleneck-free tree-based name resolution system for information-centric networking, *Computer Networks*, vol.91, no.341-355, 2015.
[23] A. A. Barakabitze, X. Tan and G. Tan, A survey on naming, name resolution and data routing in information centric networking (ICN), *International Journal of Advanced Research in Computer and Communication Engineering*, vol.3, no.10, pp.8322-8330, 2014.
[24] S. Sevilla, P. Mahadevan and J. J. Garcia-Luna-Aceves, FERN: A unifying framework for name resolution across heterogeneous architectures, *Computer Communications*, vol.56, pp.14-24, 2015.
[25] J. Hong, W. Chun and H. Jung, A flat name based routing scheme for information-centric networking, *Proc. of the 17th International Conference on Advanced Communication Technology (ICACT)*, PyeonhChang, Korea, pp.444-447, 2015.
[26] S. Nage and G. Potdar, A survey on graph partitioning techniques, *International Journal of Science and Research (IJSR)*, vol.4, no.9, pp.1656-1659, 2015.
[27] B. Hendrickson and R. Leland, *A Multi-Level Algorithm for Partitioning Graphs*, 1995.
[28] J. Chen, Y. Tang, Y. Dong et al., Reducing static energy in supercomputer interconnection networks using topology-aware partitioning, *IEEE Trans. Computers*, vol.65, no.8, pp.2588-2602, 2016.
[29] B. P. R. Killi, E. A. Reddy and S. V. Rao, Cooperative game theory based network partitioning for controller placement in SDN, *Proc. of the 10th International Conference on Communication Systems & Networks (COMSNETS)*, Bangalore, India, pp.105-112, 2018.
[30] C. Lee, D2D proxy caching strategy for 5G-based multi-zone networking services, *ICIC Express Letters*, vol.12, no.11, pp.1177-1182, 2018.
[31] N. B. Omeroglu, I. H. Toroslu, S. Gokalp et al., K-partitioning of signed or weighted bipartite graphs, *Proc. of the 2013 International Conference on Social Computing*, pp.815-820, 2013.
[32] H. Zhang, Balanced graph partitioning: Optimizing graph cut based on label swapping, *Proc. of the 2015 International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pp.184-187, 2015.
[33] J. Li and C.-C. Liu, Power system reconfiguration based on multilevel graph partitioning, *Proc. of the 2009 IEEE Bucharest PowerTech*, pp.1-5, 2009.
[34] A. Azgin, R. Ravindran and G. Wang, A scalable mobility-centric architecture for named data networking, *arXiv preprint arXiv: 1406.7049*, 2014.
[35] W. Elbreiki, S. Arlimatti, S. Hassan et al., A distributed name resolution system in information centric networks, *AIP Conference Proceedings*, vol.1761, no.1, 2016.
[36] S. Hassan, W. Elbreiki, S. Arlimatti et al., Named data object organization in distributed name resolution system for information centric network environment, *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol.9, nos.2-5, pp.119-122, 2017.
[37] A. Medina, A. Lakhina, I. Matta et al., *BRITE: Universal Topology Generation from a User's Perspective*, 2001.
[38] M. Wollschlaeger, T. Sauter and J. Jasperneite, The future of industrial communication: Automation networks in the era of the Internet of things and industry 4.0, *IEEE Industrial Electronics Magazine*, vol.11, no.1, pp.17-27, 2017.
[39] G. P. Fettweis, The tactile Internet: Applications and challenges, *IEEE Vehicular Technology Magazine*, vol.9, no.1, pp.64-70, 2014.
[40] Y. Liao, Y. Sheng, J. Wang et al., A brief survey on information centric networking proof of concepts for IMT-2020 and emerging networks, *Journal of Network New Media*, vol.7, no.6, pp.54-63, 2018.