

A Cost Aware Topology Formation Scheme for Latency Sensitive Applications in Edge Infrastructure-as-a-Service Paradigm

Himanshu Gauttam^{a,*}, K.K. Pattanaik^a, Saumya Bhaduria^a, Divya Saxena^b, Sapna^a

^aAtal Bihari Vajpayee - Indian Institute of Information Technology and Management, Gwalior, Madhya Pradesh 474015, India

^bThe Hong Kong Polytechnic University, Hong Kong

Abstract

The network topology formation in an Edge Infrastructure-as-a-Service (ElaaS) paradigm must consider the placement of Edge Computational Nodes (ECNs) so as to minimize the delay. Existing ECN placement schemes consider redundant node density, non-optimal location selection, and distance-based association, which affect the ultra-low latency requirement(s) of applications. Further, per ECN to IoT nodes association is key to efficient utilization of ECNs and delay minimization between IoT node(s) and ECN. This work proposes a *Cost-aware Edge Computational Node Placement (coECNP)* scheme for optimal topology formation in ElaaS paradigm with the objective of IoT nodes delay minimization. It formulates ECN placement problem as a constrained optimization problem. Each iteration in the location discovery module of *coECNP* identifies optimal placement location by utilizing IoT node's density on an updated set of IoT nodes and hop-distance among previous iterations' ECN locations and current candidate locations. As a result, it maximizes the number of IoT nodes that access ECN with minimum hop-distance, leading to end-to-end delay minimization. The assignment module of *coECNP* takes care of previously assigned nodes in each iteration before associating new IoT nodes to the nearest ECN to attain balanced mapping. Thus, it alleviates total delay from IoT node to respective ECN and enhances edge resource utilization to cater the application(s) near real-time execution requirement(s). The performance comparison indicates that *coECNP* achieves promising results by reducing IoT nodes delay by 23-64%, 20-66%, and 35-73% on periodic, event-based, and query-based data traffic scenarios, respectively, under various network settings, compared to the benchmark solutions.

Keywords: IoT, Edge computing, ElaaS paradigm, Topology formation, Edge resource utilization

1. Introduction

The seamless integration of edge computing into various domains such as IoT, industrial IoT (IIoT), Cyber-Physical System (CPS), etc., results in enhanced system performance and reduced operational cost [1] [2] [3] [4]. Edge computing paradigm enables the computation at the network edge and store, process, analyze most of the data generated by IoT nodes close to, or at the edge [5].

Edge computing solutions mitigate the network latency, bandwidth, data storage, security, and other compliance issues associated with cloud infrastructure [4] [5] [6] [7] [8]. The edge varies according to the required goals. Broadly, edge is categorized into three categories named as thick edge, thin edge, and micro edge [9] [10]. The thick edge consists of powerful computing and large storage space (such as data centers, cell tower data centers etc.) responsible for bulk processing (such as factory level data storage and analysis). The thin edge provides the moderate level computation and storage facilities (such as single-board computer(e.g. Raspberry Pi), industrial PC equipped with Intel i-series processor etc.). Micro edge consist of sensors which are responsible for data generation and/or actuation.

With local computation and faster decision-making the edge complements the power of the cloud to build a cost-effective, powerful IoT solution termed as *Edge Infrastructure-as-a-Service (ElaaS)*. An ElaaS setup is a network of computational nodes named as Edge computational Nodes (ECNs) capable of hosting and executing application components. These application components take inputs from various data sources and process them to cater the needs of the system.

The ElaaS paradigm has applicability into a wide range of applications such as intelligent factories, smart plants, prognostic and health management, smart homes, etc. [2] [3] [4]. It enables the application providers to deploy services efficiently and instantly. Most of these applications are latency-sensitive such as monitoring (latency < 1s), supervisory control (latency < 100ms), safety alarming (latency < 1s) etc [11][9]. For instance, shop floor applications in a smart factory are latency-sensitive, which require their treatment at various components along the chain.

The successful completion of these applications requirement(s) depends on various components such as, topological arrangement of IoT nodes, computational hardware, number of ECNs and their placement, association of IoT nodes to relevant ECN (ElaaS topology), etc. Sometimes it includes a higher degree of interconnection, cognitive automation, shifting information collection, and processing into cloud-based applications [12][9]. Specifically, when the application(s) interacts with the

*Corresponding author

Email address: hgauttam262@gmail.com (Himanshu Gauttam)

environment, both end-to-end delay from IoT node to ECN and processing delay at ECN play a paramount role, and it depends heavily on the aforementioned components. Thus, data transmission and decision making process should be optimized as timely as possible to enhance system performance [3].

This work does a systematic review of the related research in a pursuit to investigate the role of ECN placement and IoT nodes mapping among ECNs for topology formation in EIaaS paradigm. Most studies [13] [14] [15] [16] [17] in this direction take a reactive approach and focus on offloading schemes under the assumption of already deployed Edge Servers/Edge Nodes (ESs/ENs) in the network. The question of how to place ESs in a given resource-constrained IoT network to optimize network performance, is less targeted. The ES placement strategies for Mobile Edge Computing (MEC) and Wireless Metropolitan Area Network (WMAN) environment broadly focus on two aspects: communication metric and workload balancing. Along with placement, the number of ESs also impacts the deployment cost and QoS requirements. The number of ESs in the deployment is a trade-off between operator/manufacturer budget constraints and required application performance. The trade-off is optimized using budget or QoS requirement based approaches [18]. The budget based approach places a fixed number of ESs (determined by operators budget constrains) such that delay between node-ES pair is minimized, whereas, QoS based approach determines the minimum number of ESs which satisfies the given QoS parameter (such as maximum tolerable access delay for a node is h -hop). The proposed work assumes a fixed number of ECNs followed by majority of the surveyed studies discussed in [18].

Authors discussed various approaches to minimize the access delay between end devices and ESs/cloudlets [19] [20] [21] [22] [23] [24] [25] [26] [27]. The above mentioned schemes do not address the balanced workload assignment among ESs that impact edge resource utilization. However, balanced workload assignment among ESs with access delay minimization is addressed by few studies [18] [28] [29] [30] [31]. Liang et al. [28] discussed a device remapping based load balancing scheme that results in higher energy consumption of end devices and increased delay. Whereas, historic workload information is required in [18][29]. Moreover, these schemes [18] [28] [29] are restricted to use in direct (one-hop) communication scenario and utilize Euclidean distance to assign end nodes to respective ESs. In contrast, in resource-constrained IoT networks, multi-hop communication is preferred and hop distance is used as a decision parameter in end nodes assignment to minimize delay and energy consumption [32]. Density based placement schemes [31] [33] [34] suffer from issues such as redundancy, non-optimal location selection, and unbalanced node-ECN mapping.

Further, most of these schemes modeled end-to-end delay as total delay and processing delay at ES is ignored. Moreover, MEC solutions focus on performance issues associated with consumer of data, while in above discussed network set-up, performance issues associated with data transmission and end device association with appropriate ES need to be addressed [10]. Further, under the assumption that each IoT node loca-

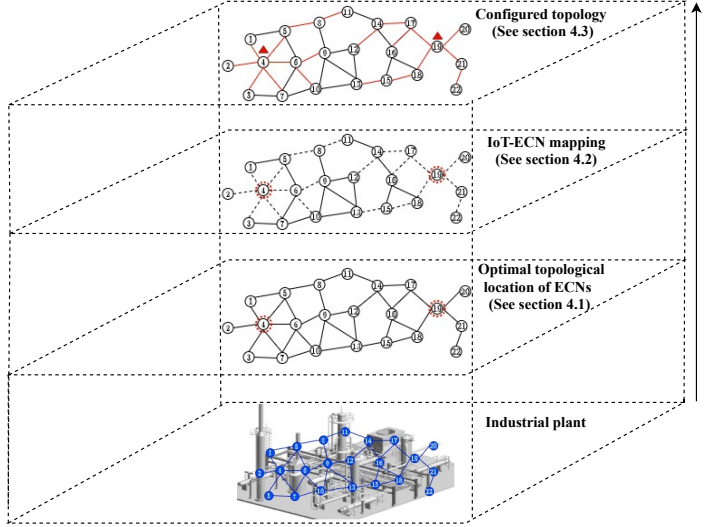


Figure 1: *coECNP* mechanism's levels of abstraction

tion as candidate location for ECN placement and each ECN as candidate for mapping the IoT nodes, the number of possible placement and mapping strategies are combinatorial. Therefore, EIaaS topology formation by selecting an optimal placement and mapping strategy among a large set of choices, which minimize the delay and maximize the efficient utilization of edge resources, is challenging. It initiates the requirement of a heuristic solution to cope with efficient placement and mapping problem in reasonable time.

This paper takes up the ECN placement and IoT nodes to ECN mapping problem for topology formation in EIaaS paradigm and formulates as a constrained optimization problem. We propose a *Cost-aware Edge Computational Node Placement (coECNP)* scheme for topology formation in EIaaS paradigm for delay minimization in order to satisfy near real-time demands of application(s). Fig. 1 depicts the modular functional architecture of *coECNP* scheme. The bottom most layer represents the physical view of an industrial plant equipped with IoT nodes (such as sensors). The remaining layers represent the logical abstraction of modules of *coECNP* scheme.

Key contributions of this work are as follows:

- We propose a *Cost-aware Edge Computational Node Placement (coECNP)* scheme for topology formation in EIaaS paradigm for latency-sensitive applications. *coECNP* scheme is structured around three modules: location identification for ECN placement, mapping of IoT nodes among ECNs, and EIaaS topology configuration.
- To identify optimal locations for ECNs placement, we propose a location discovery mechanism that considers hop-distance and eliminates redundant nodes for density computation in the process of optimal location identification. A mapping scheme is proposed named as *Minimum Set Assignment First* that minimizes the difference of mapped nodes among ECNs. Further, we extend the mapping

Table 1: Summary of ES and cloudlet placement schemes

References	Mechanism	Delay	Balanced assignment	Limitations
[19] [20] [22] [23] [24] [25] [27]	ILP, PSO, Online algorithm, Dominating set and simulated annealing, Benders decomposition, Binary based differential evolution cuckoo search, heuristic approach	Yes	No	Balanced workload/node assignment is not addressed that affects response time and edge resource utilization.
[21] [28] [30]	k -means	Yes, Yes, Yes	No, No, Yes	Euclidean distance is preferred over hop distance in IoT nodes mapping.
[18] [29]	PACK, MIP	Yes, Yes	Yes, Yes	Restricted to use for direct (one-hop) communication scenario. Prior historic workload information is required.
[31]	HAF DBC	Yes Yes	No Yes	HAF scheme did not address redundancy issue. DBC scheme randomly selects optimal location among candidate ones.
[26]	k -clustering	Yes	No	Random selection of initial k cluster centers among candidate ones leads to higher delay and reduced edge resource utilization, redundancy factor is not considered in selection of initial cluster centers.
[35]	Discrete differential evolution algorithm	No	Yes	The response time is modeled as function of computing delay and impact of communication delay on response time is ignored.
[36]	Binary gray wolf genetic scheme	No	Yes	The scheme ignores the delay aspects.
[33]	Density based multi-sink placement	Yes	No	Randomly selects sink placement location among candidate ones and limited to direct communication scenario.
[34]	Density based clustering	Yes	No	Restricted to use in direct communication scenario.

scheme to handle the critical cases in node-ECN mapping.

- The simulation experiments show that *coECNP* scheme outperforms the benchmark solutions in terms of delay and energy consumption of IoT nodes under various network settings and data traffic scenarios, respectively. *coECNP* scheme reduces delay and energy consumption by 23-64%, 20-66%, 35-73%, and 11-58%, 13-58%, 29-55%, respectively, on periodic, event-based, and query-based data traffic scenarios under various network settings.

In the following we discuss related work in section 2 followed by system model and problem formulation in section 3. The *coECNP* scheme for ECN deployment and IoT nodes mapping is discussed in section 4. Section 5 discusses the proposed scheme's theoretical analysis with respect to benchmark schemes. Section 6 presents the experimental results. Section 7 discusses the theoretical analysis of *coECNP* with various network centrality measures followed by conclusion in section 8.

2. Related Work

This section discusses some notable recent work addressing ES placement schemes as Table 1 provides a summary. The problem at hand shares some similarities with ES placement strategies in MEC, WMAN, and cloudlet placement schemes in WMAN.

Ilias Gravalos et al. [19] proposed a linear programming based gateway placement mechanism in IoT network to minimize gateway access cost from end devices while taking care of the essential QoS requirements. Yuanzhe Li et al. [20] proposed a solution for energy-aware ES placement in MEC environment. Authors adopted particle swarm optimization (PSO) technique to determine the placement locations. Meng et al. [22] discussed a cloudlet placement problem to minimize delay between users and cloudlets. Authors considered task transmission delay as total completion delay. Yang et al. [24] discussed a cloudlet placement and task allocation problem in MEC using benders decomposition to minimize delay and energy consumption. Zeng et al. [23] devised a cost-effective ES placement scheme using simulated annealing and dominating set to minimize the number of ESs while ensuring QoS requirement. Wang et al. [25] proposed a binary-based differential evolution cuckoo search algorithm to minimize the latency and deployment cost for optimal cloudlets deployment in IoT. Fan et al. [27] devised a cost aware cloudlet placement scheme in MEC to minimize cloudlet cost and average end to end delay using a heuristic solution. These schemes [19] [20] [22] [23] [24] [25] [27] are designed to minimize access delay and ignore the balanced end nodes assignment that causes poor utilization of edge resources. Further, the schemes [19] [20] [22] [23] [27] ignore data processing delay at ESs.

Liang et al. [28] discussed a location-aware service deployment scheme to minimize cloud service access latency. Authors adopted k -means clustering to form the device cluster and select Edge Cloud Server (ECS) nearest to cluster center for service instance deployment. Further, a load balancing mechanism is

proposed that remapped the mobile devices of overloaded ECS to under-loaded ECS. It increases the number of hops in communication paths that results in higher delay and energy consumption. Li et al. [21] designed an ES placement scheme using k -means algorithm to reduce the average completion time of the system. Guo et al. [30] utilized k -means algorithm to place a fixed number of ESs with the objective of communication delay minimization and balance the workload. Wang et al. [29] described an efficient ES placement strategy in MEC to minimize workload difference among ESs and access delay between end user devices and ESs using Mixed-Integer Programming (MIP). The scheme requires prior information about workload of the base station. Lahderanta et al. [18] discussed an ES placement algorithm in edge computing infrastructure, named PACK, to minimize distance between server and their associated access points and takes care of load balancing. Although, in absence of historic workload information the objective function is reduced to end-to-end latency minimization only. These schemes [18] [28] [29] utilized Euclidean distance as a decision parameter in end node assignment and restricted to direct (one-hop) communication scenario that leads to higher energy consumption and delay in resource-constrained networks.

Jia et al. [31] discussed a cloudlet placement scheme in WMAN to address access delay and optimal user to cloudlet assignment. Authors devised Heaviest Access Point First (HAF) and Density based Clustering (DBC) schemes. HAF selects the location of heavily loaded access points for the placement of cloudlets. DBC selects access points with dense user area to place cloudlets and achieves optimal user to cloudlet assignment using relative distance. DBC addresses redundancy factor to determine placement locations. However, it randomly selects optimal placement location among candidate ones that leads to non-optimal cloudlet placement that results in higher energy consumption and delay in resource poor networks.

Chin et al. [26] proposed a k -clustering algorithm for queuing model-based ES placement in mobile cloud network to minimize the network traffic. The k -clustering scheme selects k nodes having the highest request as ESs for initial cluster formation. It utilizes shortest hop distance to assign the request of remaining nodes among ESs. Next, in each cluster a new node is selected as ES, and requests of remaining cluster nodes are assigned to the corresponding node. This process runs iteratively until the traffic load difference between previous and current iteration is less than a predefined threshold. This scheme selects the initial cluster center without considering redundancy factor that leads to higher access delay and reduces the edge resource(s) utilization. Moreover, the optimal initial k ES selection (if more than k nodes having same highest request) is not addressed. Random selection of initial ESs causes higher delay and reduces edge resource utilization. Wang et al. [35] discussed a fog node deployment scheme in a smart factory using discrete differential evolution algorithm to minimize the response time and achieve load balancing. The schemes assumes that communication delay between fog node and terminal device(s) is constant. However, in realistic wireless network scenario, communication delay is defined as function of distance/hop-count and has major impact on response time. The

assumption of constant communication delay converts the response time minimization and load balancing cost function to only load balancing. Wang et al. [36] discussed a binary-based gray wolf genetic scheme for ES placement scheme in intelligent manufacturing to achieve cost reduction and load balancing. However, authors did not consider the communication and data processing delay that directly affects the system performance. The delay should be minimize in order to cater the near-real time system requirements. Kasi et al. [37] perform the comparative analysis of local search algorithms (hill climbing and simulated annealing) with genetic algorithm for ES placement in IIoT and cellular network. The analysis indicates that genetic algorithm outperforms the considered local search algorithms.

Density based clustering schemes are also utilized in WSN to reduce latency and energy consumption [33] [34]. Veeramani et al. [33] proposed a heuristic solution that selects maximum degree nodes for multi-sink placement. Peng et al. [34] discussed a density based network clustering and selects initial Cluster Heads (CHs) using node's residual energy. Since, initially each node has same energy level that leads to random CH selection. It relocates CHs role among other nodes over course of time based on some parameters. On the other hand, ECN placement location needs to be determined optimally since a fixed number of ECNs (high cost and resource-rich devices compared to IoT nodes) are placed at these locations at once and do not support location migration flexibility like CHs. These schemes only considers the direct communication scenario and balanced assignment issue remains unaddressed.

The proposed work collectively addresses the ECN placement and balanced IoT-ECN mapping problem compared to the above-discussed schemes. The proposed *coECNP* scheme determines the optimal locations for ECN placement by taking care of the above-discussed shortcomings of existing works to minimizes the maximum hop distance between IoT-ECN pair. It takes care of redundancy issue and utilizes hop distance between candidate locations and previously placed ECNs to determine optimal location(s) for ECN placement that mitigates the close placement of ECNs and leads to reduced access cost between IoT-ECN pair (see section 4.1). Next, it maps the IoT nodes among ECNs in a balanced fashion using a mapping mechanism (see section 4.2) to accomplish balanced IoT-ECN mapping. Thus, *coECNP* minimizes IoT nodes delay and utilizes edge resources efficiently.

3. System Model and Problem Formulation

The EIaaS is abstracted as an un-directed graph $G(\mathcal{V}, E)$, where, \mathcal{V} is the set of n IoT nodes defined as $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, and E is the set of links among them. The number of ECNs (k) are determined using Elbow method [38] [39]. The Elbow method is used to determine the appropriate number of clusters in clustering analysis. The cost function for the same is defined as,

$$\mathcal{J} = \sum_{i=1}^k \sum_{a \in C_i} |a - C_i|^2 \quad (1)$$

Where, a is the element of cluster C_i and k is the number of clusters ($|C_i|$). The curve between cost function (\mathcal{J}) and number of clusters (k) follows the shape of elbow and the k value corresponding to elbow point is selected as appropriate number of clusters.

Each IoT node ($v_{(i|i=1,2,\dots,n)} \in \mathcal{V}$) is enabled with sensing and limited processing, storage, communication functionalities (such as Arduino/TelosB board equipped with low storage and micro-controller). Each ECN ($u_{(j|j=1,2,\dots,k)} \in \mathcal{U}$) is a full functional EIaaS device equipped with more processing and storage capabilities (such as single board computer (example Raspberry Pi), industrial PCs etc.) than IoT nodes. Two arbitrary IoT nodes, v_i and v_j are said to be neighbor if $EU^D(v_i, v_j) \leq R_{max}$. IoT nodes communicate with respective ECN either directly or using a multi-hop path over wireless channel. It is assumed that network gateway is aware of location and IDs of IoT nodes.

In EIaaS enabled IoT network, each IoT node produces a significant amount of data that requires to be processed at respective ECN in near real-time fashion to minimize the overall delay so as to meet the applications' requirements. The total delay in this process is caused due to end-to-end (E2E) delay from IoT node to the respective ECN and the data processing delay at that ECN [4]. The total delay ($\mathcal{D}_{u_j}^{v_i}$) between i^{th} IoT node ($v_{i|1 \leq i \leq n} \in \mathcal{V}$) and respective j^{th} ECN ($u_{j|1 \leq j \leq k} \in \mathcal{U}$) is characterized by Eq. 2.

$$\mathcal{D}_{u_j}^{v_i} = \mathcal{D}_{E2E}^P(v_i, u_j) + \mathcal{D}_{proc}^{v_i}(u_j) \quad (2)$$

Where, $\mathcal{D}_{E2E}^P(v_i, u_j)$ and $\mathcal{D}_{proc}^{v_i}(u_j)$ are the end-to-end delay from i^{th} IoT node to j^{th} ECN along path P and data processing delay at j^{th} ECN for i^{th} IoT node, respectively.

End-to-End delay between an IoT node and ECN consists of link delay (transmission and propagation delay) and forwarding delay at intermediate IoT nodes. Let path P between i^{th} IoT node (v_i) and j^{th} ECN (u_j) consist of a set of links and intermediate nodes (including v_i and u_j) defined as $E(P)$ and $\mathcal{V}(P)$. Assume $\mathcal{D}_L^P(v_i, u_j)$ and $\mathcal{D}_F^P(v_i, u_j)$ represents the link delay on a link $e \in E(P)$ and data forwarding delay at an intermediate node $v \in \mathcal{V}(P)$. The $\mathcal{D}_{E2E}^P(v_i, u_j)$ is given by Eq. 3.

$$\mathcal{D}_{E2E}^P(v_i, u_j) = \sum_{e \in E(P)} \mathcal{D}_L^P(v_i, u_j) + \sum_{v \in \mathcal{V}(P)} \mathcal{D}_F^P(v_i, u_j) \quad (3)$$

It is evident from Eq. 3 that link and forwarding delay follows a proportionality relationship to the path length(hop distance) from node to respective ECN. The hop distance from IoT node to ECN varies as per ECN placement strategies. Furthermore, the data processing delay of i^{th} IoT node (v_i) at j^{th} ECN (u_j) is given by Eq. 4.

$$\mathcal{D}_{proc}^{v_i}(u_j) = \max(0, \mathcal{D}_{proc}^{v_i \neq i}(u_j)) + \mathcal{D}_{pra}^{v_i}(u_j) \quad (4)$$

Where, $\max(0, \mathcal{D}_{proc}^{v_i \neq i}(u_j))$ and $\mathcal{D}_{pra}^{v_i}(u_j)$ represents the waiting time of v_i at u_j and data processing time of v_i at u_j after getting ECN resources. The waiting time varies as per mapping schemes and $\mathcal{D}_{pra}^{v_i}(u_j)$ depends on computation requirement to process the data. Moreover, IoT nodes mapping among ECNs also impacts E2E delay and edge resource utilization.

Table 2: Notations

Symbol	Descriptions
C_i	Cluster center
a	Element of C_i
\mathcal{J}	Elbow method cost function
\mathcal{V}	Set of IoT nodes $v_{i i=1,2,\dots,n}$
\mathcal{V}'	Set of unassigned IoT nodes
E	Set of link among IoT nodes
\mathcal{U}	Set of edge computational nodes $u_{j j=1,2,\dots,k}$
$NB(v_i)$	Set of neighbors of i^{th} IoT node
S_j	Set of IoT nodes mapped to j^{th} ECN (u_j)
IND	Set of IoT nodes having den^{max}
IND^i	i^{th} element of set IND
S'	Set of $S_{j j=1,2,\dots,k}$
NH	Set of next-hop neighbors of IoT nodes
NH^{CMN}	Set of common next-hop IoT nodes
NH^U	Set of unique next-hop IoT nodes
R_{max}	Maximum transmission range of IoT nodes
$EU^D(v_i, v_j)$	Euclidean distance between IoT nodes v_i and v_j
$H^D(v_i, v_j)$	Hop distance between v_i and v_j
$den(v_i)$	Density of i^{th} IoT node
$ X $	Cardinality of set X
den^{max}	Value of maximum density
$dist_{hop}^{max}$	Hop distance of candidate IoT node, having den^{max} located at maximum hop distance from locations of previous ECNs
$index^{max}$	Index of node having maximum density
$loc(v_i)$	Location of i^{th} IoT node
$\mathcal{D}_{u_j}^{v_i}$	Total delay between i^{th} IoT node and j^{th} ECN
$\mathcal{D}_{E2E}^P(v_i, u_j)$	End-to-End delay from i^{th} IoT node to j^{th} ECN along path P
$\mathcal{D}_{proc}^{v_i}(u_j)$	Data processing delay at j^{th} ECN for i^{th} IoT node
$\mathcal{D}_L^P(v_i, u_j)$	Link delay along path P from i^{th} IoT node to j^{th} ECN
$\mathcal{D}_F^P(v_i, u_j)$	Data forwarding delay at intermediate nodes along path P from i^{th} IoT node to j^{th} ECN
$\mathcal{D}_{pra}^{v_i}(u_j)$	Data processing delay for i^{th} IoT node at j^{th} ECN after getting resources
Υ_{ij}, Ψ_{ij}	Binary variables for mapping and placement
ECN_{Dis}^{Loc}	ECN location discovery module
IoT_{Map}^{ECN}	IoT nodes mapping module
$EIaaS_{conf}$	EIaaS topology configuration module
$ECN_{ S }^{Min}$	Minimum set assignment first mapping scheme
$EECN_{ S }^{Min}$	Extended $ECN_{ S }^{Min}$ scheme
SD	Standard deviation

The above discussion formulates the foundation to consider ECN placement and IoT nodes mapping among ECNs for topology formation in EIaaS paradigm to serve latency-sensitive applications. The list of important symbols used in this work is summarized in Table 2.

3.1. Problem Formulation

The ECN placement and IoT nodes mapping among ECNs play a pivotal role in total delay minimization of IoT nodes. This work formulates the IoT nodes delay minimization problem as follows. Given the set of n IoT nodes and k ECNs, deter-

mine the k optimal locations ($loc_1, loc_2, \dots, loc_k$) to place ECNs and map IoT nodes among ECNs such that total delay of IoT nodes is minimized, i.e.

$$\text{Minimize } \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k \mathcal{D}_{u_j}^{v_i} \Upsilon_{ij} \quad (5)$$

Subject to constraints

$$\bigcup_{j=1}^k S_j = \mathcal{V} \quad (6)$$

$$\bigcap_{j=1}^k S_j = \phi \quad (7)$$

$$\sum_{j \in k} \Psi_{ij} = 1, \forall i \in n \quad (8)$$

$$\sum_{i=1}^n \sum_{j=1}^k \Psi_{ij} = k \quad (9)$$

Eq. (5) represents the cost function with the objective of delay minimization. Constraint (6) and (7) assure that each IoT node is served and by exactly one ECN. Constraint (8) represents that each ECN is deployed at only one location, while constraint (9) assures the placement of k ECNs in the network. Υ_{ij} and Ψ_{ij} are binary variables given by Eq. (10) and Eq. (11).

$$\Upsilon_{ij} = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ IoT node } (v_i) \text{ is mapped to } j^{\text{th}} \text{ ECN } (u_j) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$\Psi_{ij} = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ ECN is placed at location of } j^{\text{th}} \text{ IoT node} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

4. The coENCP mechanism

This section discusses the proposed mechanism executed at the network gateway. *coENCP* scheme is composed of: (i) ECN Location Discovery (ECN_{Dis}^{Loc}), (ii) IoT nodes Mapping (IoT_{Map}^{ECN}), and (iii) EIaaS topology configuration ($EIaaS_{Conf}$) modules, as depicted in Fig. 2. ECN_{Dis}^{Loc} and IoT_{Map}^{ECN} modules determine optimal locations for ECNs placement and optimal mapping of IoT nodes among ECNs such that IoT nodes total delay is minimized. Finally, $EIaaS_{Conf}$ module configures the EIaaS topology by establishing a connection between IoT nodes and respective ECN.

4.1. ECN Location Discovery

ECN_{Dis}^{Loc} module is responsible for discovering optimal locations to place the ECNs in the network. Algorithm 1 describes ECN location discovery procedure. The module accommodates a location discovery function f that associates each element of set \mathcal{U} to a unique element of set that consists of IoT nodes locations, defined as, $f: \mathcal{U} \rightarrow loc(\mathcal{V})$. For instance, if location discovery function selects location of i^{th} IoT node to place j^{th} ECN then $f(u_j) = loc(v_i)$.

These locations are determined based on density of IoT nodes, where, density is defined as number of neighbors of IoT node (line 7-9) given as:

$$den(v_i) = \sum_{j \in \mathcal{V}'} \theta(EU^D(v_i, v_j) - R_{max}) \quad (12)$$

Algorithm 1: ECN location discovery

```

Input :  $\mathcal{V}, \mathcal{U}, loc(v_{i|i=1,2,\dots,n}), k$ 
Output: ECNs placement locations
1 begin
2    $count = 1, \mathcal{V}' = \mathcal{V}, S_{j|j=1,2,\dots,k} = \phi$ 
3   while  $count \leq k$  do
4      $dist_{hop}^{max} = 0, den^{max} = 0$ 
5     /* Highests density IoT node
6     determination (line 5-20) */
7     for each  $v_{i|i=1,2,\dots,|\mathcal{V}'|} \in \mathcal{V}'$  do
8        $den(v_i) = 0$ 
9       for each  $v_{j|j=1,2,\dots,|\mathcal{V}'|} \in \mathcal{V}'$  do
10        if  $EU^D(v_i, v_j) \leq R_{max}$  then
11           $den(v_i) = den(v_i) + 1, NB(v_i) = v_j$ 
12        end
13        if  $den(v_i) > den^{max}$  then
14           $den^{max} = den(v_i)$ 
15           $var = 1, IND = \phi$ 
16        end
17        else if  $den(v_i) == den^{max}$  then
18           $var = var + 1,$ 
19        end
20         $IND^{var} = i$ 
21      end
22      /* Location determination for ECN
23      placement (line 21-35) */
24      if  $count = 1$  then
25         $loc(u_{count}) = loc(v_{IND^{count}})$ 
26         $S_{count} = NB(v_{IND^{count}})$ 
27      end
28      else
29        for each  $v_{r|r=1,2,\dots,|IND|} \in IND$  do
30           $dist_{hop} = H^D(u_{count-1}, v_r)$ 
31          if  $dist_{hop} \geq dist_{hop}^{max}$  then
32             $dist_{hop}^{max} = dist_{hop}$ 
33             $index^{max} = r$ 
34          end
35        end
36         $loc(u_{count}) = loc(v_{(index^{max})})$ 
37         $S_{count} = NB(v_{(index^{max})})$ 
38      end
39      /* IoT node set update */
40       $\mathcal{V}' = \mathcal{V}' - S_{count}$ 
41       $count = count + 1$ 
42    end
43  end

```

Where, $\theta(EU^D(v_i, v_j) - R_{max}) = 1$ if $(EU^D(v_i, v_j) - R_{max}) \leq 0$, and $\theta(EU^D(v_i, v_j) - R_{max}) = 0$, otherwise.

In first iteration, ECN_{Dis}^{Loc} module assigns the IoT nodes to set \mathcal{V}' and calculates the density of each IoT node. ECN_{Dis}^{Loc} determines the highest density IoT node in \mathcal{V}' (line 12-20).

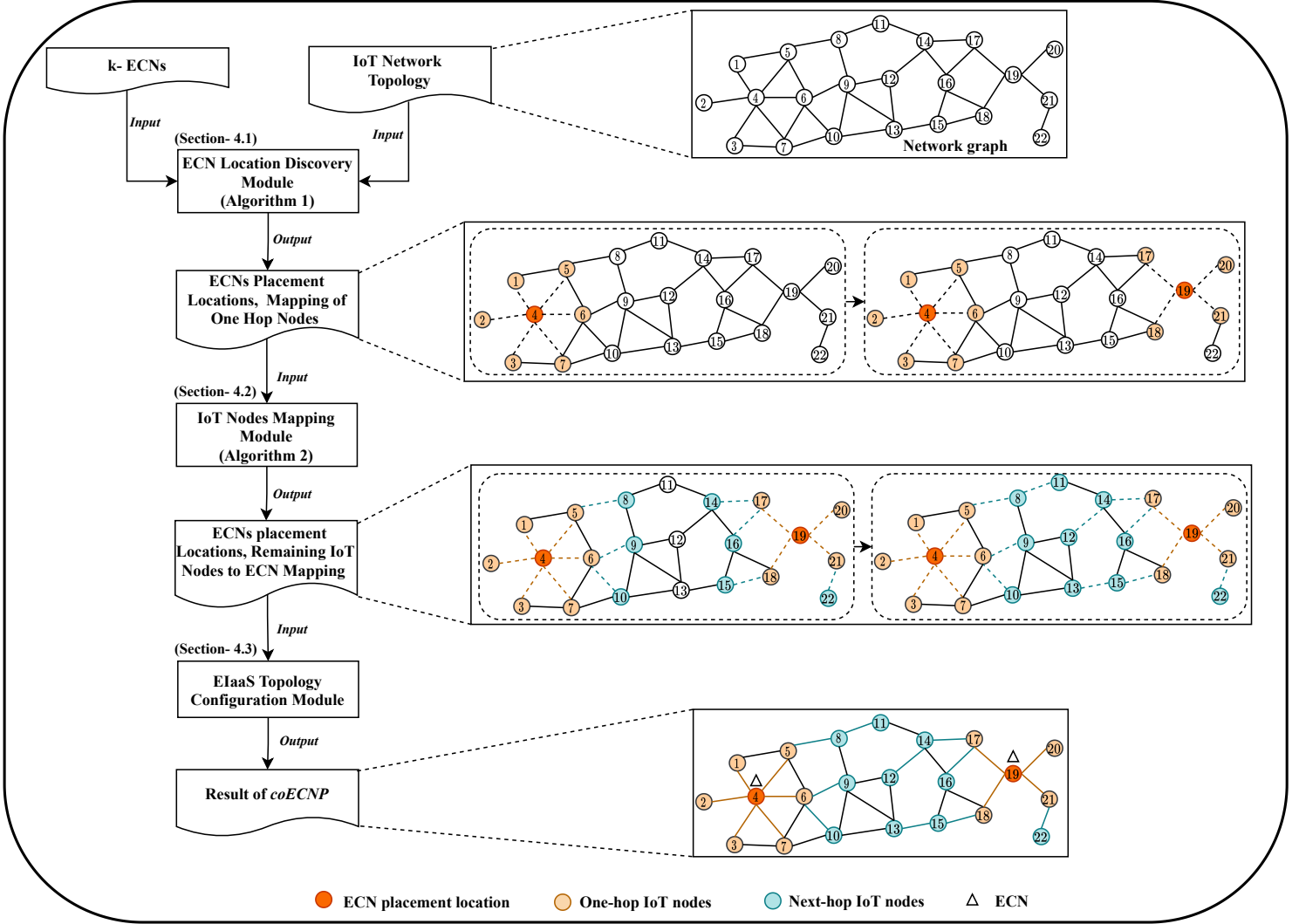


Figure 2: Illustration of proposed *coECNP* scheme

It selects location of highest density node to place first ECN. Next, ECN_{Dis}^{Loc} assigns highest density IoT node and its one-hop neighbors to a corresponding set S_1 at first ECN (line 21-34).

For example, in Fig. 2 suppose the total number of ECNs to be placed are two ($k = 2$). ECN_{Dis}^{Loc} selects v_4 as highest density node and $loc(v_4)$ to place first ECN (u_1). It also assigns v_4 and its one-hop neighbors to corresponding set S_1 . Thus, at the end of first iteration, ECN_{Dis}^{Loc} returns the optimal location to place first ECN (u_1) and set S_1 that consists of highest density IoT node and its one-hop neighbors.

In next iteration, ECN_{Dis}^{Loc} calculates the new highest density IoT node to identify the placement location of next ECN. There can be a situation where IoT nodes mapped to ECN(s) in the previous iteration(s) can again participate in density determination during the location discovery process for next ECN. This, in turn, leads to redundant density calculation for IoT nodes.

For instance, in Fig. 2, v_6 and v_9 become the maximum density IoT nodes with density value 5. The neighbor set of v_6 consists of three redundant neighbors v_4 , v_5 , and v_7 , which are already assigned to S_1 . Similarly, the neighbor set of v_9 con-

tains v_6 as redundant neighbor. To avoid such redundancy in density calculation, ECN_{Dis}^{Loc} determines the new highest density IoT node(s) on an updated set of IoT nodes. It obtains the set of updated IoT nodes by removing the nodes which are already assigned to ECN(s) from total IoT nodes (line 36).

ECN_{Dis}^{Loc} obtains $\mathcal{V} = \{v_8, v_9, \dots, v_{22}\}$ and selects $loc(v_9)$, $loc(v_{16})$, and $loc(v_{19})$ as candidate locations to place second ECN (u_2). ECN_{Dis}^{Loc} selects the location of IoT node situated at maximum hop distance from the location of previous ECN(s) among candidate locations to place next ECN (line 26-31). ECN_{Dis}^{Loc} module selects $loc(v_{19})$ to place u_2 (see Fig. 2). One-hop IoT nodes assignment to corresponding set and calculation of updated set of IoT nodes follows the same procedure as first iteration. This module returns the optimal ECN placement locations and set of one-hop IoT nodes assigned to respective ECNs.

ECN_{Dis}^{Loc} module returns $loc(v_4)$, $loc(v_{19})$ as ECN placement locations, and $S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $S_2 = \{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}\}$ as set of one-hop IoT nodes, respectively, for the scenario depicted in Fig. 2.

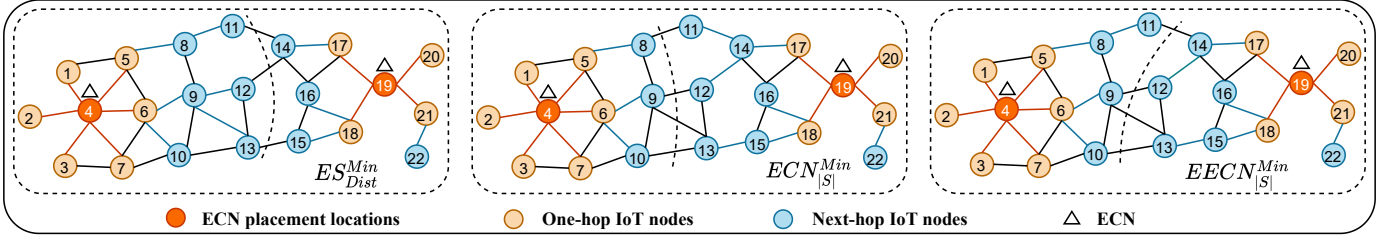


Figure 3: IoT nodes mapping under different schemes

4.2. IoT nodes Mapping

Execution of IoT_{Map}^{ECN} module is initiated after termination of ECN_{Dis}^{Loc} . IoT_{Map}^{ECN} module consists of a mapping function defined as, $g : \mathcal{V} \rightarrow \mathcal{U}$. Where, \mathcal{V} is the set of IoT nodes returned by ECN_{Dis}^{Loc} module after k^{th} iteration termed as set of remaining IoT nodes ($\mathcal{V} = \{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{22}\}$) in Fig. 2). Mapping function associates each IoT node ($v_i \in \mathcal{V}$) to a unique ECN ($u_j \in \mathcal{U}$) to accomplish efficient EIIaaS design.

Minimum Set Assignment First ($ECN_{|S|}^{Min}$) mechanism is proposed to achieve balanced mapping of IoT nodes among ECNs. It maps next-hop IoT nodes among ECNs by taking care of previously assigned IoT nodes at each ECN. It compares sets of IoT nodes mapped to ECNs ($S_{j|j=1,2,\dots,k}$) and selects the ECN with minimum IoT nodes as the candidate one for next-hop IoT nodes mapping. $ECN_{|S|}^{Min}$ scheme updates the set of remaining IoT nodes by removing currently assigned next-hop IoT nodes to candidate ECN from current \mathcal{V} . $ECN_{|S|}^{Min}$ assigns next-hop IoT nodes to candidate ECN in each iteration and terminates when updated set of remaining IoT nodes becomes empty ($\mathcal{V} = \phi$).

For example, Fig. 3 depicts a sample scenario of $ECN_{|S|}^{Min}$ based mapping for two ECNs. $ECN_{|S|}^{Min}$ mechanism selects second ECN (at $loc(v_{19})$ in Fig. 3) as candidate ECN. It maps IoT nodes $\{v_{14}, v_{15}, v_{16}, v_{22}\}$ to second ECN and updates the set of IoT nodes at respective ECN as $S_2 = \{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{14}, v_{15}, v_{16}, v_{22}\}$. $ECN_{|S|}^{Min}$ obtains the updated set of remaining IoT nodes as $\mathcal{V} = \{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\}$, after removal of currently mapped IoT nodes $\{v_{14}, v_{15}, v_{16}, v_{22}\}$ at second ECN. Thus, at the end of first iteration, seven and nine IoT nodes are mapped to first and second ECN, respectively.

$ECN_{|S|}^{Min}$ selects first ECN (at $loc(v_4)$ in Fig. 3) as candidate ECN in next iteration and assigns IoT nodes $\{v_8, v_9, v_{10}\}$ to it. After assigning these nodes to respective ECN, $ECN_{|S|}^{Min}$ obtains $\mathcal{V} = \{v_{11}, v_{12}, v_{13}\}$. In next iteration, second ECN (having 9 nodes) is selected as candidate ECN to assign IoT nodes $\{v_{11}, v_{12}, v_{13}\}$. After this assignment, set of remaining IoT nodes becomes empty ($\mathcal{V} = \phi$) that leads to the termination of $ECN_{|S|}^{Min}$. Thus at the end, IoT nodes mapped to first and second ECN given as, $S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$, $S_2 = \{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{14}, v_{15}, v_{16}, v_{22}, v_{11}, v_{12}, v_{13}\}$.

Fig. 3 depicts the comparison of proposed $ECN_{|S|}^{Min}$ scheme with *Shortest Distance Edge Server* (ES_{Dist}^{Min}) assignment scheme. ES_{Dist}^{Min} assigns next-hop IoT nodes to the nearest

ECN without taking care of previously assigned nodes. It returns $S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\}$ and $S_2 = \{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{14}, v_{15}, v_{16}, v_{22}\}$ after termination. $ECN_{|S|}^{Min}$ and ES_{Dist}^{Min} maps (10, 12) and (9, 13) nodes to first and second ECN, respectively. It indicates that the proposed $ECN_{|S|}^{Min}$ scheme minimizes the difference of assigned IoT nodes among ECNs as compared to ES_{Dist}^{Min} scheme that leads to balanced mapping of IoT nodes.

CRITICAL CASE : Although $ECN_{|S|}^{Min}$ mechanism performs better than ES_{Dist}^{Min} scheme, in some cases, it may lead to unbalanced mapping of IoT nodes among ECNs. For instance, j^{th} and $(j+1)^{th}$ ECN consists of $|S_j|$ and $|S_{(j+1)}|$ nodes, where $|S_j| > |S_{(j+1)}|$. Next, $ECN_{|S|}^{Min}$ selects $(j+1)^{th}$ ECN as candidate ECN to assign next-hop IoT nodes. There can be some nodes in the set of next-hop nodes of $(j+1)^{th}$ ECN that also belongs to next-hop node set of j^{th} ECN. $ECN_{|S|}^{Min}$ assigns all nodes to $(j+1)^{th}$ ECN that reduces the number of nodes for j^{th} ECN, causes unbalanced assignment. For instance, IoT nodes $\{v_{11}, v_{12}, v_{13}\}$ belong to next-hop nodes for both ECNs, however, $ECN_{|S|}^{Min}$ assigns these nodes to second ECN (see $ECN_{|S|}^{Min}$ scheme based mapping of IoT nodes in Fig. 3).

To handle such unbalanced mapping of IoT nodes, an extended version of $ECN_{|S|}^{Min}$ scheme is proposed termed as *Extended-ECN* ($EECN_{|S|}^{Min}$).

4.2.1. Extended-ECN $_{|S|}^{Min}$

The core idea of the $EECN_{|S|}^{Min}$ scheme is to take care of unique and common next-hop neighbors during the mapping of remaining IoT nodes among ECNs. Fig. 4 depicts the i^{th} iteration instance of IoT nodes to ECN mapping. Algorithm 2 describes the steps involved in $EECN_{|S|}^{Min}$. The $EECN_{|S|}^{Min}$ scheme sorts the set S' that comprises the set of IoT nodes at ECNs ($S' = \{S_1, S_2, \dots, S_k\}$) in descending order (line 3). It selects last two adjacent elements S'_k and $S'_{(k-1)}$ (where $|k|$ is the cardinality of set S') from sorted S' (line 6). $EECN_{|S|}^{Min}$ selects the ECNs corresponding to S'_k and $S'_{(k-1)}$ as candidate ECNs to map the next-hop IoT nodes. For instance, if sorted S' is given as $S' = \{S_1, S_3, S_4, S_2\}$ then last two adjacent element S'_k and $S'_{(k-1)}$ represents the S_2 (set of IoT nodes at second ECN) and S_4 (set of IoT nodes at fourth ECN), respectively.

$EECN_{|S|}^{Min}$ determines set of next-hop neighbors for candidate ECNs. IoT nodes that belongs to set of next-hop neighbors of both candidate ECNs are termed as common next-hop IoT nodes. The set of unique next-hop IoT nodes is obtained by re-

Algorithm 2: Extended-ECN_{|S|}^{Min} scheme

Input : \mathcal{V}' , $S_{j|j=1,2,\dots,k}$ **Output**: Balanced IoT-ECN mapping

```
1 begin
2   while  $\mathcal{V}' \neq \phi$  do
3      $S' = \text{Sort}(S_{j|j=1,2,\dots,k})$ ,  $m = k$ ,  $c = 0$ 
4     while  $m > 1$  do
5       /*Identification of next-hop, common, and unique IoT node (line 6-17) */
6        $c = c + 1$ ,  $NH_{T_m}$ 
7        $T_m = S'_m$ ,  $T_{m-1} = S'_{m-1}$ 
8        $NH_{T_{(m-1)}} = NB(v_{i|i=1,2,\dots,|T_{(m-1)}} \in T_{(m-1)}) \in \mathcal{V}'$ 
9       /*  $k^{\text{th}}$  comparision ( $k = 1$ ) */
10      if  $c == 1$  then
11         $NH_{T_m} = NB(v_{i|i=1,2,\dots,|T_m|} \in T_m) \in \mathcal{V}'$ 
12         $NH^{CMN} = NH_{T_m} \cap NH_{T_{(m-1)}}$ 
13         $NH_{T_m}^U = NH_{T_m} - NH^{CMN}$ 
14      end
15      /*  $k^{\text{th}}$  comparision ( $1 < k < (k - 1)$ ) */
16      else if  $c > 1$  then
17         $NH^{CMN} = NH_{T_m}^U \cap NH_{T_{(m-1)}}$ 
18         $NH_{T_m}^U = NH_{T_m}^U - NH^{CMN}$ 
19      end
20       $NH_{T_{(m-1)}}^U = NH_{T_{(m-1)}} - NH^{CMN}$ 
21      /* Unique and common IoT nodes' mapping (line 19-37) */
22       $T_m = T_m \cup NH_{T_m}^U$ 
23      /*  $(k - 1)^{\text{th}}$  comparision */
24      if  $m == 2$  then
25         $T_{(m-1)} = T_{(m-1)} \cup NH_{T_{(m-1)}}^U$ 
26      end
27       $d = |T_m| - |T_{m-1}|$ 
28       $C_1 = \phi$ ,  $C_2 = \phi$ 
29      if  $d > 0$  then
30         $C_1 = v_{i|i(i=1,2,\dots,d \in N^{CMN})}$ 
31         $T_{m-1} = T_{m-1} \cup C_1$ 
32      end
33      else if  $d < 0$  then
34         $C_1 = v_{i|i(i=1,2,\dots,d \in N^{CMN})}$ 
35         $T_m = T_m \cup C_1$ 
36      end
37       $N^{CMN} = N^{CMN} - C_1$ 
38       $C_1 = v_{i|i(i=1,2,\dots, \lceil \frac{|N^{CMN}|}{2} \rceil \in N^{CMN})}$ 
39       $C_2 = v_{i|i(i= \lceil \frac{|N^{CMN}|}{2} \rceil + 1, \dots, |N^{CMN}| \in N^{CMN})}$ 
40       $T_m = T_m \cup C_1$ ,  $T_{m-1} = T_{m-1} \cup C_2$ 
41       $S'_m = T_m$ ,  $S'_{m-1} = T_{m-1}$ 
42       $m = m - 1$ 
43    end
44    /* IoT node set update */
45     $\mathcal{V}' = \mathcal{V}' - S'$ 
46  end
47 end
```

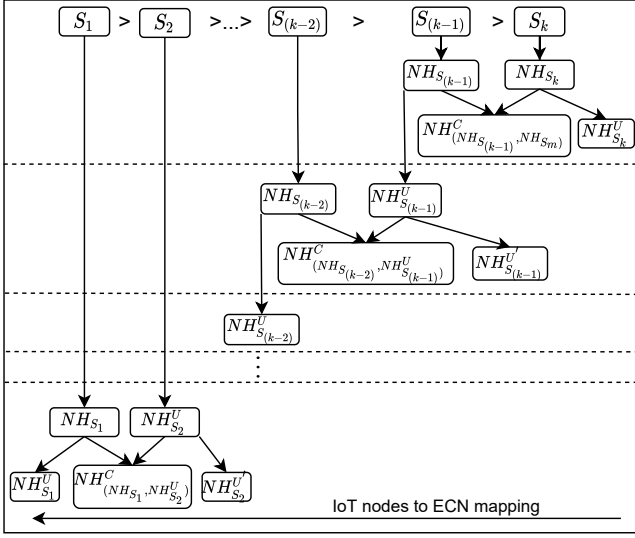


Figure 4: IoT nodes to ECN mapping using $EECN_{|S|}^{Min}$

moving the set of common next-hop IoT nodes from next-hop neighbors set at each candidate ECN (line 7-17).

$EECN_{|S|}^{Min}$ scheme selects the k^{th} ECN (ECN corresponding to S'_k) and maps the respective unique next-hop IoT nodes to it (line 18). After mapping unique next-hop neighbors to S'_k , proposed scheme calculates the cardinality difference (d) between set S'_k and $S'_{(k-1)}$. The value of cardinality difference (d) determines the assignment policy for the nodes belonging to the set of common next-hop IoT nodes. $EECN_{|S|}^{Min}$ maps common nodes equally between both candidate ECNs in case of zero difference ($d = 0$). In other cases (when value of $d \neq 0$), proposed scheme selects the candidate ECN having minimum nodes to assign d nodes from common next-hop IoT nodes. Remaining common next-hop IoT nodes are assigned equally between both candidate ECNs (line 22-37).

$EECN_{|S|}^{Min}$ performs the similar procedure between next-hop neighbors of $S'_{(k-2)}$ and unique neighbor set of $S'_{(k-1)}$ in next comparison. Thus, it performs $(k - 1)$ comparison per iteration to map IoT nodes to the relevant ECNs. In each iteration's last comparison ($(k - 1)^{th}$), the value of d is determined after assigning unique next-hop IoT nodes to both candidate ECNs (line 19-20). The mapping procedure of common next-hop IoT nodes follows the same as previous ones.

Thus, $EECN_{|S|}^{Min}$ scheme achieves the objective of balanced mapping of IoT nodes among ECNs by taking care of currently assigned, unique, and common next-hop IoT nodes at each ECN. The proposed $EECN_{|S|}^{Min}$ (see Fig. 3 for IoT nodes mapping using $EECN_{|S|}^{Min}$ scheme) described in the following through an example.

- $EECN_{|S|}^{Min}$ applies sorting and selects u_1 (ECN at $loc(v_4)$ in Fig. 3) and u_2 (ECN at $loc(v_{19})$ in Fig. 3) as the candidate ECNs. Further, it determines the set of next-hop neighbors for both candidate ECNs as $NH_{u_1} = \{v_8, v_9, v_{10}\}$ and $NH_{u_2} = \{v_{14}, v_{15}, v_{16}, v_{22}\}$.
- Set of next-hop IoT nodes NH_{u_1} and NH_{u_2} did not have

any common node, therefore, $NH^{CMN} = \phi$. Set NH_{u_1} and NH_{u_2} are assigned to set of unique next-hop nodes at ECNs as $NH_{u_1}^U$ and $NH_{u_2}^U$.

- It assigns $NH_{u_1}^U$ and $NH_{u_2}^U$ to S_1 and S_2 , respectively. In such a way, $EECN_{|S|}^{Min}$ accomplishes two-hop IoT nodes mapping at respective ECNs.
- Next, $EECN_{|S|}^{Min}$ maps the IoT nodes located at three-hops distance from ECNs. It selects u_1 and u_2 as candidate ECNs and determines set of next-hop nodes as $NH_{u_1} = \{v_{11}, v_{12}, v_{13}\}$ and $NH_{u_2} = \{v_{11}, v_{12}, v_{13}\}$.
- Both ECNs have same nodes in set of next-hop IoT nodes, therefore, $NH_{u_1}^U = \phi$, $NH_{u_2}^U = \phi$ and $NH^{CMN} = \{11, 12, 13\}$.
- $EECN_{|S|}^{Min}$ determines value of d by taking cardinality difference of S_1 and S_2 , before mapping common next-hop IoT nodes.
- Since value of d is one, therefore, $EECN_{|S|}^{Min}$ maps one node from NH^{CMN} to S_2 and remaining nodes in NH^{CMN} are assigned equally between S_1 and S_2 .
- $EECN_{|S|}^{Min}$ scheme returns the sets of IoT nodes mapped at first and second ECN (u_1 and u_2) as $S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$ and $S_2 = \{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{14}, v_{15}, v_{16}, v_{22}, v_{12}, v_{13}\}$ after termination.
- $EECN_{|S|}^{Min}$ scheme maps 11 IoT nodes at each ECN.

It can be inferred from the above discussed illustration that difference of IoT nodes mapped between u_1 and u_2 is minimum in $EECN_{|S|}^{Min}$ compared to $ECN_{|S|}^{Min}$ and ES_{Dist}^{Min} , hence, leads to balanced mapping of IoT nodes.

Lemma 1. $EECN_{|S|}^{Min}$ minimizes total delay by reducing forwarding delay at intermediate IoT nodes and achieves efficient utilization of ECNs.

Proof. $EECN_{|S|}^{Min}$ scheme categorizes next-hop IoT nodes into unique and common IoT node set before mapping to ECNs. For example, proposed scheme selects u_i and u_j as candidate ECNs in q^{th} ($1 < q \leq h_{max}$) iteration. $EECN_{|S|}^{Min}$ determines next-hop nodes for i^{th} and j^{th} ECN as $NH_{u_i}^{(q-1)}$ and $NH_{u_j}^{(q-1)}$. It obtains the common and unique next-hop node for i^{th} and j^{th} ECN as: $NH_{(u_i, u_j)}^{CMN(q)} = NH_{u_i}^{(q-1)} \cap NH_{u_j}^{(q-1)}$, $NH_{u_i}^{U(q)} = NH_{u_i}^{(q-1)} - NH_{(u_i, u_j)}^{CMN(q)}$, and $NH_{u_j}^{U(q)} = NH_{u_j}^{(q-1)} - NH_{(u_i, u_j)}^{CMN(q)}$, respectively. It associates $NH_{u_i}^{U(q)}$ and $NH_{u_j}^{U(q)}$ to u_i and u_j . Further, it associates $NH_{(u_i, u_j)}^{C(q)}$ between u_i and u_j such that $|(u_i) - (u_j)|$ is minimized that enhances ECNs utilization. Further, assume that v_i and v_j are the $(q - 1)^{th}$ hop IoT nodes mapped to ECN u_i and u_j that share ' α ' nodes as common in set $NH_{(u_i, u_j)}^{CMN(q)}$. The $EECN_{|S|}^{Min}$ scheme maps common ' α ' nodes between u_i and u_j in balanced manner rather than mapping them randomly to any ECN. Hence, it reduces the data forwarding burden on respective IoT node that leads to data forwarding delay minimization. \square

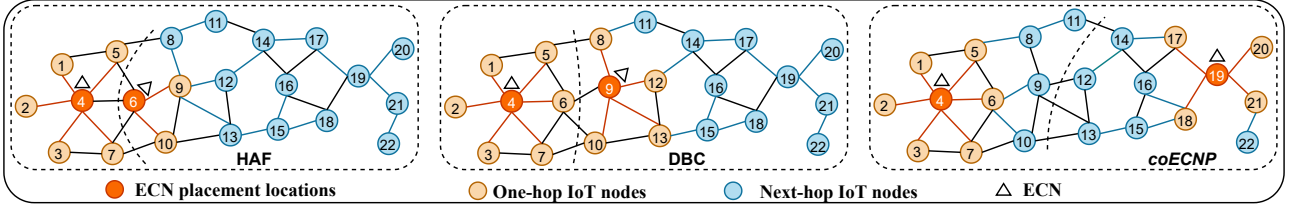


Figure 5: ECN placement and IoT nodes mapping among ECNs under various schemes

4.3. ElaaS Topology Configuration

$ElaaS_{Conf}$ module takes care of ECN placement at identified locations (by setting $\Psi_{ij} = 1$) and connection establishment between IoT nodes and respective ECN to configure the ElaaS topology. Once ECNs are placed, each ECN broadcast an ECN_Info packet that contains IoT-ECN mapping information. On reception of ECN_Info packet from an ECN, IoT node searches for its ID in the received message and transmits an Ack message to the ECN after successful ID matching. Thus, each IoT node is associated with its respective ECN (see ElaaS topology configuration module in Fig. 2).

4.4. Complexity Analysis

This subsection discusses the computational complexity of $coECNP$ scheme. ECN_{Dis}^{Loc} module determines IoT nodes density in $O(n^2)$ time and this procedure is repeated k times. Thus, time complexity of proposed location discovery module is $O(kn^2)$. $EECN_{|S|}^{Min}$ sorts $S_{j|j=1,2,\dots,k}$ based on number of associated IoT nodes that takes $O(k \log k)$ time and performs $(k-1)$ comparisons in each iteration. This procedure is repeated h_{max} times, where h_{max} is the network wide maximum hop distance from an IoT node to its respective ECN. Thus, time complexity of $EECN_{|S|}^{Min}$ scheme is $O(h_{max}(k \log k + (k-1)))$.

5. Comparative Analysis

This section discusses an example scenario to show the effectiveness of proposed scheme over HAF and DBC schemes. Fig. 5 depicts a sample scenario for ECN placement and IoT nodes mapping to corresponding ECN using proposed ($coECNP$), HAF, and DBC schemes. $coECNP$ scheme assigns (11,11) nodes to both ECNs with ECN access cost as 35 hops. HAF did not address the redundancy to determine ECNs placement locations, hence, leads to higher access cost than proposed scheme. Although, DBC takes care of redundancy in ECNs placement locations determination but fails to select the optimal location among candidate locations. Both HAF and DBC assign IoT nodes to ECN located at the shortest distance without taking care of previously assigned IoT nodes, which leads to an unbalanced assignment. HAF and DBC assign (6,16) and (7,15) IoT nodes among ECNs with ECN access cost as 66 hops and 47 hops, respectively. Moreover, $coECNP$ also reduces the maximum hop distance (h_{max}) from 6 hops (in case of DBC) and 7 hops (in case of HAF) to 3 hops.

It is obvious from the discussed example that the $coECNP$ scheme outperforms HAF and DBC schemes in terms of ECN

access cost and balanced IoT nodes mapping among ECNs, which plays a critical role in accomplishing objective function. Furthermore, proposed scheme takes $O(kn^2)$ time in ECN placement as same as DBC. Although, HAF performs ECN placement in $O(n^2)$ but leads to higher access cost. Moreover, HAF and DBC schemes perform IoT nodes mapping in $O(kn)$, whereas, proposed scheme achieves optimal mapping for IoT nodes in $O(h_{max}(k \log k + (k-1)))$. The above discussion indicates that proposed scheme determines optimal ECN placement location in same computational cost as DBC and attains optimal IoT nodes mapping among ECNs with lower computational cost in comparison to HAF and DBC.

6. Experiments and Results

Performance of the proposed $coECNP$ scheme is evaluated using discrete event simulator NS-3.30 in terms of ECN access cost, IoT nodes mapping among ECNs, energy consumption, average total delay, and edge resource utilization.

6.1. Experimental Setup

This section describes the parameter value used in experimentation. The simulation experiments are performed using NS-3.30 running on Ubuntu 18.04 with Intel core i5 processor (3.10 GHz) and 6GB RAM. The initial experimental setup consists of 50 randomly deployed IoT node in the area of $200 \times 200 m^2$ (see Fig. 6a). Subsequently, a denser network is constructed by varying the number of nodes (see Fig. 6b and Fig. 6c) and their transmission range (see Fig. 7a and Fig. 7b). The transmission range is initialized as $20m$ (lower transmission range results in disconnected graph). Table 3 summarizes the simulation parameters. The parameters in Table 3 are set accordingly to the specifications of TelosB sensor nodes as described in [40]. Energy consumption in transmitting B bits from IoT node to respective ECN is modeled using energy model discussed in [41]. We perform the simulation experiments using periodic, event, and query based data traffic patterns to evaluate the performance of $coECNP$ in various scenarios. In the simulation environment, we assumed a clear Line of Sight (LoS) and did not consider any moving obstacle such as a robot, vehicle, etc. The values for energy consumption in transmission and reception of one-byte packet are approximately $40.2 \mu J$ and $37.2 \mu J$, respectively. $coECNP$ scheme is compared with k -means, HAF, DBC, k -clustering, and Random (RND) ECN placement schemes in terms of ECN access cost (hops), IoT nodes mapping among ECNs, energy consumption (Joules), average total delay (Seconds), and edge resource utilization.

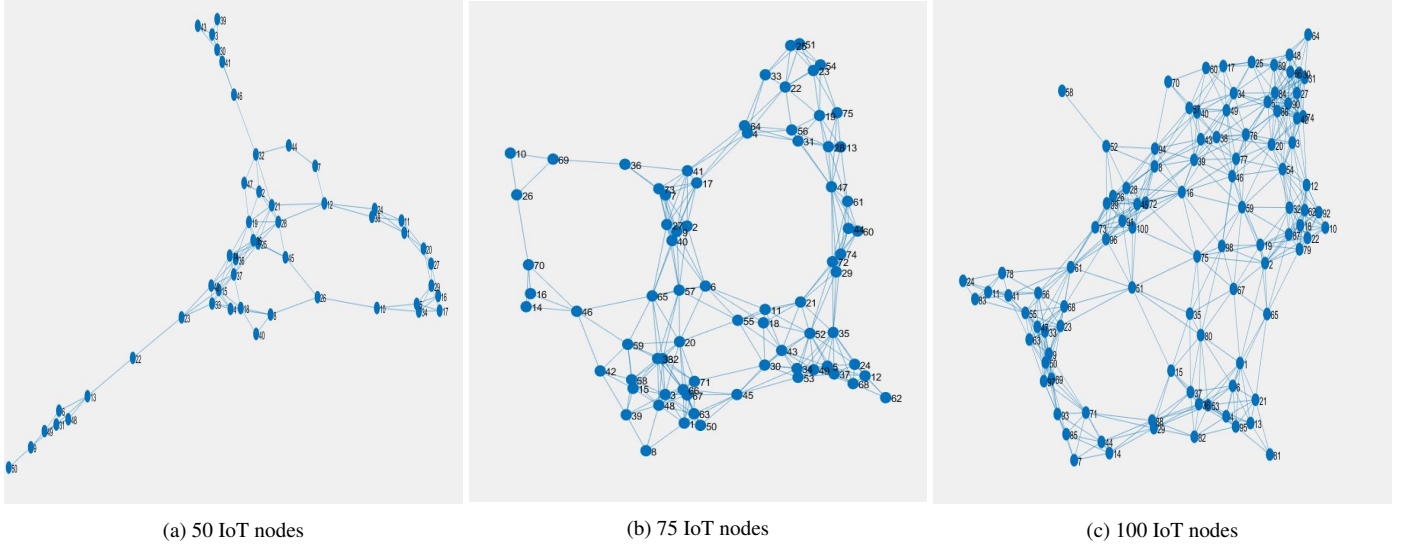
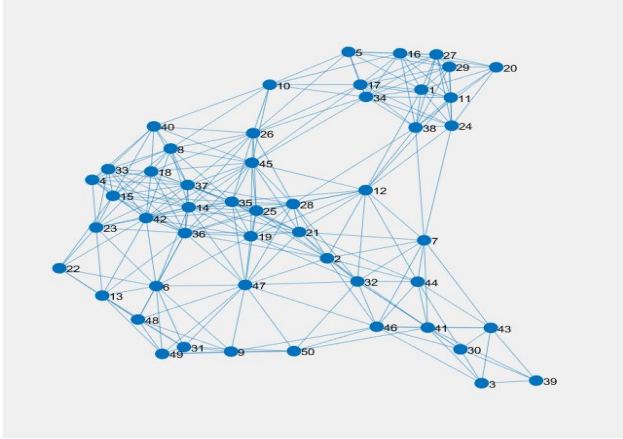
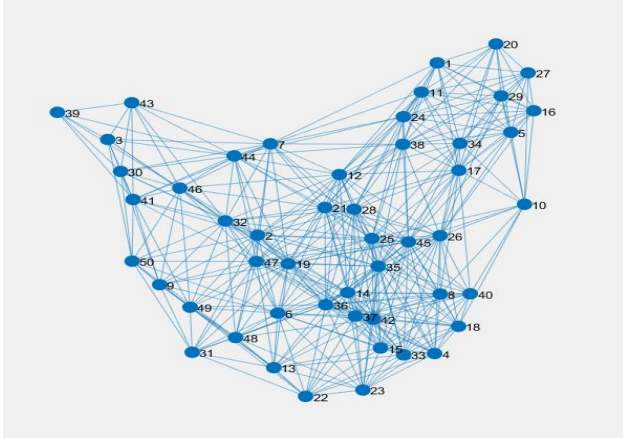


Figure 6: Network topology under varying number of IoT nodes



(a) 50 IoT nodes, $R_{max}=30$ m



(b) 50 IoT nodes, $R_{max}=40$ m

Figure 7: Network topology under varying transmission range of IoT nodes

Table 3: Simulation parameters

Parameter	Value
Monitored area	$200 \times 200 m^2$
Number of IoT nodes	50-100
Number of ECNs	3
Supply voltage	3 V
Sleep state current	0.001 mA
Idle state current	0.021 mA
Transmit state current	18.8 mA
Receive state current	17.4 mA
IoT node's initial energy	5 J
Data size	200 byte
Data rate	250 Kbps

6.2.1. ECN access cost

It is defined as the function of hop distance from IoT node to its corresponding ECN. ECN access cost is highly dependent on the location of ECNs and varies according to the placement strategies. Moreover, the $\mathcal{D}_{E2E}^P(v_i, u_j)$ (refer Eq. 3) shares the proportionality relation with ECN access cost.

6.2.2. IoT nodes mapping among ECNs

Proposed scheme uses the standard deviation (SD) as a metric to evaluate the IoT nodes mapping among ECNs. IoT nodes mapped to j^{th} ECN (u_j) is given by $|S_j|$. The SD is given by Eq. 13.

$$SD = \sqrt{\frac{\sum_{i=1}^k (|S_j| - \bar{S})^2}{k}} \quad (13)$$

Where, \bar{S} represents the average value of nodes mapped for all ECNs. Small SD indicates more balanced IoT nodes mapping among ECNs. This parameter impacts the total delay of IoT node (refer Eq. 4).

6.2. Performance metrics

coECNP scheme is evaluated with respect to the following parameters.

6.2.3. Average total delay

It is used to test the effectiveness of proposed scheme. It is characterized by Eq. 5 (see section 3.1). To achieve more realistic data processing time at ECN ($(\mathcal{D}_{pra}^{vi}(u_j))$ in Eq. 4), it is assumed that ECN is 200 times faster than IoT node and average data processing time at IoT node is considered as 400ms [16].

6.2.4. Edge resource utilization

It is used to measure the performance of proposed scheme in terms of utilization of edge resources. It is measured using the standard deviation of load distribution among ECNs. The load among ECNs must be distributed equally so as to utilize edge resources in a balanced fashion [2]. The lower SD value represents the balanced load distribution among ECNs, whereas, higher SD value represents that ECNs are underutilized/overutilized.

6.2.5. Energy consumption

Energy is a critical resource for resource constrained IoT nodes [42]. The overall energy consumption from IoT nodes to their respective ECN follow the proportionality relationship with hop distance. Hence, it can be minimized by placing ECNs at optimal location and efficiently mapping nodes among them. The total Energy Consumption (EC_{total}) in transmitting B bits between an IoT-ECN pair along path P is given as [41]:

$$EC_{total} = \sum_{i \in V(P)} (EC_{trans}^i + EC_{rcv}^i) \quad (14)$$

Where, $V(P)$ represents the set of nodes along path P and EC_{trans}^i , EC_{rcv}^i represent the energy consumed in transmission and reception of B bits at node i , respectively.

6.3. ECN access cost

This subsection discusses the ECN access cost for various ECN placement schemes under different network settings.

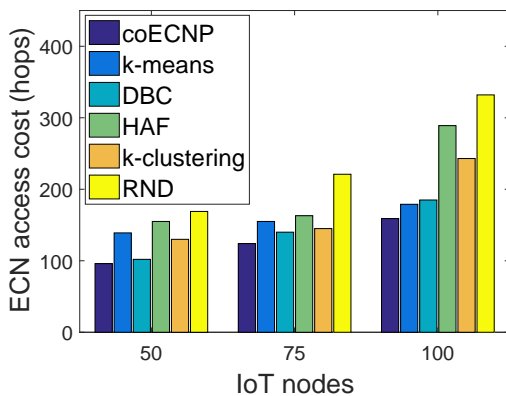


Figure 8: ECN access cost under varying IoT nodes

6.3.1. ECN access cost under varying number of IoT nodes

Fig. 8 depicts that *coECNP* scheme obtains minimum ECN access cost compared to benchmark ECN placement techniques in the network of 50, 75, and 100 nodes, respectively. *coECNP* reduces ECN access cost by 32%, 6%, 39%, and 44% compare to *k-means*, DBC, HAF, and RND schemes in the network of 50 nodes. *coECNP* adopts hop metric over distance metric for assignment of IoT nodes. It leads to the reduction of network wide ECN access cost. Therefore, *coECNP* performs better than *k-means*.

HAF selects the locations of maximum density IoT nodes to place ECNs. It does not take care of the redundancy that occurred in the density calculation. For example, in Fig. 5, HAF selects the location of highest density node's ($loc(v_4)$ and $loc(v_6)$) to place the ECNs. In this scenario, maximum ECN access cost is increased from 3 hops (in case of *coECNP*) to 8 hops. It also leads to higher network wide ECN access cost than *coECNP* scheme. In this scenario, HAF returns network wide ECN access cost as 66 hops.

DBC scheme overcomes redundancy issue occurred in HAF by removing the already assigned IoT nodes. DBC selects $loc(v_4)$ to place first ECN (u_1) (see Fig. 5). DBC evaluates the placement location for second ECN (u_2) by removing the one-hop nodes mapped to u_1 . It selects $loc(v_9)$, $loc(v_{16})$, and $loc(v_{19})$ as the candidate location to place u_2 . However, DBC takes care of the redundancy issue to determine the location to place the next ECN. Still, it did not provide a mechanism that selects the optimal location among these candidate locations. For instance, if DBC selects $loc(v_9)$ to place u_2 then it achieves network wide ECN access cost as 47 hops and maximum ECN access cost as 6 hops.

ECN_{Dis}^{Loc} module in *coECNP* scheme assigns more one-hop IoT nodes than HAF by taking care of redundancy in the process of location identification for ECN placement. ECN_{Dis}^{Loc} module also overcomes the shortcoming of DBC scheme. It determines the hop distance between candidate locations and already placed ECNs. Further, it selects the IoT node situated at farthest hop distance among candidate locations for ECN placement. *coECNP* scheme selects the $loc(v_{19})$ (location of non-redundant highest density node located at farthest hop distance from location of u_1) to place u_2 (see Fig. 5 for *coECNP* scheme). It reduces network wide ECN access cost from 47 hops (in case of DBC) to 35 hops and maximum ECN access cost from 6 hops to 3 hops. RND placement scheme places the ECN randomly that leads to higher ECN access cost than other schemes.

Further, we evaluate the *coECNP* scheme in the network of 75 and 100 nodes. In these cases, network becomes dense compare to previous scenario due to the addition of new IoT nodes. It leads to better connectivity among IoT nodes and eliminates the possibilities of longer paths from IoT nodes to corresponding ECN. In case of 75 IoT nodes, *coECNP* achieves 23%, 11%, 25%, and 45% less ECN access cost than *k-means*, DBC, HAF, and RND schemes, respectively. In dense network scenario, difference of ECN access cost between proposed scheme and DBC is increased. Moreover, in this scenario two ECN loca-

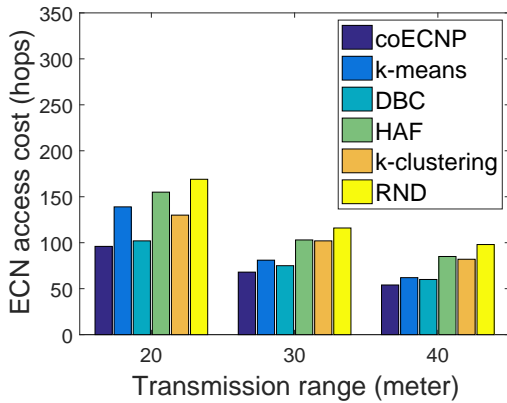


Figure 9: ECN access cost under varying transmission range

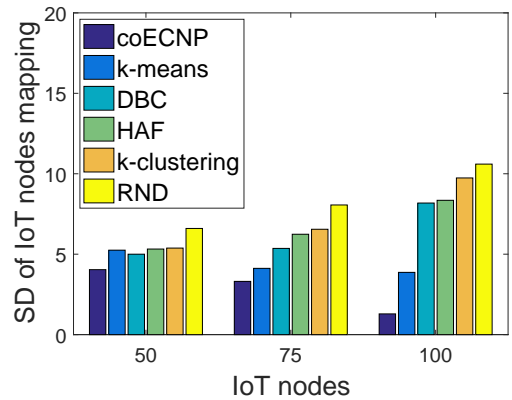


Figure 10: IoT-ECN mapping under varying IoT nodes

tions selected by HAF share zero redundant nodes. It reduces ECN access cost difference between HAF and *coECNP* compared to network of 50 nodes. Similar to the 75 nodes network scenario, 100 nodes network achieves better connectivity among IoT nodes. The *coECNP* scheme reduces the ECN access cost by 11%, 14%, 46%, and 52% than *k-means*, DBC, HAF, and RND strategies.

The above discussion indicates that ECN access cost between *coECNP* and *k-means* is reduced as the number of nodes increases in the network, due to better connectivity among nodes. As the network becomes dense, the availability of candidate locations for ECN placement also increases. Since, DBC selects a random location among the candidate ones that leads to higher access cost than *coECNP*. The difference in ECN access cost between *coECNP* and HAF is affected by redundancy factor that occurred in ECN location discovery phase.

The performance of proposed *coECNP* scheme is compared with *k-clustering* scheme [26]. The performance of *k-clustering* is highly dependent on the selection of initial cluster centers. It does not consider the redundancy and optimal cluster center selection aspects during the cluster center initialization phase that results in higher ECN access cost. *coECNP* scheme obtains 26%, 15%, and 35% lower ECN access cost compared to *k-clustering* scheme in the network of 50, 75, and 100 nodes, respectively.

6.3.2. ECN access cost under varying transmission range

coECNP is further analyzed with respect to ECN access cost under different network scenarios by varying the transmission range of IoT nodes. Fig. 9 represents that *coECNP* achieves minimum ECN access cost than benchmark mechanisms in the network of varying transmission range. *coECNP* reduces ECN access cost by 16%, 9%, 34%, and 41% than *k-means*, DBC, HAF, and RND schemes, respectively at 30m transmission range. The increase in transmission range creates new communication links among IoT nodes that enhance the connectivity among nodes than previous case (20m transmission range). It also reduces the hop distance between IoT nodes and their corresponding ECN.

Due to the better connectivity among IoT nodes, access cost

difference in proposed and *k-means* scheme is reduced. *coECNP* reduces ECN access cost by 13%, 10%, 35%, and 43% than *k-means*, DBC, HAF, and RND ECN placement strategies with 40m transmission range. It is concluded from the above discussion that *coECNP* performs better than benchmark schemes under various network settings. The difference of ECN access cost between *coECNP* and *k-means* is reduced as network becomes dense. DBC and HAF scheme follows the same trends of previous dense network scenarios (similar to the network of 75 and 100 IoT nodes).

coECNP scheme obtains 33% and 34% lower ECN access cost as compared to *k-clustering* scheme in the network of 50 IoT nodes with 30m and 40m transmission range, respectively.

6.4. IoT nodes mapping among ECNs

This subsection compares the performance of *coECNP* scheme to benchmark schemes with respect to mapping of IoT nodes among ECNs.

6.4.1. IoT nodes mapping among ECNs under varying number of nodes

Fig. 10 represents the standard deviation (SD) value of IoT nodes mapping among ECNs in the network of 50, 75, and 100 nodes for proposed *coECNP* and benchmark schemes. *coECNP* achieves minimum SD value in each network setting, hence outperforms other schemes. The mapping of IoT nodes is highly affected by the placement location of ECNs. Improper placement of ECNs causes unbalanced mapping of IoT nodes (refer Fig. 5). ECN_{Dis}^{Loc} module assures efficient placement of ECNs that enhance the chances of balanced mapping. The DBC, *k-means*, HAF, and RND schemes map nodes to the nearest ECN without taking care of already assigned nodes at that ECN that causes unbalanced mapping of IoT nodes.

coECNP considers both currently mapped IoT nodes and hop distance metric (refer IoT_{Map}^{ECN} module) to achieve balanced mapping of IoT nodes. In IoT_{Map}^{ECN} module, $EECN_{|S|}^{Min}$ mechanism categorizes the next-hop IoT nodes into unique and common nodes. $EECN_{|S|}^{Min}$ scheme maps unique and common nodes among ECNs in balanced fashion. It minimizes the difference of mapped IoT nodes among ECNs. *coECNP* achieves 24%,

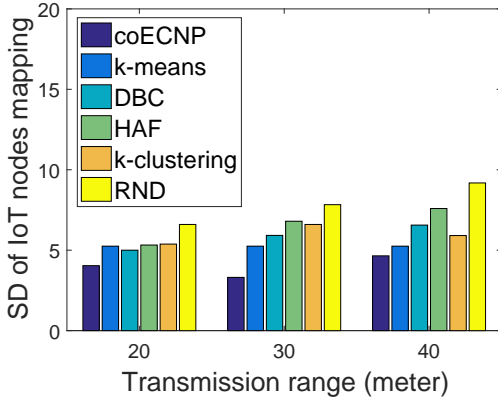


Figure 11: IoT-ECN mapping under varying transmission range

20%, 25%, and 39% more balanced mapping than k -means, DBC, HAF, and RND schemes in the network of 50 nodes. Fig. 10 indicates that as number of nodes increase the value of SD is reduced for $coECNP$. This reduction occurs due to the availability of more next-hop nodes. $coECNP$ achieves 20%, 38%, 47%, 58% and 66%, 84%, 84%, 88% more balanced mapping of IoT nodes among ECNs compared to k -means, DBC, HAF, and RND schemes in the network of 75 and 100 nodes, respectively.

The k -clustering scheme does not take up balanced IoT-ECN mapping into consideration, whereas the mapping module of $coECNP$ scheme ensures balanced IoT-ECN mapping. $coECNP$ obtains 25%, 49%, and 85% more balanced IoT-ECN mapping than k -clustering scheme in the network of 50, 75, and 100 nodes, respectively.

6.4.2. IoT nodes mapping among ECNs under varying transmission range

$coECNP$ scheme is further investigated under various network settings by varying the transmission range of IoT nodes in the network of 50 nodes. Fig. 11 depicts the SD of IoT nodes mapping for network of 50 nodes with 20m, 30m, and 40m transmission range, respectively. The SD of IoT nodes mapping among ECNs remains same for k -means scheme in each transmission range setting (due to Euclidean distance based assignment). In this case, SD values reflect less deviation compare to network scenario of varying nodes due to the less availability of next-hop nodes. $coECNP$ achieve less SD values in case of 30m transmission range than 20m due to better connectivity among nodes. $coECNP$ assigns 80% nodes in ECN_{Dis}^{Loc} module and get less next-hop nodes that increases SD value compared to previous network settings (20m and 30m) in case of 40m transmission range.

$coECNP$ obtains 37%, 44%, 52%, 58%, and 13%, 30%, 39%, 49% more balanced mapping of IoT nodes in case of 30m and 40m than k -means, DBC, HAF, and RND schemes. It can be inferred from the above results that the difference of SD values between $coECNP$ and benchmark scheme is reduced as network approaches from multi-hop (for 20m and 30m) to one-hop (case of 40m) scenario.

SD value of IoT nodes mapping among ECNs in Fig. 10 and 11 depicts that $coECNP$ scheme achieves more balanced IoT node's mapping than benchmark mechanisms under various network scenarios. SD values follow a decreasing trend in the network of varying IoT nodes. Further, SD value increases as network approach from multi-hop to one-hop scenario (case of 40m transmission range) under varying transmission range. Although, the SD value trend (increasing/decreasing/random) also depends on network topology for different scenarios. $EECN_{|S|}^{Min}$ mechanism in IoT_{Map}^{ECN} module assures that proposed scheme always achieves more balanced mapping of IoT nodes than benchmark schemes.

$coECNP$ scheme achieves 49% and 21% more balanced mapping as compared to k -clustering in the network of 50 nodes with 30m and 40m transmission range, respectively.

Under the assumption that each IoT node generates constant traffic periodically, the number of IoT nodes assigned to corresponding ECN can be treated as the load of that ECN. In such scenario, SD of IoT nodes mapping can be interpreted as SD of load among ECNs. The above discussion indicates that proposed $coECNP$ scheme achieves minimal SD. Hence, it also leads to more balanced load assignment among ECNs. Further, the balanced load assignment among ECNs leads to better edge resource utilization. Under the above-mentioned assumption $coECNP$ achieves balanced load distribution among ECNs that reduces the possibility of over/under loading of ECNs.

6.5. Scalability analysis

The scalability of a scheme is defined as the measure of its ability to maintain the network performance efficiency as some parameters of the network increases. We consider optimal scalability aspect for analysis with respect to a triplet defined as :Environment, Independent parameter(s), and Observed metric(s). The environment consists of parameters that define operational condition of the network such as network area, initial distribution of nodes etc. Independent parameters are the number of nodes and transmission range. The observed metric(s) is the quantity of interest measure in the network such as ECN access cost and IoT-ECN mapping (measured using SD of mapped nodes among ECNs) in our case. The environment initially consists of a randomly deployed network of 50 IoT nodes (refer Fig. 6a).

We discuss the optimal scalability analysis of $coECNP$ scheme with respect to benchmark schemes (HAF, DBC, k -means, k -clustering) on the observed metrics. A scheme is termed as optimally scalable with respect to a given triple if no other method is more scalable with respect to the same triple.

During the ECNs location discovery process Gateway (GW) node determines the density of every node. For a node, density is defined as the number of one-hop neighbors (see Eq. 12). GW node uses nodes location information to find the one-hop neighbors of each node. Upon finding the density of all the nodes, the location of the node with the highest density value is selected as the location for first ECN placement. This procedure of determining the location for first ECN is common for HAF, DBC, and $coECNP$ scheme. Hereafter the procedure for subsequent placement of ECNs differ for these schemes. This vari-

ation in respective mechanisms influence the observed metrics (ECN access cost and SD of mapped IoT nodes among ECNs).

The HAF scheme selects the locations for $(k - 1)$ ECNs in greedy manner based on the decreasing density values of nodes. As network moves from sparse to denser, where, density of network is given by the actual number of edges among nodes divided by the total number of possible edges in network. It implies that a denser network approaches towards maximum network density value (one). Therefore, for a denser network it is likely that the nodes will have higher node densities. In case of denser network the probability of a node having higher density is more if it is a neighbor of a high density node. Therefore, in denser networks the probability of one-hop neighbors (with respect to highest density node) having next higher density is more. This leads to close placement of ECNs that results in higher ECN access cost and unbalanced IoT-ECN mapping indicated by high SD values.

The DBC scheme excludes the highest density node and its one-hop neighbors in each iteration for selecting the next ECN location. This enhances the ECN placement location compared to HAF schemes. However, in case of multiple candidate locations with the same density value, the DBC scheme randomly selects one. In such scenario, it leads to higher ECN access cost and higher SD values for IoT-ECN mapping.

The performance of k -clustering scheme highly depends on the selection of initial cluster centers. The k -clustering scheme selects the initial cluster centers in greedy manner (as similar to HAF) and does not consider optimal cluster center selection among candidate ones (as similar to DBC) that leads to higher ECN access cost and higher SD value for IoT-ECN mapping.

The coECNP scheme selects the optimal location for ECN placement in each iteration by determining the hop distance between ECN locations of previous iterations and current iterations candidate locations. It selects the next ECN location among the candidates having maximum hop distance from the ECN locations of previous iterations. The coECNP scheme performs better compared to HAF and DBC in terms of optimal placement of ECNs thus better ECN access cost and balanced IoT-ECN mapping (indicated by the lower SD values).

From the above discussion we conclude that coECNP scheme performs better when network scale up. The HAF and DBC schemes does not scale well with respect to ECN access cost and IoT-ECN mapping and an increasing trend is seen in the observed metrics (see Fig. 8 and Fig. 10). The proposed coECNP shows a significant improvement with respect to HAF and DBC schemes, and marginal with respect to k -means scheme. Thus, the coECNP scheme overallly scales well with denser networks in terms of the observed metrics.

6.6. Average total delay

This subsection discusses the performance of *coECNP* scheme with benchmark schemes in terms of average total delay (see Eq. 5) of IoT nodes. The total delay depends on ECN placement and mapping of IoT nodes among ECNs (refer section 3). It is evident from the above discussion that proposed *coECNP* scheme achieves optimal ECN placement (see section

6.3) and balanced mapping of IoT nodes (see section 6.4) compared to benchmark schemes.

This work evaluates the performance of *coECNP* scheme in an industrial use case scenario. Edge computing solutions enhance overall operational efficiency in industrial use case(s) by minimizing the delay. The data traffic pattern between IoT nodes and ECNs varies as per application requirements. These data traffic patterns are divided into three categories as (i) periodic data traffic, (ii) event-based data traffic, and (iii) query-based data traffic [42].

IoT nodes transmit data to respective ECN over a fixed interval in periodic data traffic scenario. In case of event-based data traffic scenario, IoT nodes of event-region(s) transmit more data traffic than normal nodes. Whereas, IoT nodes transmit data based on the demand of end-user(s) in query-based scenario. For instance, in an industrial equipment health monitoring application, IoT nodes (such as sensors installed on machine equipment) transmit data to respective ECN (according to the above-mentioned data traffic pattern(s)). ECN processes the received data to keep track of the current health status of machine and take decision(s) accordingly that ensures overall operational efficiency.

We conduct the experiments to test the performance of *coECNP* scheme in terms of IoT nodes delay under various network settings on periodic, event-based, and query-based data traffic scenarios. IoT nodes transmit data to respective ECN over an interval of one second in periodic data traffic scenario. Event and query based data traffic scenarios are generated by varying the data generation interval at IoT nodes (nodes having lower data generation interval indicates the event traffic) and amount of data transmitted from IoT nodes (in query-based scenario end user can demand varying amount of data from different IoT nodes).

Fig. 12 depicts an instance of event and query based traffic scenarios for the network of 100 IoT nodes. Fig. 13 depicts IoT node's average total delay under periodic, event-driven, and query-based scenarios for the network of 50, 75, and 100 nodes, respectively. *coECNP* scheme achieves minimum delay in each network setting for each data traffic scenario compared to benchmark schemes. *coECNP* scheme determines the optimal ECN placement locations and assures efficient mapping of nodes among ECNs using location discovery and mapping modules. The location discovery module obtains the optimal locations for ECN placement that minimizes the maximum hop-distance between IoT node and respective ECN (see section 6.3). As a result, the E2E delay between IoT node-ECN pair is minimized. The mapping module maps the IoT nodes to the nearest ECN and minimizes the difference of mapped IoT nodes among ECNs (see section 6.4). It alleviates E2E delay between IoT node and respective ECN, and data processing delay at that ECN. Thus, proposed *coECNP* scheme leads to total delay minimization for IoT nodes than its counter part schemes.

coECNP reduces the average total delay by 53%, 11%, 69%, 71%, and 37%, 23%, 42%, 66%, and 23%, 26%, 53%, 65% compared to k -means, DBC, HAF, and RND schemes in the network of 50, 75, and 100 IoT nodes, respectively. In case of event-based data traffic, *coECNP* achieves 45%, 11%, 75%,

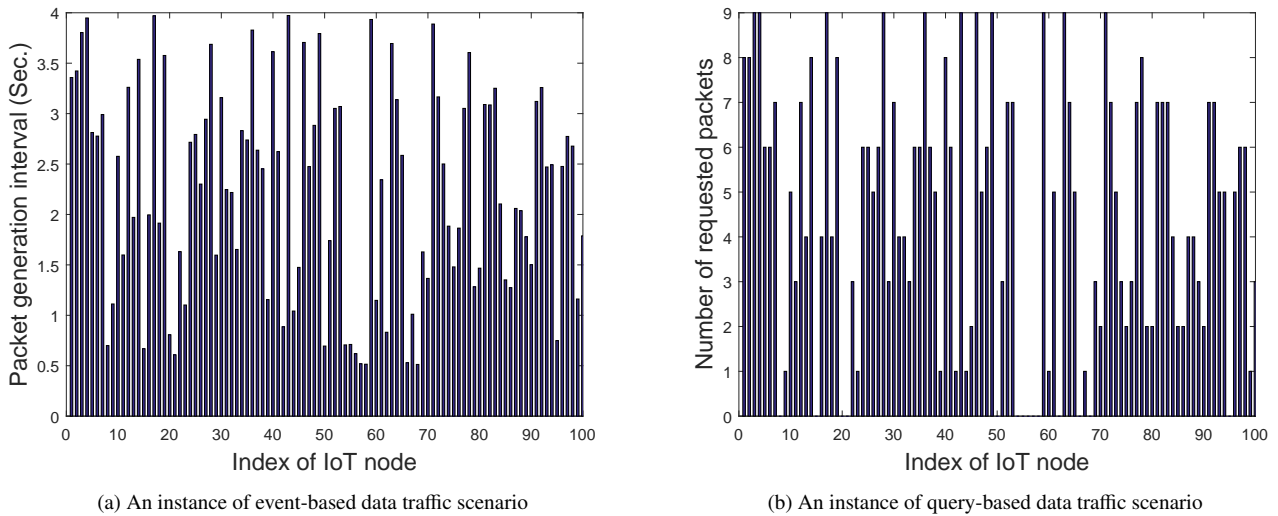


Figure 12: Sample scenario of event and query based data traffic

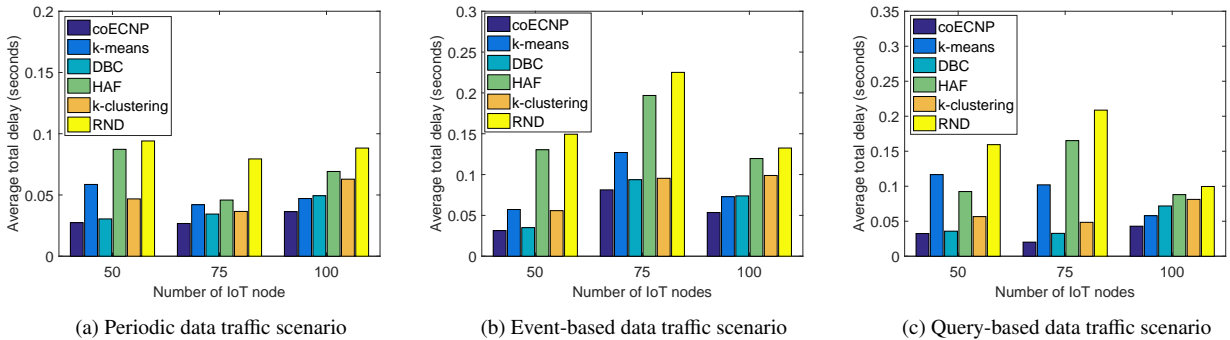


Figure 13: IoT nodes' average total delay under periodic, event, and query based data traffic scenarios

78%, and 36%, 13%, 59%, 64%, and 36%, 25%, 56%, 61% average total delay reduction than k -means, DBC, HAF, and RND schemes for the network of 50, 75, and 100 nodes, respectively. $coECNP$ reduces the average total delay by 72%, 10%, 64%, 86%, and 80%, 40%, 87%, 90%, and 26%, 40%, 51%, 57% in comparison to k -means, DBC, HAF, and RND schemes under query-based data traffic in the network of 50, 75, and 100 nodes, respectively.

We compare the performance of $coECNP$ scheme with k -clustering in the network of 50, 75, and 100 nodes on periodic, event-based, and query-based data traffic, respectively. $coECNP$ scheme reduces the average total delay by 41%, 27%, 42% for periodic data traffic. It obtains 44%, 15%, 46% lower average total delay than k -clustering under event-based data traffic. In case of query-based data traffic scenario, $coECNP$ scheme reduces average total delay by 43%, 55%, 47%, respectively.

$coECNP$ assumes that each event/query has same priority and probability of occurrence of event/query at each IoT node is equal. However, application-specific requirements (such as temporal requirement based on event/query type) can be accomplished by designing the suitable task(event/query) scheduling mechanisms over the proposed EIaaS paradigm. The schedul-

ing involves the decision making procedure about how data should be transmitted, and how edge resources are utilized by taking care of event/query types (such as, critical task, normal task, query with strict temporal requirements etc.) in order to meet the application-specific requirements [2].

6.7. Edge resource utilization

This work utilizes the standard deviation of load distribution among ECNs as an indicator to measure the efficacy of $coECNP$ scheme in terms of edge resource utilization. The smaller SD value indicates the balanced edge resource utilization, whereas, higher SD value represents the over/under utilization of edge resources. Fig. 14 depicts the standard deviation of load distribution among ECNs for $coECNP$, k -means, DBC, HAF, and RND schemes in various network settings under different data traffic scenarios.

$coECNP$ scheme obtains minimum SD values of ECNs load distribution than benchmarks schemes in periodic traffic scenario. It indicates that proposed scheme utilizes the edge resources in balanced fashion. Each IoT node transmit same amount of data periodically to respective ECN in periodic traffic scenario. On the other hand, in event/query based scenario data traffic variation among ECNs depends on various factors

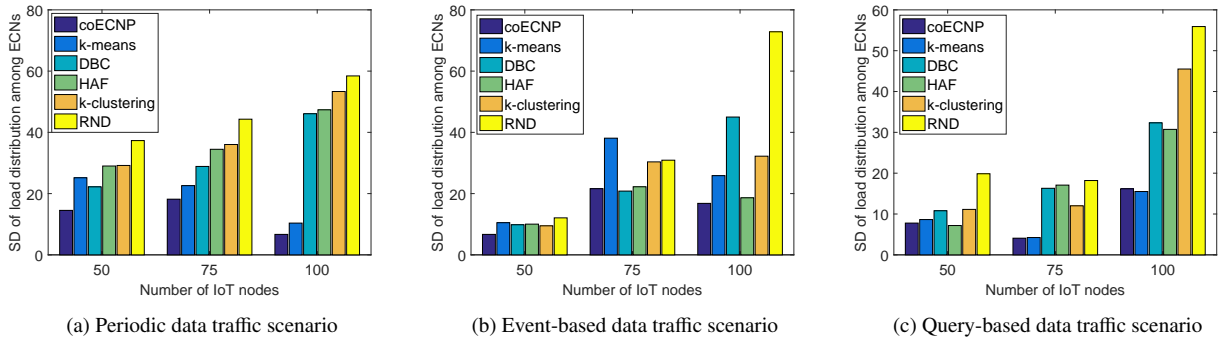


Figure 14: SD of ECNs' load distribution under periodic, event, and query based data traffic scenarios

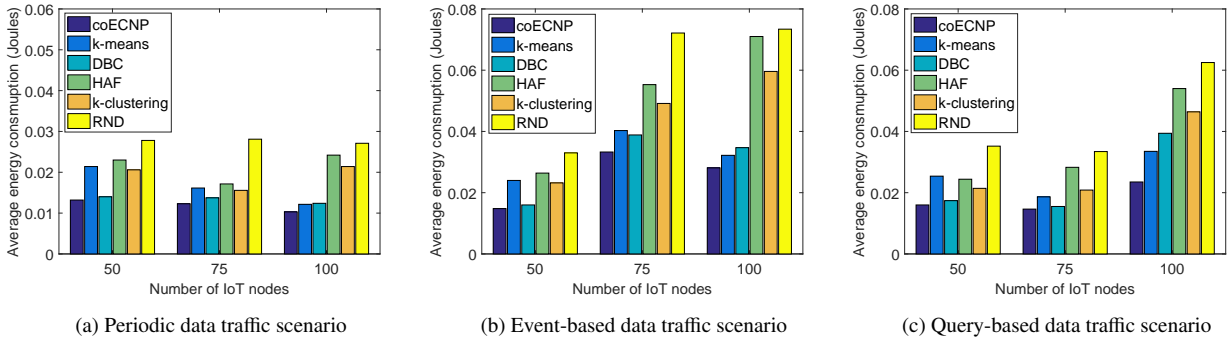


Figure 15: IoT nodes energy consumption under periodic, event, and query based data traffic scenarios

such as number of IoT nodes in event/query region, amount of transmitted data etc. *coECNP* scheme also achieves better or approximately equal (see *coENCP* and DBC schemes SD values in Fig. 15b, and *coENCP* and *k-means* schemes SD values in Fig. 15c) edge resource utilization due to optimal mapping of nodes. Data traffic in event/query based scenario does not reflect the consistent pattern as a periodic traffic scenario. Therefore, it requires an efficient inter-ECN communication mechanism to achieve balanced load distribution among ECNs.

The mapping module of *coECNP* scheme takes care of balanced IoT-ECN mapping that leads to balanced utilization of edge resources as compared to *k-clustering* scheme on different data traffic patterns under various network settings.

6.8. Energy consumption

Energy is a critical resource for resource constrained IoT nodes [42]. This subsection discusses the energy consumption of IoT nodes for various ECN placement schemes in various network settings under various data traffic scenarios. Energy consumption of IoT node depends on the number of hops between IoT node and respective ECN that varies according to ECN placement and mapping schemes. *coENCP* scheme performs optimal ECN placement (see section 6.3) and balanced mapping (see section 6.4) that minimizes the energy consumption of IoT nodes compared to benchmark schemes.

Fig. 15 depicts the average energy consumption of IoT nodes under various data traffic scenarios in the network of 50, 75, and 100 nodes for *coECNP* and benchmark schemes, respectively. *coECNP* achieves minimum energy consumption among

various ECN placement schemes due to optimal ECN placement and efficient mapping of nodes among ECNs. *coECNP* achieves 38%, 6%, 43%, 53%, and 24%, 11%, 28%, 56%, and 15%, 17%, 57%, 62% less energy consumption compared to *k-means*, DBC, HAF, and RND schemes in the network of 50, 75, and 100 nodes under periodic data traffic, respectively. In case of event-based data traffic, *coECNP* scheme reduces energy consumption by 38%, 11%, 44%, 55%, and 18%, 14%, 38%, 54%, and 13%, 19%, 60%, 62% than *k-means*, DBC, HAF, and RND schemes for the network of 50, 75, and 100 nodes, respectively. Further, *coECNP* reduces the energy consumption by 37%, 9%, 28%, 55%, and 21%, 8%, 48%, 56%, and 29%, 40%, 56%, 62% in compared to *k-means*, DBC, HAF, and RND schemes for query based traffic in the network of 50, 75, and 100 IoT nodes, respectively.

Fig. 15 depicts that *coECNP* scheme obtains lower energy consumption in comparison to *k-clustering* in the network of 50, 75, 100 nodes under different data traffic patterns. *coECNP* scheme reduces energy consumption by 36%, 21%, 52% for periodic data traffic. *coECNP* obtains 36%, 33%, 53%, and 22%, 30%, 49% lower energy consumption as compared to *k-clustering* scheme on event and query-based data traffic, respectively.

7. Theoretical Analysis

This section represents the theoretical analysis of proposed *coECNP* scheme with respect to various network centrality

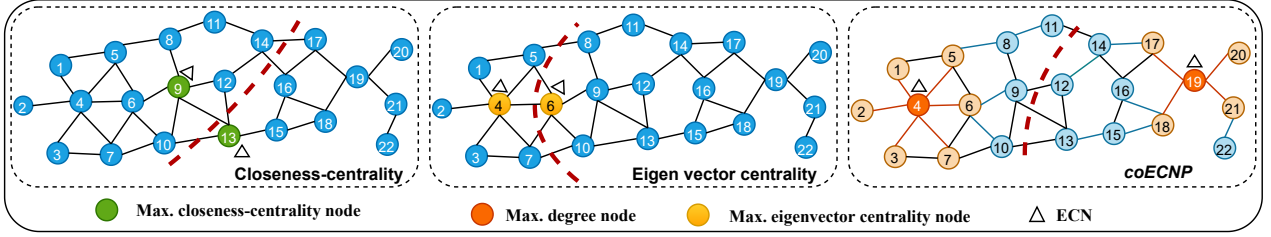


Figure 16: ECN placement under different centrality measure schemes

measures to indicate the efficacy of *coECNP* schemes. This work adopts the degree centrality measure with certain extensions (refer section 4.1) to identify the optimal ECN placement locations. Closeness centrality measure defines the closeness of a node to all other nodes in the network. On the other hand, eigenvector centrality measures the importance of node by considering the degree of both node and its neighbors.

Lemma 2. *coECNP* scheme identifies optimal ECN placement locations compared to closeness-centrality scheme.

Proof. The results in Fig. 16 indicates that the nodes located at the center of graph have more closeness centrality than other nodes. Closeness-centrality based scheme selects k nodes having maximum closeness centrality values (in descending order) for ECN placement as $n_{\text{closenessCen}}^{\{i|i=1,2,\dots,k\}}$ in the network of n nodes. On the other hand, *coECNP* iteratively selects the location of k nodes ($n_{\text{coECNP}}^{\{i|i=1,2,\dots,k\}}$) as optimal ECN placement locations that have maximum density and satisfy $\bigcap_{i=1}^k n_{\text{coECNP}}^{\{i|i=1,2,\dots,k\}} = \phi$. Closeness centrality scheme only consider the distance measure in ECN placement, whereas, *coECNP* maximizes the number of nodes that access respective ECN with minimum distance. Further, it also takes care of appropriate mapping of IoT nodes among ECNs taking hop distance and balanced assignment into consideration for delay minimization. \square

Lemma 3. *coECNP* scheme maximize the number of nodes that access respective ECN with in one-hop compared to eigenvector centrality scheme.

Proof. Given the network of n nodes, assume *coECNP* and eigenvector centrality based scheme assigns $n_{\text{coECNP}}^{1\text{-hop}}$ and $n_{\text{eigenCen}}^{1\text{-hop}}$ nodes as 1-hop neighbors to ECNs. $n_{\text{coECNP}}^{1\text{-hop}}$ is the union of one-hop nodes assigned to different ECNs i.e. $n_{\text{coECNP}}^{1\text{-hop}} = \bigcup_{i=1}^k n_i^{1\text{-hop}}$, where, n_i is the set of one-hop nodes associated to i^{th} ECN. Further, *coECNP* scheme takes redundancy issue into account in identification of maximum density nodes i.e. $\bigcap_{i=1}^k n_i^{1\text{-hop}} = \phi$. However, eigenvector centrality scheme selects ECN placement locations by considering density of nodes and its neighbors but did not considers redundancy issue that leads to $\{(n - n_{\text{eigenCen}}^{1\text{-hop}}) > (n - n_{\text{coECNP}}^{1\text{-hop}})\}$. Further, ECN placement without addressing redundancy issue also leads to unbalanced nodes assignment among ECNs (refer eigenvector centrality scheme in Fig. 16). Therefore, it can be concluded that ECN placement location determined by *coECNP* maximize the number of nodes that access respective ECN with in one-hop to minimize delay. \square

Lemma 4. *coECNP* scheme determines ECN placement location $\left(\frac{n}{k}\right)$ times faster than closeness centrality and eigenvector centrality schemes.

Proof. *coECNP* schemes takes $O(kn^2)$ computational time to determine the optimal ECN placement locations in the network of n nodes. Whereas, closeness centrality and eigenvector centrality schemes identify the optimal locations for ECN placement in $O(n^3)$. It indicates that proposed scheme achieves optimal results with $\left(\frac{n}{k}\right)$ times reduced computational cost. \square

8. Conclusion

This work investigates the role of ECN placement and mapping of IoT nodes among ECNs for topology formation in ElaaS paradigm to serve latency-sensitive applications. We propose a solution, *coECNP*, that collectively addresses ECN placement and IoT nodes mapping issues to minimize the delay of IoT nodes and enhance edge resource utilization in order to accomplish an optimal topology formation in ElaaS paradigm. *coECNP* consists of three modules. The location discovery function in ECN_{Dis}^{Loc} module selects the optimal locations for ECNs placement to minimize E2E delay between IoT node and respective ECN with $O(kn^2)$ computational cost. Further, we propose $ECN_{|S|}^{Min}$ and $EECN_{|S|}^{Min}$ mapping schemes under IoT_{Map}^{ECN} module that minimizes the difference of mapped IoT nodes among ECNs with computational cost of $O(h_{\max}(k \log k))$. These schemes minimize the total delay of IoT nodes and enhance edge resource utilization. Finally, $ElaaS_{conf}$ module accomplishes the ElaaS topology formation by utilizing the information received from ECN_{Dis}^{Loc} and IoT_{Map}^{ECN} modules. We conduct the simulation experiments to test the effectiveness of *coECNP* scheme with benchmark schemes. Simulation results indicate that *coECNP* outperforms benchmark schemes in terms of average total delay, energy consumption, and edge resource utilization on various data traffic scenarios under various network settings.

Future work includes the extension of *coECNP* scheme in ElaaS paradigm by designing efficient mechanism(s) such as an intelligent inter-ECN coordination mechanism that maximizes balanced utilization of edge resources and satisfies application-specific requirements.

References

- [1] T. Zhang, Y. Li, and C. L. P. Chen, "Edge computing and its role in Industrial Internet: Methodologies, applications, and future directions," *Information Sciences*, vol. 557, pp. 34–65, 2021.
- [2] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.
- [3] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238–78259, 2018.
- [4] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [6] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
- [7] C. Li, Y. Xue, J. Wang, W. Zhang, and T. Li, "Edge-oriented computing paradigms: A survey on architecture design and system management," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–34, 2018.
- [8] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.
- [9] L. C. Mitch Tseng Todd Edmunds, "Introduction to edge computing in IIoT." <https://www.iiconsortium.org/white-papers.htm>, 2018.
- [10] M. Wopata, "IIoT edge computing what it is and how it is becoming more intelligent." <https://iot-analytics.com/iiot-edge-computing-what-it-is-and-how-it-is-becoming-more-intelligent/>, 2020.
- [11] R. Candell, M. Hany, K. Lee, Y. Liu, J. Quimby, and C. Remley, "Guide to Industrial Wireless Systems Deployments," 2018.
- [12] 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.
- [13] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.
- [14] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proc. of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019.
- [15] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.
- [16] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.
- [17] X. Chen, Q. Shi, L. Yang, and J. Xu, "ThriftyEdge: Resource-efficient edge computing for intelligent IoT applications," *IEEE network*, vol. 32, no. 1, pp. 61–65, 2018.
- [18] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekkilä, and M. J. Sillanpää, "Edge computing server placement with capacitated location allocation," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 130–149, 2021.
- [19] I. Gravalos, P. Makris, K. Christodoulouopoulos, and E. A. Varvarigos, "Efficient gateways placement for Internet of Things with QoS constraints," in *2016 IEEE Global Comm. Conf.*, pp. 1–6, IEEE, 2016.
- [20] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International Conference on Edge Computing (EDGE)*, pp. 66–73, IEEE, 2018.
- [21] B. Li, K. Wang, D. Xue, and Y. Pei, "K-means based edge server deployment algorithm for edge computing environments," in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, pp. 1169–1174, IEEE, 2018.
- [22] J. Meng, W. Shi, H. Tan, and X. Li, "Cloudlet placement and minimum-delay routing in cloudlet computing," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 297–304, IEEE, 2017.
- [23] F. Zeng, Y. Ren, X. Deng, and W. Li, "Cost-effective edge server placement in wireless metropolitan area networks," *Sensors (Switzerland)*, vol. 19, no. 1, 2019.
- [24] S. Yang, S., Li, F., Shen, M., Chen, X., Fu, X., & Wang, Y. (2019). Cloudlet placement and task allocation in mobile edge computing. *IEEE Internet of Things Journal*, 6(3), 58535863. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.
- [25] Z. Wang, F. Gao, and X. Jin, "Optimal deployment of cloudlets based on cost and latency in Internet of Things networks," *Wireless Networks*, vol. 26, no. 8, pp. 6077–6093, 2020.
- [26] T. Chin, Y. Chen, and K. Lyu, "Queueing Model Based Edge Placement for Work Offloading in Mobile Cloud Networks," *IEEE Access*, vol. 8, pp. 47295–47303, 2020.
- [27] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 926–937, 2019.
- [28] T.-Y. Liang and Y.-J. Li, "A location-aware service deployment algorithm based on k-means for cloudlets," *Mobile Info. Systems*, 2017.
- [29] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.
- [30] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.
- [31] M. Jia, J. Cao, and W. Liang, "Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks," *IEEE Transactions on Cloud Computing*, no. 4, pp. 725–737, 2017.
- [32] D. Kumar and R. B. Patel, "Multi-hop data communication algorithm for clustered wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 7, no. 1, p. 984795, 2011.
- [33] S. Veeramani and S. N. Mahammad, "An Approach to Place Sink Node in a Wireless Sensor Network (WSN)," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1117–1127, 2020.
- [34] Z. Peng, R. Daraei, S. M. Ahmadpanahi, A. S. Danesh, S. Siadat, P. Pirozmand, and R. J. Oskouei, "Proposing a Density-Based Clustering Approach (DBCA) to Aggregate Data Collected from the Environment in Arid Area for Desertification," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.
- [35] J. Wang, D. Li, and Y. Hu, "Fog Nodes Deployment Based on Space-Time Characteristics in Smart Factory," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3534–3543, 2021.
- [36] Z. Wang, W. Zhang, X. Jin, Y. Huang, and C. Lu, "An optimal edge server placement approach for cost reduction and load balancing in intelligent manufacturing," *The Journal of Supercomputing*, 2021.
- [37] S. K. Kasi, M. K. Kasi, K. Ali, M. Raza, H. Afzal, A. Lasebae, B. Naeem, S. ul Islam, and J. J. P. C. Rodrigues, "Heuristic Edge Server Placement in Industrial Internet of Things and Cellular Networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10308–10317, 2021.
- [38] D. Marutho, S. H. Handaka, E. Wijaya, and Others, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," in *2018 Int. Seminar on Application for Tech. of Information and Communication*, pp. 533–538, IEEE, 2018.
- [39] L. Lovén, T. Lähderanta, L. Ruha, T. Leppänen, E. Peltonen, J. Riekkilä, and M. J. Sillanpää, "Scaling up an Edge Server Deployment," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 1–7, 2020.
- [40] Sapna, K. K. Pattanaik, and A. Trivedi, "A dynamic distributed boundary node detection algorithm for management zone delineation in Precision Agriculture," *Journal of Network and Computer Applications*, vol. 167, p. 102712, 2020.
- [41] R. K. Verma, K. K. Pattanaik, S. Bharti, and D. Saxena, "In-network context inference in IoT sensory environment for efficient network resource utilization," *Journal of Network and Computer Applications*, vol. 130, pp. 89–103, 2019.
- [42] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.