2022 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Optimal Node Selection in Communication and Computation Converged IoT Network

Rolden Fereira[1], Chathurika Ranaweera[1], Jean-Guy Schneider[2], Kevin Lee[1]
[1]*School of Information Technology, Deakin University, Geelong, VIC 3220, Australia.*
[2]*Faculty of Information Technology, Monash University, Clayton, VIC 3168, Australia.*
rjfereira@deakin.edu.au

***Abstract*—In Internet of Things (IoT), the things collect, relay information, and processes the information collectively and take self-automated actions. With growing complexities in the IoT domain and its architectures, a convergence of computation and communication technologies is becoming a key challenge to meet the stringent demands of advanced IoT use cases. An exponential increase in IoT devices and stringent communication and network constraints such as latency and bandwidth from advanced IoT use cases, including autonomous vehicles, eHealth, and smart grid, make it challenging to use the current IoT infrastructure. Further, most IoT architectures proposed so far focus on a single IoT use case, with one dimension on communication and the other on computation architectures. However, in emerging IoT networks, all computation layers and multiple IoT use cases must be supported in a single IoT architecture to tackle the exponential growth in IoT applications cost-effectively and energy-efficiently. To address this challenge, we propose an optimal node selection framework that considers all three computation layers (edge, fog, and cloud) for load balancing and optimizing resource allocations in an IoT architecture. The proposed approach is evaluated through simulation results. The results provide an insight into how the proposed framework can be used to allocate the best suitable node in the IoT architecture and process the requests whilst using a minimal number of nodes in the architecture and satisfying the network and application requirements.**

I. Introduction

Ongoing developments and innovations in computation, networking, and communication are reshaping the internet of things (IoT) landscape. With a prediction of an exponential increase of IoT devices to 500 billion by 2030, current wireless network technologies and cloud computation capacity will fall short of meeting the stringent and varying requirements of diverse IoT application use cases [1]. Even though the fifth generation (5G) wireless communication, cloud computing, and fog computing are becoming integral parts of diverse IoT applications, many challenges are yet to be tackled to gain the full benefits these technologies have to offer in heterogeneous, decentralized interconnected IoT networks.

In recent years, researchers have focused on proposing an IoT architecture for a single application use case and its different sub-applications. The benefits of the architectures have been evaluated based on the quality of service (QoS) metrics, task offloading, and energy efficiency [2]. Recent research has been focused on issues and hurdles faced via communication network providers in meeting the demands of network and application constraints [3], [4]. To achieve the full potential of IoT and provide end-to-end communication to support required QoS, efficient caching at different layers and employing distributed computation needs further investigation.

In particular, there is a need for a generalized IoT framework that can be used irrespective of the application use case that considers both application and network requirements. Previous investigations focused on a single use case and a limited number of quality of service metrics such as delay, bandwidth, and energy consumption. To address these challenges and meet the stringent requirements of emerging IoT applications, we must enable full convergence of communication and computation technologies. This can be achieved by developing intelligent frameworks and paradigms that have considered these technologies at the design and operational phase of IoT frameworks and architectures.

This paper considers a distributed IoT architecture comprised of cloud, fog, and edge layers. We also provide a brief introduction to various emerging technologies in communication, computation, and IoT architecture, highlighting the importance of each technology from an IoT architecture perspective. We then investigate a mechanism that can be used in such a versatile architecture to efficiently serve diverse IoT applications ranging from energy monitoring to healthcare.

The key contributions of this paper are 1) the investigation of an IoT architecture with the usage of fog, cloud, and edge layers 2) the proposal of an optimal node selection mechanism based on Integer Linear Programming(ILP) to optimally select a computation node for processing various IoT applications while minimizing the usage of resources in the entire IoT network architecture 3) provides insights into the development of a generalized framework that can be used to guarantee the various requirements of IoT applications and communication networks.

The remainder of this paper is structured as follows. First, we provide a literature review in Section II focusing on IoT architectures, load balancing and node selection frameworks, and emerging IoT applications. In Section III, we elaborate on the opportunities identified from the literature review and our prime focus of this paper. In Section IV, we discuss the IoT network architecture in consideration, followed by a detailed description of the mathematical formulation of the proposed optimal node selection framework. The evaluation of the proposed framework is presented in Section V. Finally, the paper concludes in Section VI.

978-1-6654-6497-0/22/$31.00 ©2022 IEEE

DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00075

II. Literature Review

IoT has adopted cloud, fog, and edge computing for data processing and storage. A typical IoT architecture can consist of one or two of these layers, depending on IoT applications' various data processing and storage needs. For example, IoT applications that require low-latency communication, such as Telehealth, need to be processed closer to the user at the edge layer to satisfy the application requirements.

Figure 1 illustrates different computation layers that IoT architectures can use. As shown in the figure, at the edge layer, edge nodes with processing capability are deployed near the base station to process the data in the vicinity of the data sources and end-users [5], [6]. As the edge nodes are closer to the users, this architecture can minimize the latency, save network bandwidth, and provide more secure network connectivity. On the other hand, the fog layer consists of light processing nodes comprising of modern virtualized and scalable platforms for computation management, network administration, and storage services. The fog nodes can also be deployed closer to the user where more space is available such as central offices [6]. Most IoT applications we use today utilize only the cloud layer. The cloud layer consists of nodes enabled with distributed computing using pooled, virtualized, and scalable resources, with managed and controlled computing power, complimented by scalable storage and flexible services widely spread over the internet [6]. Depending on different computation technologies, caching mechanisms used and types of IoT gateways, and IoT devices connected, different IoT architectures and frameworks have been proposed for their optimal operations. We briefly discuss each of those research works in the following sub-sections.

A. IoT Architectures

IoT layered architectures enable the usage of advanced computation technologies, thus enabling us to manage the resources more feasibly and efficiently. Each layer of the IoT network architecture helps handle and optimize IoT requests.

Edge computing-based architectures emphasize IoT architectures wherein processing is done very close to the end user. The content delivery network is one of the proposed edge computing solutions, which includes deploying and distributing many servers at various geographic locations to increase caching and reliability [7]. Many edge computing solutions like amazon Greengrass, IBM Watson IoT, Cisco edge, Microsoft Azure IoT Edge, and SAP Leonardo Edge Services have been recommended to manage IoT applications for efficient processing and caching. Edge computing helps reduce operational costs, latency, energy efficiency, near real-time data analysis, and network load.

Fog computing-based IoT architectures focus on implementing a computation layer near the edge layer and collating the fog nodes for faster computation and better security. The energy-aware fog computing techniques in IoT were also investigated. For example, in [8], a mechanism to offload the computations tasks to enhance energy efficiency has been proposed. Wherein energy-intensive computation task is offloaded

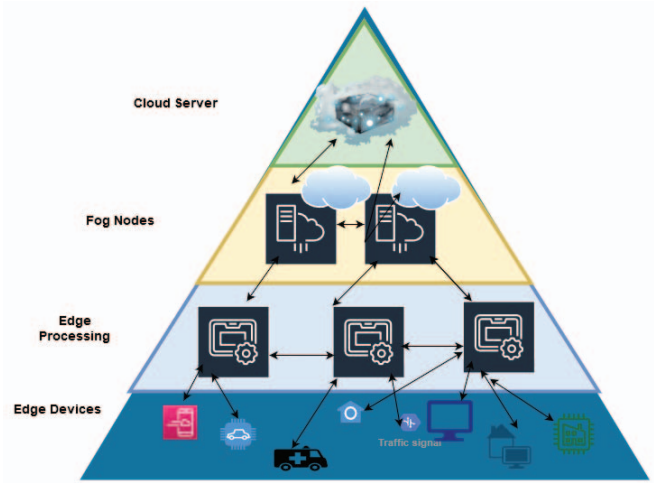

Fig. 1. IoT Architecture

to under-utilized helper nodes at the network level. The task helper nodes are structured into either clustered architecture, centralized architecture, or distributed architecture, based on the location of the fog node and resource allocation [9], [10].

Reliability plays an important role when deciding the infrastructure for an IoT architecture for a specific IoT application use case. For example, the reliability investigation of different architectures was presented in which the suitability of a reliable region for processing the data in an IoT architecture is tested using a reliability-based network framework for optimizing target-based performance [11], [12]. Further, the edge computing service reliability was also investigated using Markov chain-based method to achieve a balance between network operating cost and multi-access edge computation [13]. In mobile edge computing, task computation and task offloading problems have been addressed using power minimization techniques for the entire network considering the delay and non-reliability [14]. The usage of optimization frameworks for minimizing the offloading failure probability was also investigated in the literature [15], [16]. The reliability of fog computing for smart mobile applications has also been investigated using techniques such as anomaly detection and probability functions for successful transmission against predefined threshold value [17]–[19].

B. Load balancing and node selection frameworks

Due to the growing number of IoT devices and increasingly stringent demands of IoT use cases, it is challenging to manage the IoT architecture and its resources optimally. In this subsection, we review the previous research that focused on load balancing, node selection, and optimization techniques to overcome the challenge of efficient resource utilization.

Since the amount of computation and communication resources is limited, it is necessary to distribute the tasks among different resources. To balance the load on various resources, we require efficient resource offloading and sharing mechanisms to achieve the optimal utilization of resources

and optimal performance. In [20], authors investigated the use of classic optimization techniques for load balancing, such as ant colony optimization (ACO) and particle swarm optimization (PSO). The use of network function virtualization, software-defined networking, and network slicing in a mobile network for load balancing and improving the availability and traffic offloading was investigated in [21]. Further, mobile resource-sharing frameworks on mobile edge servers have been explored to share the edge resources among multiple IoT devices in [22]. Energy saving and task offloading techniques by switching on/off of devices were also investigated in [23], [24]. These frameworks focused on load balancing and resource sharing on a single layer. However, the resource utilization and load balancing considering the usage of all three layers needs further investigation.

Node selection mechanisms in an IoT architecture play a crucial role in resource allocation and guaranteeing the QoS requirements of end users. In [25], authors proposed a K-means-based optimization framework for fog node placement and forgy method, mid-point method, sorted cluster mid-point, and partition mid-point methods for fog node selection. This optimized number of fog node deployments helps in reducing the latency. The fog node selection methods based on random selection, shortest estimated buffer, profit function, and shortest estimated latency have been considered for improving energy consumption, packet loss, and increasing hit ratio at fog nodes [26]. Moreover, an active node selection mechanism has been investigated in IoT-based sensing applications in the fog layer using genetic algorithms and greedy selection mechanisms [27]. A task allocation based on clustering and data aggregation techniques is used for efficient node selection [28].

Dense deployment of fog nodes with a combination of unsupervised machine learning using integration of k-means clustering and Principal Component Analysis (PCA) fog computing design can be done. K-means clustering method is used for selecting an accurate optimal fog node, and PCA is used for detecting signal changes on each sub-channel and abnormal interference at fog nodes [29]. From the above literature review, it can be observed that most of the research is emphasized using only one layer for processing the data in the close vicinity of the end user in an IoT architecture. With emerging modern IoT applications and their stringent QoS and computation requirements, it is inevitable to use all three layers in an IoT architecture efficiently. This is the most challenging part, in addition to optimal node selection at these layers for processing the request.

C. Network and Application QoS Constraints

In emerging IoT applications such as smart homes, intelligent transportation systems, smart cities, smart health care, and Industry 4.0/5.0, we need to consider the design requirements, including low device cost, low deployment cost, long battery life, extended coverage, security, privacy, and support for the massive number of devices. IoT applications are supported through various communication technologies ranging from low-range wireless networks to wide-area wireless networks [30], [31]. The latest communication technologies, such as 5G, would be able to support ultra-reliable low latency communication (uRLLC), enhanced mobile broadband (eMBB), and machine type communication (eMTC) to satisfy the requirements of IoT applications [32]. However, with growing machine-type communication(MTC), the IoT network infrastructure still faces challenges such as scalability, network management, inter-operability, and heterogeneity [33], [34].

To overcome the challenges faced by the communication network, we need to introduce the full convergence of communication and computation technologies. Different IoT applications have additional requirements. For example, applications such as autonomous vehicles require fast processing of videos with higher data rates (between 512 Gbps-1024 Gbps) closer to the end-users while communicating control messages with minimal latency in a few milliseconds. The eHealth applications consist of sub-applications like telehealth and telesurgery, which require a high data rate(5-512Gbps). Telehealth applications do not have ultra-low latency requirements. However, telesurgery applications that involve remote robotic surgeries do require very low latency in the range of a few milliseconds. Even smart grid applications consist of various sub-applications like advanced metering infrastructure, synchrophasor applications, and supervisory control and data acquisition (SCADA) applications, which require data transfer rates in the range of 5-75Mbps and latency in the range of 1-200 milliseconds. Though 5G would be able to meet the stringent requirements of most IoT applications, the key challenge in the future would be incorporating the growing number of IoT devices and stringent IoT application requirements. Hence we need to consider the convergence of communication, computation, and caching using advanced technologies like the sixth generation (6G) wireless network, network slicing, virtualization, and load balancing.

III. Research Opportunities and Focus

Through the literature review, we identified the research opportunities to support and enhance emerging IoT applications through the convergence of communication and computation technologies. This section discusses these opportunities, followed by the motivational focus of the work presented in this paper.

Most IoT architectures are designed from an end-user and application layer perspective considering a single type of user case which would not be feasible for large-scale deployments. This is mainly because of scalability, flexibility, and interoperability issues. Therefore, developing an IoT network architecture is necessary, considering the different requirements of diverse IoT applications and the flexibility and scalability in the deployment and operations.

IoT architectures' numerical simulations and validations are limited to a few QoS metrics, such as energy, latency, and bandwidth. In the validation process, numerical data have been used rather than real-world data to validate most IoT application use cases. Therefore, it is crucial to investigate how these different architectures would impact the performance

of multiple QoS metrics, including resource capacity, delay, reliability, cost, and efficiency. It is worthwhile to investigate the performance of these architectures on real data, to capture the unpredictable behaviors of real networks and evaluate their efficiency.

Most of the node selection and task offloading mechanisms presented so far mainly focused on collating the nodes to meet the resource requirements, enhancing energy efficiency, and optimal path selection considering either cloud or fog layers. However, the node selection mechanism to optimally select nodes between edge, fog, and cloud for processing needs further investigation. Further, the research using edge computation in IoT network architecture has addressed edge computation in mobile devices or IoT devices at the edge node. Therefore, how stationery edge servers can be used for processing along with fog or cloud is another challenge that needs to be mastered to support emerging IoT applications.

Another critical challenge that needs attention is how computation and communication resources can be combined for varying IoT application use cases and how to implement real-time dynamic service-oriented traffic detection at each layer of the network architecture. To address these challenges and limitations in the emerging IoT landscape, IoT requires sufficient support for ubiquitous communications, aggregation, and real-time access to services and information. Therefore, usage of cloud, fog, and edge computing, network slicing, 5G/6G needs to be considered in the design phase of an IoT architecture to gain the benefits of these technologies in the device layer, network layer, and application layer of emerging IoT applications [35]. This will enable the data's collection, processing, and storage dynamically at each layer when and where necessary.

Since the number of IoT devices and their requirements are increasing exponentially, efficient allocation, load balancing, and utilization of computing and communication resources would also need further investigation. In particular, when a request is generated at the edge layer by an IoT application, we need to decide which node is suitable to process the data depending on the QoS requirements of the application and the network limitations of the IoT network architecture.

To overcome the challenges associated with using all three cloud, fog, and edge layers for supporting IoT applications, in this paper, we investigate an IoT network architecture incorporating modern computation and communication technologies. The next challenge in such an architecture is optimally selecting a node to process IoT applications. Therefore, in this paper, we propose an optimal node selection framework for processing the requests coming in from various IoT applications to meet network and application-specific constraints.

IV. IoT Architecture and Optimal Node Selection Framework

In this section, we elaborate on the IoT network architecture in consideration and mathematical formulation of the proposed optimal node selection framework.

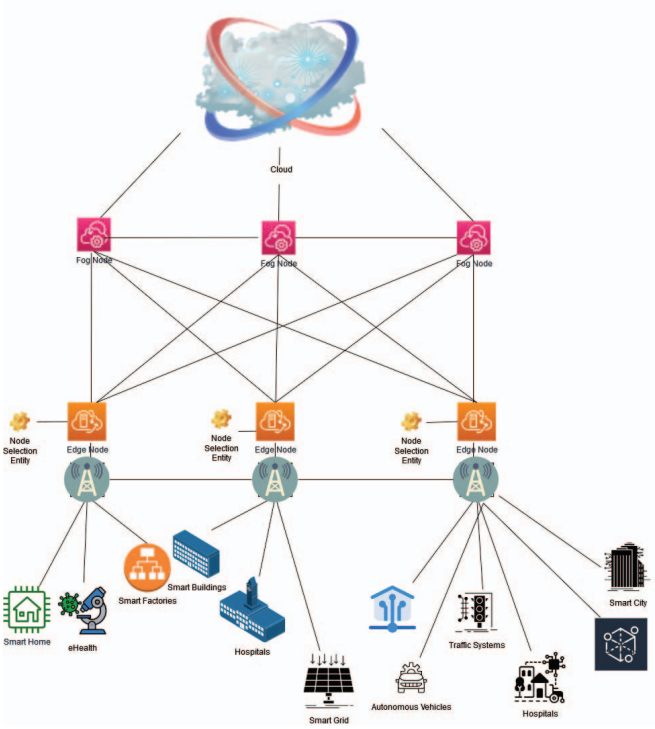

Fig. 2. IoT Network Architecture Diagram.

A. IoT Architecture

Our proposal considers a generalized IoT network architecture consisting of fog, cloud, and edge computing capabilities. This architecture could enable diverse emerging IoT use cases discussed before.

Figure 2 illustrates the IoT network architecture we have proposed to use for serving diverse IoT applications. The architecture consists of a cloud layer at the top tier, which is responsible for processing the IoT request with less stringent requirements. Then, the fog layer at the middle tier consists of fog nodes responsible for local aggregation, processing, and analysis. The last tier is the edge layer, where the edge devices from different users are connected. This network architecture uses edge servers deployed at the edge layer near the base stations. The edge servers would be responsible for processing the data requests based on bandwidth, resource, and latency requirements and allocate the request to the most feasible node in either layer. The node selection decision would be made at the edge layer using an optimization framework. An entity would be designed at the edge layer for efficient node selection and to meet the application-specific QoS requirements. Once the entity decides on an optimal node to process the data, then the data will be moved to that specific node for further processing.

In the following subsection, we explain and provide the mathematical formulation of the proposed optimal node selection framework.

B. Optimization Framework

For load balancing in an IoT network architecture and for supporting advanced IoT use cases, it is advantageous to use a node selection mechanism in the computation and communication converged architecture to ensure efficient operation. The proposed framework ensures that the minimum number of nodes is always used in the architecture. At the same time, all the requirements of 1) IoT applications, such as latency and bandwidth, and 2) communication networks, such as connectivity, bandwidth availability, and delay, are being met. We developed the optimization framework based on Integer Linear Programming(ILP).

In the following subsections, we define the sets, parameters, and variables used to model the optimization framework, followed by a detailed description of the objective function and constraints.

C. Parameters and Sets

We consider different parameters and sets to represent various network and computing nodes, communication links and their limits, locations, and the requirements of diverse IoT applications.

1) Network Parameters:

- nC: Total number of cloud nodes
- nF: Total number of fog nodes
- nE: Total number of edge nodes
- nL: Total number of all nodes in the network
- $R_f[l]$: Resource capacity of the fog node at location l
- $R_e[l]$: Resource capacity of the edge node at location l
- $R_c[l]$: Resource capacity of the cloud node at location l
- B_f: Communication bandwidth capacity supported by all the fog nodes
- B_e: Communication bandwidth capacity supported by all the edge nodes
- B_c: Communication bandwidth capacity supported by all the cloud nodes
- $d[l][j]$: Network delay between l^{th} node and j^{th} node, represents the delay between nodes in the network
- $g[l][j]$: Represents the connectivity between two nodes in the network, $g[l][j]$ =1 if l^{th} node and j^{th} node are connected with each other
- $L_c[l]$: Set of all locations l where the cloud node c has been deployed
- $L_f[l]$: Set of all locations l where the fog node f has been deployed
- $L_e[l]$: Set of all location l where the edge node e has been deployed
- L: Set of all locations of nodes in the network graph

2) Application Request Parameters:

- job: Total number of jobs/ requests
- j_r: Job resource requirement
- j_b: Job bandwidth requirement
- j_l: Job latency requirement
- j_o: Job origin node

3) Sets:

- $E = 1..nE$: Set of edge nodes
- $F = 1..nF$: Set of fog nodes
- $C = 1..nC$: Set of cloud nodes
- $Lo = 1..nL$: Set of nodes, $Lo = E \cup F \cup C$
- $jobn$: $1..job$ Range of all the jobs/ requests

D. Variables

- $e[j][e]$: Boolean variable, $e[j][e] = 1$ if j^{th} job is allocated to e^{th} edge node, $e[j][e] = 0$ otherwise, where $e \in E$ and $j \in jobn$
- $f[j][f]$: Boolean variable, $f[j][f] = 1$ if j^{th} job is allocated to f^{th} fog node, $f[j][f] = 0$ otherwise, where $f \in F$ and $j \in jobn$
- $c[j][c]$: Boolean variable, $c[j][c] = 1$ if j^{th} job is allocated to c^{th} cloud node, $c[j][c] = 0$ otherwise, where $c \in C$ and $j \in jobn$
- E_a[e]: Boolean variable, E_a[e] =1 if e^{th} edge node is an active node, E_a[e] =0 otherwise where $e \in E$
- F_a[f]: Boolean variable, F_a[f] =1 if f^{th} fog node is an active node, F_a[f] =0 otherwise where $f \in F$
- C_a[c]: Boolean variable, C_a[c] =1 if c^{th} cloud node is an active node, C_a[c] =0 otherwise where $c \in C$

E. Objective Function

The objective function of the framework is to minimize the number of nodes used in the entire network. The objective function is defined in equation 1, which minimises the total number of active edge (E_a), fog (F_a) and cloud nodes (C_a).

$$Minimize(\sum_{x \epsilon E} E_a[x] + \sum_{y \epsilon F} F_a[y] + \sum_{z \epsilon C} C_a[z]) \quad (1)$$

F. Constraints

The framework minimizes the number of active nodes whilst satisfy the requirements of the network and IoT applications. In this subsection, we formulate the constraints relevant to these requirements.

1) Network Constraints:

- When we allocate a new job/request to a node, we need to make sure that each job request is always processed at only one node at a time in the entire IoT network architecture. The equation 2 defines this constraint where the e, f, j variables store the values of the each job allocated at each edge, fog and cloud node locations, respectively.

$$\sum_{x \epsilon E} e[j][x] + \sum_{y \epsilon F} f[j][y] + \sum_{z \epsilon C} c[j][z] == 1, \forall\ j\ in\ jobn \quad (2)$$

- In the framework, we also need a constraint for limiting the total number of active nodes to make sure the number of active nodes are always less then or equal to maximum number of nodes deployed at each fog edge and cloud layer. Equations 4, 3, 5 are defined to satisfy this constraint where E_a, F_a, C_a are active nodes at edge,

fog and cloud layer and nE, NC, nF are the maximum nodes supported at each layer respectively.

$$\sum_{l \epsilon E} E_a[l] \leq nE \tag{3}$$

$$\sum_{l \epsilon F} F_a[l] \leq nF \tag{4}$$

$$\sum_{l \epsilon C} C_a[l] \leq nC \tag{5}$$

2) Application Constraints:

- We also need to verify that the new requests are allocated to a node for processing, has connectivity with the communication node where the requests is originated. Further, the network connectivity should also be able to satisfy the delay requirement of the request coming from the IoT application. Equations 6, 7, 8 are defined to satisfy delay, connectivity constraints for edge, fog and cloud layer nodes and are always less then or equal to the latency required by the IoT job request.

$$e[j][a]*(g[j_o[j]][a+nC+nF]*(d[j_o[j]][a+nC+nF]) <= j_l[j] \ ,\forall \ j \ in \ jobn \ ,\forall \ a \ in \ E \tag{6}$$

$$f[j][a] * (g[j_o[j]][a + nC] * (d[j_o[j]][a + nC]) <= j_l[j] \ ,\forall \ j \ in \ jobn \ ,\forall \ a \ in \ F \tag{7}$$

$$c[j][a] * (g[j_o[j]][a] * (d[j_o[j]][a]) <= j_l[j] \ ,\forall \ j \ in \ jobn \ ,\forall \ a \ in \ C \tag{8}$$

- When we allocate a new request to a node, we need to make sure the resource requirement of the job can be satisfied by the remaining resource processing capacity of the allocated node at edge, fog or cloud layer. The equations 9, 10, 11 help in maintaining this constraint by checking whether the resource requirement of the job when subtracted from the resource capacity of the node is always greater than or equal to zero at all three layers.

$$R_e[l] - \sum_{j \epsilon jobn} e[j][l] * j_r[j] \geq 0, \forall \ l \ in \ E \tag{9}$$

$$R_f[l] - \sum_{j \epsilon jobn} f[j][l] * j_r[j] \geq 0, \forall \ l \ in \ F \tag{10}$$

$$R_c[l] - \sum_{j \epsilon jobn} c[j][l] * j_r[j] \geq 0, \forall \ l \ in \ C \tag{11}$$

- We also need to verify the bandwidth requirement of the job against the bandwidth supported by the nodes at edge, fog and cloud. The bandwidth constraints at edge, fog, cloud nodes are defined in Equations 12, 13, 14, respectively. This is achieved by checking whether the bandwidth requirement of the job when subtracted from the bandwidth capacity supported by the node, is always greater than or equal to zero at all three layers.

$$B_e[l] - \sum_{j \epsilon jobn} e[j][l] * j_b[j] \geq 0, \forall \ l \ in \ E \tag{12}$$

$$B_f[l] - \sum_{j \epsilon jobn} f[j][l] * j_b[j] \geq 0, \forall \ l \ in \ F \tag{13}$$

$$B_c[l] - \sum_{j \epsilon jobn} c[j][l] * j_b[j] \geq 0, \forall \ l \ in \ C \tag{14}$$

- We need a constraint to make sure that a job allocated at a location for each layer is less than or equal to nodes activated at each layer. This constraint is defined in edge, fog, and cloud layers using Equations 15, 16, and 17, to make sure that when a job is allocated at a node e, f, c at each layer, the respective node is active E_a, F_a, C_a at that respective layer.

$$e[j][l] \leq Ea[l], \forall \ j \ in \ jobn \forall \ l \ in \ E \tag{15}$$

$$f[j][l] \leq Fa[l], \forall \ j \ in \ jobn \forall \ l \ in \ F \tag{16}$$

$$c[j][l] \leq Ca[l], \forall \ j \ in \ jobn \forall \ l \ in \ C \tag{17}$$

- We also need a constraint to make sure the total number of nodes getting activated in a given layer for processing the incoming IoT request could be greater then or equal to initial number of activated nodes deployed. This constraint is satisfied in edge, fog, and cloud layers using Equations 18, 19, and 20, respectively where total number of nodes activated at edge (E_a), fog (F_a), and cloud layer (C_a) can be greater than or equal to total number of nodes actually deployed and active at each layer during initial deployment.

$$E_a[l] \geq L_e[l], \forall \ l \ in \ E \tag{18}$$

$$F_a[l] \geq L_f[l], \forall \ l \ in \ F \tag{19}$$

$$C_a[l] \geq L_c[l], \forall \ l \ in \ C \tag{20}$$

V. Evaluation

In this section, we evaluate the proposed framework by using it to optimally select nodes for processing under different scenarios. We use IBM CPLEX for the implementation of the framework. Figure 3 illustrates the IoT network graph we considered for the evaluation. The network graph consists of two cloud nodes, five fog nodes, and eight edge nodes. Figure. 3 also shows the connectivity between different nodes. For the evaluation, the nodes were numbered at different layers. The resource capacity and bandwidth supported by each node are also indicated in Fig. 3 with notations R and B, respectively. The network delay between nodes is also

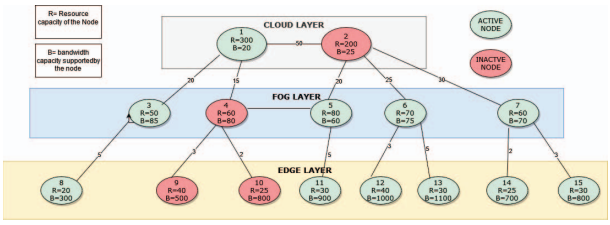

Fig. 3. Data set: network graph

illustrated in the figure. The delays between all the nodes were pre-calculated and stored in parameter d.

All the considered nodes deployed in the network are at different locations and connected via a mesh network with varying resource capacity, varying bandwidth supported by each node and varying latency. For verifying our node selection optimization framework, we would consider that varying numbers of IoT requests are received by our framework and test its efficiency in efficiently processing all the IoT requests using the least number of active nodes. Each new request is also associated with a few parameters, including its computation resource (j_r), bandwidth(j_b), and latency (j_l) requirements and its origin node (j_o). Therefore, each new request with job number j can be defined as "Job $j[j_r, j_b, j_l, j_o]$". We use four scenarios for evaluating the framework.

In the first scenario, cloud node 2, fog node 4, and edge nodes 9 and 10 are deactivated. Therefore, 11 computation nodes are active in our considered IoT network. For example, we consider a single IoT request from an autonomous vehicle with a low latency requirement received at node 12, which can be defined as Job 1[40,1000, 1, 12]. When we applied our optimization framework, the job was allocated to the active edge node 12, as shown in Figure 4, without activating any additional node in the network. The dotted lines between nodes and new requests indicate the selected node for processing a particular job.

In the second scenario, we used the same IoT network graph with 11 active nodes as in scenario 1. However, here we have increased the total number of IoT requests received to six from eHealth, autonomous vehicles, and smart grid IoT

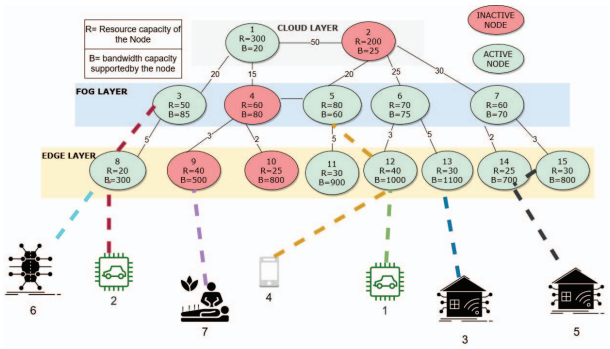

Fig. 4. Graphical illustration of the optimal solutions: three scenarios

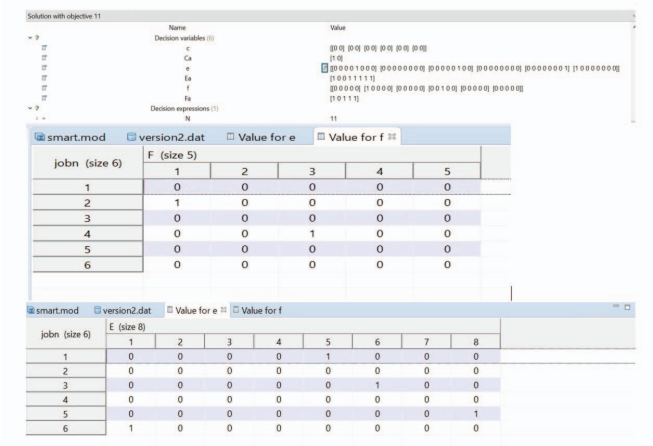

Fig. 5. Optimal results of Scenario 2

applications, with varying requirements, which were defined as Job 1[40,1000,1,12], Job 2[50,85,6,8], Job 3[30,1100,1,13], Job 4[80,60,13,12], Job 5[30,800,5,14], and Job 6[20,300,1,8]. Using our optimization framework, we obtained the optimal allocations for jobs (ranging from 1 to 6) as shown in Fig. 4. As shown in the figure, jobs 1, 2, 3, 4, 5, and 6 were allocated to edge node 5, fog node 1, edge node 6, fog node 3, edge node 8, and edge node 1, respectively. We have also illustrated this solution in the screenshot from the CPLEX program shown in Fig. 5. As shown in the figure, active edge and fog nodes are denoted by the variables e and f, respectively. As we can see from the result, the number of active nodes remained the same. Therefore, it is clear from the results that the framework makes sure all the requests are processed with minimum utilization of resources and nodes.

In the third scenario, we used the same IoT network graph

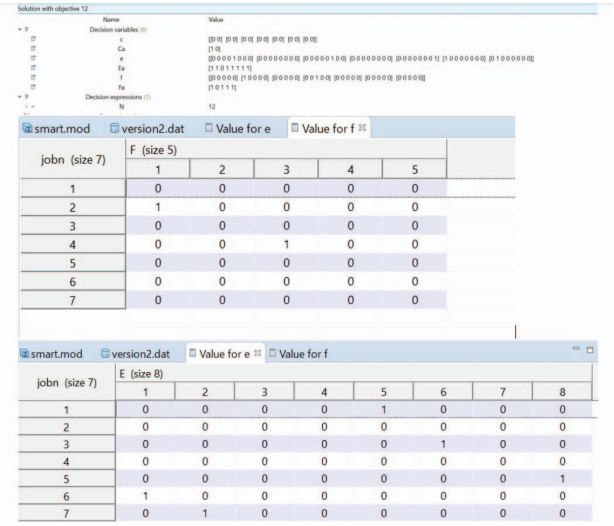

Fig. 6. Scenario 3 Results.

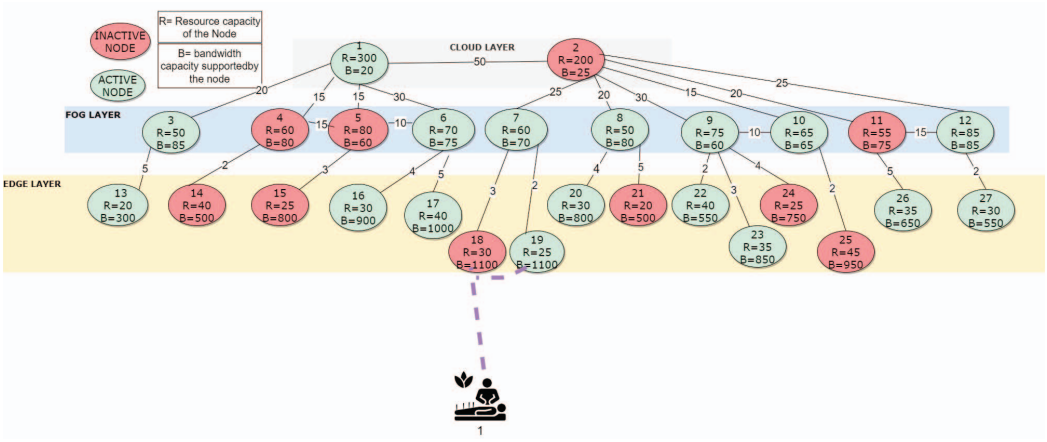

Fig. 7. Graphical illustration of the optimal solution: Fourth scenario

with 11 active nodes as in scenario 2. Here, we considered seven new job requests, a combination of six job requests considered in scenario 2 and one additional job request from the eHealth IoT application, Job 7[40,500,1,9], as shown in Fig. 4. The optimization framework produced the allocations shown in Fig. 4. Jobs 1, 2, 3, 4, 5, 6, and 7 were allocated to edge node 5, fog node 1, edge node 6, fog node 3, edge node 8, edge node 1, and edge node 2, respectively. Edge node 2 was inactive before and got activated to serve Job 7. The e and f variables that illustrate the optimal allocations are shown in Fig. 6. As we can see from the results, the framework activates an inactive node to process the IoT requests. From this scenario, we can understand that if the IoT network architecture falls short of resources to process all the IoT requests with the currently active nodes, it can activate inactive nodes in the architecture to process them while satisfy their requirements.

In the fourth scenario, we have considered a more complex network diagram as shown in Fig. 7 with two cloud nodes, ten fog nodes, and fifteen edge nodes for evaluation. Figure 7 also shows connectivity between different nodes and highlights active and inactive nodes in the entire network graph. Here we have considered a single job request from the eHealth application defined as Job 1[25,1000,5,18] being received at edge node 7, which is an inactive node. Considering our IoT network architecture and the network graph, if we did not use any optimization algorithms for node selection, then the inactive edge node 7 would be activated to process the request. This will cause overutilization of the number of nodes and underutilization of resources in the network. However, using the proposed optimization framework, an already active edge node 8 is used to process this new eHealth job request rather than activating inactive node 7. The optimal allocation is also shown in Fig. 7. This results in using the minimum number of already active nodes to process the IoT request and enhancing the network resource utilization.

From the above evaluation, we can conclude that the considered IoT network architecture and the ILP framework can optimally allocate and process the requests originating from different IoT applications, including smart grid, autonomous vehicles, and e-Health, whilst satisfying the application and network requirements. ILP is an NP-hard problem, and we used CPLEX to find the solutions under different network scenarios. The solution time could drastically vary with the size of the data set and the computation capabilities of the device running the framework. Therefore, in our future work, we would also explore a heuristic approach for node selections and test the framework on actual data sets from different IoT applications to compare the results against the optimal solutions obtained by the proposed optimization framework.

VI. Conclusion

In this paper, we studied an IoT architecture with edge, fog, and cloud computation layers to support emerging IoT applications and how we can optimally allocate different IoT requests to nodes at different layers for processing. We have proposed an optimal node selection framework based on ILP for efficiently processing the IoT requests with varying stringent network and application constraints whilst minimizing the number of nodes used. The approach was evaluated using IBM CPLEX implementation, which validated the feasibility of our approach in handling the IoT requests from upcoming IoT use cases such as autonomous vehicles, eHealth, and smart gird efficiently under different scenarios. The results demonstrated the efficiency of our proposed framework. Planned future works include developing heuristic approaches that can be used for real-time node selections and comparing the results with the solution from the proposed optimal node selection framework.

References

[1] P. Hu, W. Chen, C. He, Y. Li, and H. Ning, "Software-defined edge computing (sdec): Principle, open iot system architecture, applications, and challenges," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5934–5945, 2020.

[2] M. Zerifi, A. Ezzouhairi, and A. Boulaalam, "Overview on sdn and nfv based architectures for iot environments: challenges and solutions," in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, 2020, pp. 1–5.

[3] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020.

[4] J. Yao and N. Ansari, "Fog resource provisioning in reliability-aware iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8262–8269, 2019.

[5] M. A. Bouras, F. Farha, and H. Ning, "Convergence of computing, communication, and caching in internet of things," *Intelligent and Converged Networks*, vol. 1, no. 1, pp. 18–36, 2020.

[6] A. Buzachis, A. Galletta, A. Celesti, M. Fazio, and M. Villari, "Development of a smart metering microservice based on fast fourier transform (fft) for edge/internet of things environments," in *2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*, 2019, pp. 1–6.

[7] M. Alrowaily and Z. Lu, "Secure edge computing in iot systems: Review and case studies," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 440–444.

[8] M. S. Raghavendra, P. Chawla, and A. Rana, "A survey of optimization algorithms for fog computing service placement," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 259–262.

[9] M.R.Bobouakouk, A. Abdelli, and L.Mokdad, "Survey on the cloud-iot paradigms: Taxonomy and architectures," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–6.

[10] T.Goethals, F.De Turck, and B.Volckaert, "Near real-time optimization of fog service placement for responsive edge computing," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–17, 2020.

[11] S. Liu, C. Guo, F. Al-Turjman, K. Muhammad, and V. H. C. de Albuquerque, "Reliability of response region: a novel mechanism in visual tracking by edge computing for iiot environments," *Mechanical systems and signal processing*, vol. 138, p. 106537, 2020.

[12] H. Liu, L. T. Yang, M. Lin, D. Yin, and Y. Guo, "A tensor-based holistic edge computing optimization framework for internet of things," *IEEE Network*, vol. 32, no. 1, pp. 88–95, 2018.

[13] B. Han, S. Wong, C. Mannweiler, M. R. Crippa, and H. D. Schotten, "Context-awareness enhances 5g multi-access edge computing reliability," *IEEE Access*, vol. 7, pp. 21 290–21 299, 2019.

[14] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *2017 IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1–7.

[15] N. Vance, M. T. Rashid, D. Zhang, and D. Wang, "Towards reliability in online high-churn edge computing: A deviceless pipelining approach," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2019, pp. 301–308.

[16] J. Liu and Q. Zhang, "Reliability and latency aware code-partitioning offloading in mobile edge computing," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–7.

[17] J. Pereira, L. Ricardo, M. Luís, C. Senna, and S. Sargento, "Assessing the reliability of fog computing for smart mobility applications in vanets," *Future Generation Computer Systems*, vol. 94, pp. 317–332, 2019.

[18] K. Dantu, S. Y. Ko, and L. Ziarek, "Raina: Reliability and adaptability in android for fog computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 41–45, 2017.

[19] J. Wang, K. Liu, B. Li, T. Liu, R. Li, and Z. Han, "Delay-sensitive multi-period computation offloading with reliability guarantees in fog networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2062–2075, 2020.

[20] M. Hussein and M. Mousa, "Efficient task offloading for iot-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37 191–37 201, 2020.

[21] R. Silva, D. Santos, F. Meneses, D. Corujo, and R. L. Aguiar, "A hybrid sdn solution for mobile networks," *Computer Networks*, vol. 190, p. 107958, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621000931

[22] R.Cong, z.Zhao, G.Min, and C. Y.Jiang, "Edgego: A mobile resource-sharing framework for 6g edge computing in massive iot systems," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[23] K. Lin, Y. Li, Q. Zhang, and G. Fortino, "Ai-driven collaborative resource allocation for task execution in 6g enabled massive iot," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[24] U. M. Malik, M. A. Javed, S. Zeadally, and S. u. Islam, "Energy efficient fog computing for 6g enabled massive iot: Recent trends and future opportunities," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[25] P. Maiti, J. Shukla, B. Sahoo, and A. K. Turuk, "Qos-aware fog nodes placement," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, 03 2018, pp. 1–6.

[26] T. Rahman, X. Yao, G. Tao, H. Ning, and Z. Zhou, "Efficient edge nodes reconfiguration and selection for the internet of things," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4672–4679, 2019.

[27] A. Alagha, S. Singh, R. Mizouni, A. Ouali, and H. Otrok, "Data-driven dynamic active node selection for event localization in iot applications - a case study of radiation localization," *IEEE Access*, vol. 7, pp. 16 168–16 183, 2019.

[28] S. A. Aboalnaser, "Energy–aware task allocation algorithm based on transitive cluster-head selection for iot networks," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, 2019, pp. 176–179.

[29] Y. Li, Y. Zhang, Y. Liu, Q. Meng, and F. Tian, "Fog node selection for low latency communication and anomaly detection in fog networks," in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2019, pp. 276–279.

[30] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "Iot connectivity technologies and applications: A survey," *IEEE Access*, vol. 8, pp. 67 646–67 673, 2020.

[31] A. Nirmalathas, T. Song, S. Edirisinghe, K. Wang, C. Lim, E. Wong, C. Ranaweera, and K. Alameh, "Indoor optical wireless access networks-recent progress (invited)," *IEEE/OSA Journal of Optical Communication and Networking*, vol. 13, no. 2, pp. A178–A186, Feb 2021.

[32] C. Ranaweera, A. Nirmalathas, E. Wong, C. Lim, P. Monti, L. W. Marija Furdek, B. Skubic, and C. M. Machuca, "Rethinking of optical transport network design for 5g/6g mobile communication," *IEEE Future Networks Tech Focus*, vol. 12, 2021.

[33] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[34] N. Gupta, S. Sharma, P. K. Juneja, and U. Garg, "Sdnfv 5g-iot: A framework for the next generation 5g enabled iot," in *2020 International Conference on Advances in Computing, Communication Materials (ICACCM)*, 2020, pp. 289–294.

[35] C. Ranaweera, J. Kua, I. Dias, E. Wong, C. Lim, and A. Nirmalathas, "4g to 6g: disruptions and drivers for optical access (invited)," *IEEE/OSA Journal of Optical Communication and Networking*, vol. 14, no. 2, pp. A143–A153, Feb 2022.