

An Efficient Quantum based D2D Computation and Communication Approach for the Internet of Things

M S Mekala^{a,b} · Gautam Srivastava^{c,g} · Jerry
Chun-Wei Lin^d · Gaurav Dhiman^e · Ju H. Park^f ·
Ho-Youl Jung^{a,*}

Received: DD Month YEAR / Accepted: DD Month YEAR

Abstract We can say with some clarity that the Internet of Things (IoT) can be made up of a set of embedded devices such as light detection sensors, ranging (LiDAR) sensors, and millimetre-wave (mmWave) sensors. These sensors generate a massive amount of data in which limited communication capacity is available to share a massive amount of data to Fog-based Roadside Units (RSU) for data process and analysis service. Fog-based RSU has become an emerging paradigm in intelligent transportation but needs research attention to design intelligent decision-making methods for data communication and computation at Fog-based RSU. To address these issues, we design a two-level Quantum based D2D Computation, Communication (QDC^2) approach. First, design a bandwidth allocation strategy based on spatial importance score factors to resolve embedded devices' data transmission issues. Second, design an adaptive equilibrium service offloading strategy based on device-centric measurements to assess the computation capacity and performance rate for resolving Fog-node computation consistency issues. Additionally, Fog-based RSU is interconnected with LAN helps to optimize service latency. Simulation results show that our approach achieved a high service reliability rate (79.56%), low error rate (0.9%), and an execution delay of 22.5s for 15 devices than state-of-art approaches.

Keywords Fog-based quantum computing · Data communication and computation · LiDAR · Cameras · Deep Learning · Feature fusion

1 Introduction

In Information and Communication Technology (ICT), the Internet of Things (IoT) is an emerging paradigm to fulfil the Industry 4.0 requirements. Recent AI-IoT frameworks impact the cost, performance and energy efficiency of IoT devices, network technologies (such as 4G and 5G) [1], and distributed computing architectures that led to the explosive growth of edge-service reliability rate. Specifically, new distributed applications related to transportation, healthcare, automated vehicles, cloud robotics, smart homes and cities, video surveillance and streaming require edge

^a Dept. of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, Korea. E-mail: msmekala@yu.ac.kr ·

^b RLRC for Autonomous Vehicle Parts and Materials Innovation, Yeungnam University, Gyeongsan 38544, Korea.

^c Dept. of Math and Computer Science, Brandon University, Brandon, Canada. E-mail: srivastavag@brandonu.ca

^f Western Norway University of Applied Sciences, Bergen, Norway. E-mail: jerrylin@ieee.org

^e Department of Computer Science, Panjab University, Patiala, India. E-mail: gdhiman0001@gmail.com

^f Depart. of Electrical Engineering, Yeungnam University, Gyeongsan, Korea. E-mail: jessie@ynu.ac.kr

^g Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan.

* Corresponding: Ho-Youl Jung, E-mail: hoyoul@yu.ac.ke

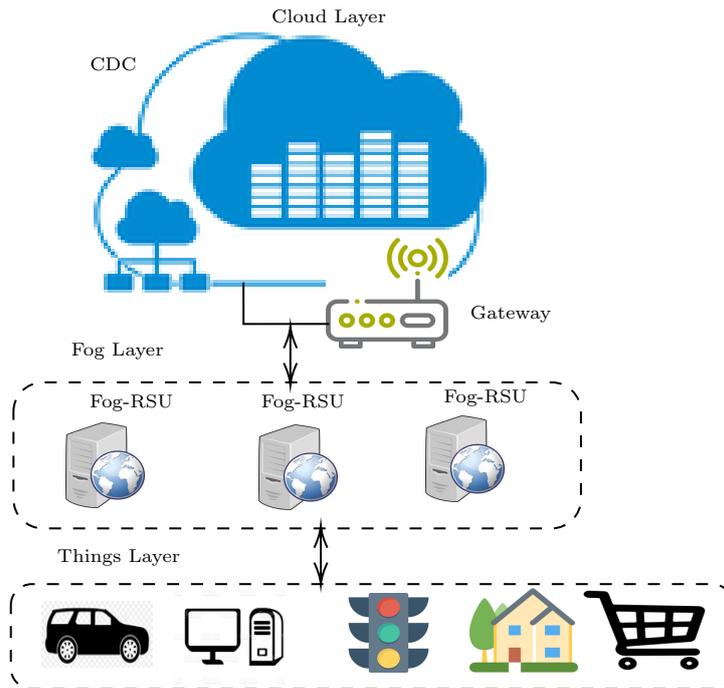


Fig. 1: Structure of Fog Computing

computing orchestration to achieve low latency [2]. It led to unprecedented growth in data and latency, which demands inadequate resources to handle and manage data. It is a global challenge to optimize resource consumption. The new distributed applications may be bandwidth-hungry (video surveillance, video conferencing, traffic monitoring), latency-critical (automated vehicles, robotic surgery, safety). The edge applications require high data availability, service reliability, and low latency to meet the requirements of time constraint service execution. Large-scale deployment of industrial IoT devices is expected to play a vital role in the development of smart cities, which will generate large volumes of aggregated cellular data that may choke the network [3]. Fog computing is assumed to be the eminent solution to formulate the issues by accommodating the limited cloud services near to the framework as depicted in Fig. 1.

Research Motivation: The Fog devices are interconnected with LAN, which helps to minimize the latency due to consistent data availability, demand-based service offloading, and annoying computation resources. Service offloading, bandwidth resource allocation are dependent on the computation capacity of Fog-server. The computation services require a set of feature selection and classification services to enhance data accuracy, impacting the consumption of computation resources. Since the edge devices generate enormous data that demand heterogeneous computation resources, it is challenging to process with classical computing to meet time-sensitive application requirements and service offloading, bandwidth allocation among fog devices that are geographically distributed. Limited resource capacity causes offloading the service to another suitable Fog-RSU device based on computing and communication capacity characteristics to accomplish low latency and high data quality. In this regard, incorporating quantum computing mechanics based on quantum theory formulates the work's objective with a highly scalable and reliable service rate.

1.1 Quantum Theory Overview

In classical computer, the memory unit state represents with binary bits (0 or 1); in Quantum Computing (QC), quantum bit (q - bit) holds a unit of information like a classical bit, but the state could be 0 , 1 , or in a quantum linear superposition of both and it is represented as $\partial|0\rangle + \delta|1\rangle$. The quantum states are represented as $|0\rangle$, $|1\rangle$ and a pair of complex numbers (∂, δ) is formulated as $|\partial|^2 + |\delta|^2 = 1$ [4]. The probability of *state* - 0 and *state* - 1 refer as $|\partial|^2$, $|\delta|^2$, respectively. A

qubit ($q - bit$) with n -size which holds 2^n types of states and we observe in Eq. 1

$$\begin{aligned} \text{State} - 0 &\rightarrow \left[\partial_{01} \mid \partial_{02} \mid \cdots \mid \partial_{0n} \right] \\ \text{State} - 1 &\rightarrow \left[\delta_{11} \mid \delta_{12} \mid \cdots \mid \delta_{1n} \right] \end{aligned} \quad (1)$$

For instance, a quantum individual with 3 ($q - bits$) is described with Eq. 2

$$\begin{aligned} \text{State} - 0 &\rightarrow \left[\frac{1}{\sqrt{2}} \mid \frac{1}{\sqrt{2}} \mid \frac{\sqrt{3}}{2} \right] \\ \text{State} - 1 &\rightarrow \left[\frac{1}{\sqrt{2}} \mid -\frac{1}{\sqrt{2}} \mid \frac{1}{2} \right] \end{aligned} \quad (2)$$

It also be formulated as: $\left(\frac{\sqrt{3}}{4}\right) |000\rangle + \left(\frac{1}{4}\right) |001\rangle - \left(\frac{\sqrt{3}}{4}\right) |010\rangle - \left(\frac{1}{4}\right) |011\rangle + \left(\frac{\sqrt{3}}{4}\right) |100\rangle + \left(\frac{1}{4}\right) |101\rangle - \left(\frac{\sqrt{3}}{4}\right) |110\rangle - \left(\frac{1}{4}\right) |111\rangle$ It refers that, the estimating probability of the states are $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$, are $\frac{3}{16}, \frac{1}{16}, \frac{3}{16}, \frac{1}{16}, \frac{3}{16}, \frac{1}{16}, \frac{3}{16}, \frac{1}{16}$, *and* $\frac{1}{16}$, respectively. In [5], quantum computing-inspired optimization methods have been designed to maximize the data accuracy by assessing the sensor characteristics such as sensor vicinity and sensor space. The performance is measured by resolving vehicular routing problems based on WiSense nodes, and quantum simulators. The nemours data computing issues can be resolved in terms of data accuracy (by selecting feasible feature set and classification mechanism), service offloading (by designing a novel decision system), and resource allocation per demand. In this regard, there is a scope to design Quantum based D2D Computation, Communication (QDC^2) approach and the main contributions are listed as follows.

1. Design an adaptive equilibrium service offloading strategy based on device-centric measurements to assess server computation capacity to minimize the computation delay.
2. Design a bandwidth allocation strategy based on spatial importance score factor to resolve data transmission delay.
3. Analysing MATLAB and FogSim simulation results to accomplish high service reliability and system performance accuracy.

The rest of the paper is arranged as follows. Section 2 reviews the related works on range-free localization technologies. The system model is presented and the proposed localization algorithm is detailed described in Section 3. Experimental results and discussions are given in Section 4. Finally, Section 5 describes conclusions and possible future work.

2 Related work

This section describes the recent works related to our paper objectives, which are formulated as follows

In [6], quantum-based spatial-temporal service execution strategy for cloud environment based on novel designed prediction model through kernel-based machine learning algorithm. They were estimated service execution duration and space occupation parameters before assigning the service and has achieved 26.7% enhanced performance than existing methods. In [7], the Quantum-behaved particle swarm optimization (QPSO) approach has been designed to enhance the performance of particle swarm optimization (PSO) with low computing cost to meet time-constraint application requirements. They were considered quantum gates and operations during the optimization process. In [8], the Quantum-based particle swarm optimization method has been designed to allocate the resources, and they were concentrated on minimizing the iterations to achieve a high optimal rate.

In [9], a crossover quantum particle swarm optimization algorithm has been designed for non-uniform IoT node deployments based on node density and node coverage. The differential evaluation operator and crossover operator have been used to achieve high precision coverage. In [10], a Quantum-inspired green communication framework for Energy Balancing in sensor-enabled IoT systems (Q-EBioT) has been designed to optimize the cost of paths based on rotation angle and position measurements. The simulation outcomes show a significant increment in a network lifetime.

A Quantum computing-based evolutionary algorithm (QEA) has been designed to formulate network optimization issues [11]. Additionally, the QEA concentrated on resolving the routing issues of IoT frameworks. In [12] a quantum-inspired ant-based routing algorithm has been designed based on energy consumption and hop count to optimize the IoT framework routing issues. Q-bits, Q-gate parameters are considered to design the ant-based routing technique for network consolidation. Quantum-inspired evolutionary algorithm-based clustering technique has been designed to accomplish hierarchical routing issues of IoT framework [13]. Subsequently, a quantum-inspired

genetic algorithm has been designed to optimize the service execution based on a novel routing strategy for IoT [14]. In [15, 16], a Deep Q-learning Network (DQN) has been designed through the Q-value of a targeted node to diminish training time duration during complex issues to enhance performance accuracy. Each layer state value has been estimated, but in the last layer, those values are added into a single element to enhance the system performance [17]. The Joint Policy Gradient Method (JCOR) has been designed to estimate and formulate device-centric issues, where several factors have been examined asynchronously to improve performance accuracy [18].

3 Model Construction

This section focus on assessing two objectives based on quantum theory. A set of fog-RSU nodes are denoted as $S = \{j : 1, 2, \dots, m\}$, $V = \{i : 1, 2, \dots, n\}$ is a set of service requests from vehicle v , and $K = \{k : 1, 2, \dots, k_n\}$ a set of parameters. Initially, *service offloading* constraints are formulated for which a set of parameters are considered to select the feasible fog-node for service offloading. The computation capacity of fog-RSU node is defined as follows

$$\Delta_j^t = \prod_{j=1}^m \left(\frac{s^e + s^c}{s_b} \right) - \frac{TES}{TAS} \quad (3)$$

where s^e , s^c , s^b denotes number of processing elements, processor capacity, node bandwidth which are pre-defined variables. Total execution services (TES), total assigned services (TAS) are two essential factors to consider during node capacity estimation. The workload of node is estimated as follows

$$\varpi_j^t = \frac{v^t}{\Theta_j^t} \quad (4)$$

where v^t , Θ_j^t refers service arrival rate at time t , average required service execution time to execute the arrived services. Average required service execution time $\Theta_{j,ct}^t$ is estimated as follows.

$$\Theta_j^t = \Theta_{j,ct}^t \times \Theta_{j,trt}^t \quad (5)$$

Where $\Theta_{j,ct}^t$ is a service execution time, $\Theta_{j,trt}^t$ is a required data transmission time, which are defined as follows

$$\Theta_{j,ct}^t = \frac{h_i^t \times \ell_v}{\Delta_{avc}^t} \quad (6)$$

$$\Theta_{j,trt}^t = \frac{\ell_v}{s_{v \rightarrow s}^{b,t}} \quad (7)$$

where ℓ_v , h_i^t refers data size, computing intensity of service respectively. $s_{v \rightarrow s}^{b,t}$ is a transmission rate at time t between vehicle to Fog-RSU is estimated as follows

$$s_{v \rightarrow s}^{b,t} = s^b \log_2 \left(\frac{1 + \varepsilon_{po}^v g_{s,v}^t}{(np)^2} \right) \quad (8)$$

Where, s^b is RSU bandwidth capacity, ε_{po}^v is vehicle transmission energy and np , $g_{s,v}^t$ is energy noise and channel gain between vehicle and RSU. The RSU cost is formulated as

$$\mathbb{C}_s^t = \sum_{v_s \in v} \left(c_{vn} \cdot s_{v \rightarrow s}^{b,t} \cdot \varpi_j^t + c_{df} (1 - q_v^t) \ell_v \right) + c_{se} \cdot L_s \cdot \nabla_j^t, \quad (9)$$

where, v_s bunch of vehicles associated with RSU, c_{vn} , c_{df} , c_{sc} denotes virtual network cost, data sharing cost, service execution cost respectively and ℓ_v denotes anticipated service request size, $q_v^t \in \{0, 1\}$ is a data fetching decision, whereas $q_v^t = 0$ denotes the RSU fetch the data from server, L_s is load of RSU. The required data is not available, then data fetching may affect the computation performance, leads computation delay [19, 20, 21]. The service offload cost ∇_j^t is estimated as follows

$$\nabla_j^t = \frac{\Delta_{j,rc}^t + \ell_v}{s_{v \rightarrow s}^{b,t}} + \Theta_j^t \quad (10)$$

The cost threshold value is estimated as follows

$$\Upsilon_{j,c}^t = \max(\mathbb{C}_s^t) \quad (11)$$

$$D_{fe,t} = \sum_{s=1}^m o_v^t q_v^t \left(\frac{\ell_v}{s^b (v_d^s)} \right), \quad (12)$$

where, v_d^s refers service demand on RSU s and $o_v^t \in \{0, 1\}$ is a data offloading decision between vehicle and RSU, whereas $o_v^t = 0$ denotes the vehicle executes the request by itself other wise offload to nearby RSU. The communication delay is estimated as follows

$$D_{com,t} = \sum_{s=1}^m (1 - o_s^t) D_{cm,t}^s + \left(o_s^t + D_{tran,t}^s + \sum_{v=1}^V D_{cm,t}^v \right), \quad (13)$$

where, vehicle computation delay is estimated as follows

$$D_{cm,t}^v = \sum_{v_s \in v} \frac{\mu_v^t \times \ell_v \times \kappa_v^{\ell_v}}{\kappa_v}, \quad (14)$$

where μ_v^t is a vehicle coverage capacity of vehicle, $\kappa_v^{\ell_v}$ is anticipated MIPS for execution of bit-data, and κ_v vehicle processor capacity. RSU transmission delay is as follows

$$D_{tran,t}^s = \sum_{v_r \in v} \frac{\mu_s^t \times \ell_v}{\phi_{r,v}^t}, \quad (15)$$

where μ_s^t is RSU coverage capacity. The RSU computation delay is estimated as follows

$$D_{cm,t}^m = \frac{L_s}{\left(\kappa_s / \ell_v \times \kappa_v^{\ell_v} \right) - L_s}, \quad (16)$$

where, $\kappa_s / \ell_v \times \kappa_v^{\ell_v}$ denotes anticipated service rate. Now, the required data availability or fetching and concerning data transfer delay are augmented together to assess the network computation performance or computation delay as follows

$$D = D_{fe,t} + D_{com,t} \quad (17)$$

Algorithm 1 illustrates the decision of service execution request, and RSU computation cost estimation and analysis through quantum-based cost optimization strategy. Line-1 defines the entail parameters. Line -2 defines the service queue initial value, which is $\rho = 0$. The estimation of RSU service queue cost concerning time is carried out with line-3, and line-3 measures the service list of RSU. Here, $s_{v \rightarrow s}^{b,t}$ is a threshold of service communication cost. If the service cost is less than the threshold value, measure the total queue cost and update it as a threshold value observed in Line 4-7. Line-8 estimates the communication delay between the vehicle and RSU. The RSU computation delay is estimated with Eq. 17. Before estimating the computation cost, the computation delay and communication delay are should be less than the threshold values. Line-13 estimates the system cost of RSU. As per the arrived service request, if the RSU cost is moderate, then continue the execution; otherwise, the service request is offloaded to RSU because the computation cost is too high, which causes to violate the threshold condition in line-23.

3.1 Quantumizing Computation and Communication parameters

In the previous section, a set of Fog-RSU node parameters are discussed which helps to select the suitable Fog-RSU node for offloading the computation required services. The incorporation of the QDC^2 approach in estimating the listed parameters is important in terms of qubits. A qubit is treated as a probabilistic parameter that should satisfy the conditional normalization. The parameter probability helps to classify the state of each parameter by satisfying conditional parameter probability value $p \geq$ threshold value, then the parameter is in state-1 or δ ; else state-0 or ∂ which concludes Fog-RSU node state concerning the parameter. The mathematical representation is as follows

$$q - bit|state\rangle = \sqrt{(1-p)}|0\rangle + \sqrt{p}|1\rangle \quad (18)$$

The above derivation is part of conditional normalization and similarly, the rest of parameters based on the objective wise is derived. *The state-0 of Fog-RSU node indicates low availability to*

Algorithm 1: Quantum based D2D Computation and Communication Cost Analysis

input : $\kappa_s / \ell_v \times \kappa_v^{\ell_v}$, S number of RSU, V number of service request from vehicle v and T number of time slots

output: Cost minimization

- 1 Let $\kappa_s / \ell_v \times \kappa_v^{\ell_v} \neq 0$, $\phi_{s,v}^t \neq 0$, $D \neq 0$, $c_{vn} \neq 0$, $c_{df} \neq 0$, $c_{se} \neq 0$;
- 2 **for** each $t = 1$ to T **do**
- 3 **if** $s_{v \rightarrow s}^{b,t} \leq s_{v \rightarrow s}^{b,t}$ **then**
- 4 Estimate service queue cost of RSU as follows;
- 5 $\kappa_s / \ell_v \times \kappa_v^{\ell_v}$;
- 6 The service queue length is
- 7 $\rho_{v,t+1}^s = \max\{\rho_{v,t}^s + s_{v \rightarrow s}^{b,t} - s_{v \rightarrow s}^{b,t}, 0\}$;
- 8 Estimate communication delay;
- 9 $s_{v \rightarrow s}^{b,t} = \text{hlog}_2 \left(\frac{1 + \phi_{p o g_{r,v}}^t}{(np)^2} \right)$;
- 10 Estimate computation delay with Eq. 17;
- 11 **if** $s_{v \rightarrow s}^{b,t} \leq \text{threshold} \ \&\& \ D \leq \text{threshold}$ **then**
- 12 Estimate the system cost ;
- 13 $\mathbb{C}_s^t = \sum_{v_s \in v} \left(c_{vn} \cdot s_{v \rightarrow s}^{b,t} \cdot \varpi_j^t + c_{df} (1 - q_v^t) \ell_v \right) + c_{se} \cdot L_s \cdot \nabla_j^t$;
- 14 **end**
- 15 **if** $\mathbb{C}_r^t \leq \text{threshold}$ **then**
- 16 Continue the execution with RSU;
- 17 **end**
- 18 **else**
- 19 Switch to other RSU or RSU updates the service through the server;
- 20 **end**
- 21 **end**
- 22 **else**
- 23 Offload the request to the feasible RSU when $\rho_{v,t+1}^s \leq 1 \ \&\& \ \text{Queue cost} \leq \Upsilon_{j,c}^t$;
- 24 **end**
- 25 **end**

accommodate offloading service. Subsequently, the Fog-RSU node feasibility is assessed based on the mutation of all parameter states. Let us assume K number of parameters considered to assess the node feasibility at time t and state of node is derived as follows

$$\begin{bmatrix} \text{state} & k_1 & k_2 & k_n \\ \partial & \sqrt{(1-p_{k_1})} & \sqrt{(1-p_{k_2})} & \cdots & \sqrt{(1-p_{k_n})} \\ \delta & \sqrt{p_{k_1}} & \sqrt{p_{k_2}} & & \sqrt{p_{k_n}} \end{bmatrix} \quad (19)$$

Now, the Deep Convolutional Network (DCN) is deployed to assess the node feasibility factor, and each neuron comprises associated weight and parameter value. The neuron denoted as $\omega_{k_1} \times \sqrt{p_{k_1}}$, where ω_{k_1} associated weight of parameter. The feasibility of node is assessed based on DCN, where

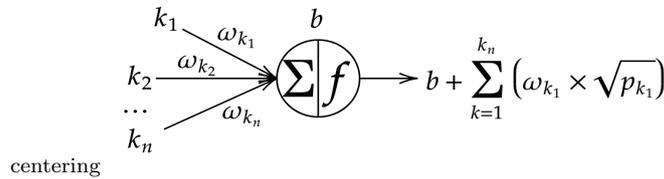


Fig. 2: Quantum based DCN

1 indicates a high potentiality rate to serve the offloaded services in terms of computation and communication cost, offloading cost parameter. These are evaluated using qubit individuals and the functional flow is illustrated in Fig. 3. The figure illustrates the quantum decision analysis mechanism to select the feasible node. In the first iteration, the status of all devices is measured based on node-centric parameters at the first step. In the second step, the individual qubits are

employed to assess the state of each parameter as mathematically described below. The qubit pooling process is initiated based on DCN considering parameter associated weight in step three. The third step significantly impacts analyzing the decision on service offloading based on conditional probability to select the feasible node.

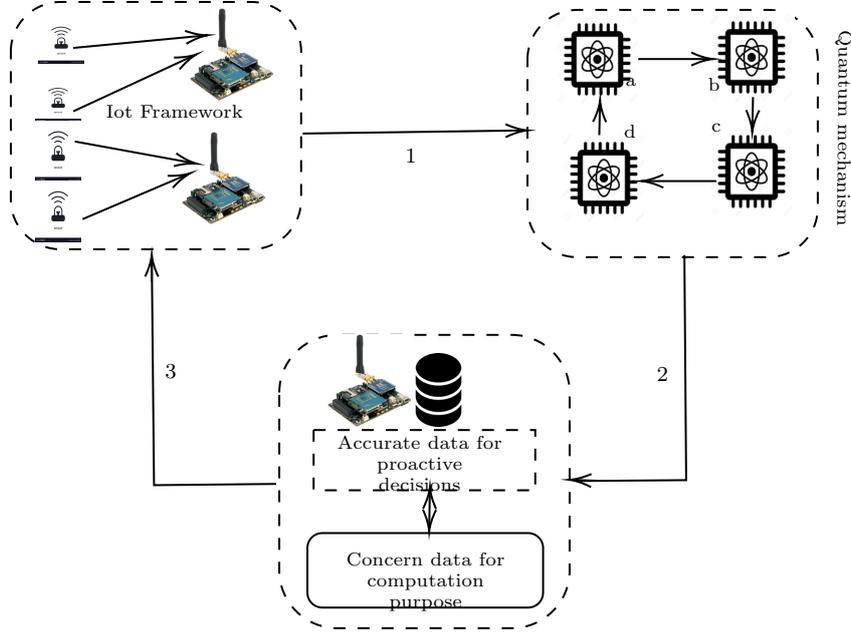


Fig. 3: Quantum decision analysis flow

3.2 Quantumizing bandwidth allocation strategy

Let us assume, LiDAR or mmWave sensors are deployed in IoT framework, and the underlying objective of the sensors is to share or transfer the sensed data to Fog-RSU node when the data require more computation resources for effective computation and communication. However it is important to note that, in this instance the vehicle device or on-board unit has been considered as incapable to process the data for such cases there is a need to enhance the bandwidth capacity of device. After the data has been received by the Fog-RSU node, the in-built computation system assess the objects and their spatial information. In other words, given objective is to identify the defeated vehicle and enhance its bandwidth capacity through QDC^2 approach to optimize the allocation based on few parameters as follows

$$\chi_{sb} = \xi_v^t \times \left(\sqrt{s_{v \rightarrow s}^{b,t}} \times \sqrt{D_{com,t}} \right) \quad (20)$$

where $\sqrt{s_{v \rightarrow s}^{b,t}}$, $\sqrt{D_{com,t}}$ are estimates through qubits as per the above procedure, and ξ_v^t is a spatial weight score factor which estimates based on number of objects and data accuracy rate. However, it is confined that, the estimation of object in each frame based on point-cloud data, ξ_v^t weight factor derivation is not concentrated in this example, but considered as pre-defined variable. The allocation index is formulated as follows

$$\chi_{sb} = \begin{cases} Allocation, & \leq Max(\chi_{sb}) \\ Continue, & Min(\chi_{sb}) \leq \chi_{sb} \leq Max(\chi_{sb}) \\ uncertain, & \leq Min(\chi_{sb}) \end{cases} \quad (21)$$

Table 1: Quantum analysis

Fog-RSU Node	parameter accuracy rate	Cost Error rate	$\sqrt{\text{error rate}}$	Normalized value	State-of-node
$s_a \rangle$	0.85	0.75	0.866025	0.736122	$\triangleq s_a 1 \rangle$
$s_b \rangle$	0.31	1.15	1.072381	0.332438	$\triangleq s_b 0 \rangle$
$s_c \rangle$	0.56	1.01	1.004988	0.562793	$\triangleq s_c 1 \rangle$
$s_d \rangle$	0.69	0.95	0.974679	0.672529	$\triangleq s_c 1 \rangle$

3.3 Service offloading theoretical Analysis

As discussed in the above sections, the individual qubits of each parameter are estimated to assess the state of the Fog-RSU node whether is feasible to accommodate the offloading service or not. In this regard, an example of a problem is derived for thorough analysis. Fig. 4 illustrates a four-quadrant graph with a circle that indicates the LAN coverage of fog-nodes, the eclipse coverage area indicates to offload the service without leveraging the parameters. In our example, this scenario is neglected because to resolve complex computations. A quantum gate (h-gate and CNOT gate) is considered to assess the possible solutions based on listed parameters as follows

$$Z(\theta) = \begin{bmatrix} \text{Cos}(\theta) & -\text{Sin}(\theta) \\ \text{Sin}(\theta) & \text{Cos}(\theta) \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} s'_a(\partial, \delta) \\ s'_b(\partial, \delta) \end{bmatrix} = Z(\theta) \begin{bmatrix} s_a(\partial, \delta) \\ s_b(\partial, \delta) \end{bmatrix} \quad (23)$$

Where $Z(\theta)$, θ refers rotation operator, rotation angle respectively. In our example, four angle values are considered. The angle change guides to search the suitable Fog-RSU node. The $s_a|state \rangle$ is one of the Fog-node among four except the red-point node. The red-point node is the node which offloads the service to the suitable node. Our objective is find out the suitable node. Assume $s_a|state \rangle$ node has 0.75% cost error rate and the parameter state is considered as 1 because the value is more than ≥ 0.5 , and parameter accuracy rate is 0.85%. In this example one parameter is considered to assess the feasible node and $\omega_{k_1} \times \sqrt{p_{k_1}}$ is used to assess the normalized value and Table 1 illustrates the analysed values. $s_a|state \rangle$ has 0.73% normalized value which is ≥ 0.5 and is equal to $\triangleq s_a|1 \rangle$ and node-a is feasible to accommodate the offloading services. In case the $s_a|state \rangle$ is busy in terms of huge service queue line, then second feasible node is $s_d|state \rangle$ because it has low cost error rate than other two devices and measurement accuracy is also feasible.

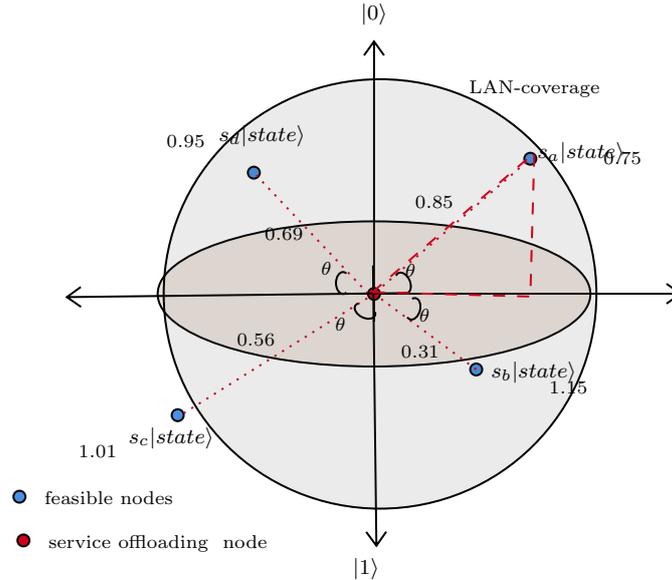


Fig. 4: Quantumized service offloading mechanism

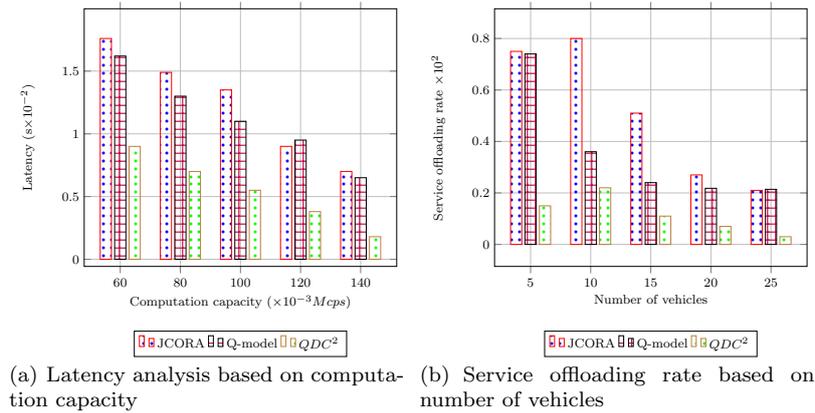


Fig. 5: Latency and service offloading analysis based on computation capacity and vehicle count

4 Experimental Results Analysis

The efficiency of the QDC^2 approach is examined by considering $s = 4$ fog nodes, $v = 25$ vehicle devices connected to a single edge router with a service arrival rate $\nu = \{1, 2, 3, \dots, 12\}$. Fifty sub-channels are considered because spectral bandwidth is $\varpi = 50$, and the DCN network comprises two hidden layers and ten neurons. The neuron or parameter count and accuracy are dependent, and several parameters consideration is dependent on the node capability. The computation capacity is measured based on CPU cycles/s count. The QDC^2 approach is examined in two levels (vehicle level and fog-level). The vehicle level consists of one edge router with 25 vehicles, and the fog level consists of vehicles+router+switch+base station. Framework topology is treated as constant in heterogeneous environments, and service offloading is being analyzed at the vehicle level and fog-level, and computing and communication abilities of devices are listed in Table 2. Two recent approaches (JCORA and Q-model) are considered to assess the performance

Table 2: Device computation and communication capacities

Device type	Computation	Communication
Vehicle	0.13	0.19
Base Station	0.5	1.5
Switch enabled Fog-RSU node	1.9	4
Gate-Way capacity	4.5	5.2

of the proposed system. Fig. 5 illustrates the latency and service offloading analysis based on computation capacity and vehicle count, respectively. The Fog-RSU node computation capacity and service execution rate are inversely propositional to each other. If computation capacity increases, then the service execution rate is drastically decreased because of estimating the node potentially before assigning the service load, as can observe in Fig.5(a). Quantum computing influences the measurement of the node-centric parameters by changing the rotation angle based on the nearby nodes. Fig. 5(b) illustrates the service offloading rate based on vehicle density. The capacity of a vehicle has been estimated through an individual qubit assessment procedure, which influences the execution of service at a local computing system called vehicle execution. In case the vehicle is not potentially rich with resources, then the vehicle offloads the services to a nearby Fog-RSU server, and the same conditional offloading strategy is applied to Fog-RSU nodes. However, it is essential to note that, in this scenario, we have not considered edge-vehicle computing. The offloading rate is drastically reduced as the increment of vehicle count from $V = 5 - 25$.

Fig. 6 illustrates service execution time analysis based on service arrival rate. QDC^2 approach has achieved low execution time of arrived services than the state-of-art approaches in Fig. 6(a). In our simulation, the service arrival rate is varied from 10-50, and the execution time has been drastically increased as increment in service arrival rate. The quantum computing integration has an immense impact on service execution time because the selection of Fog-RSU nodes is processed through qubit-parameter state analysis. The Deep-Convolutional Network is implied to assess the

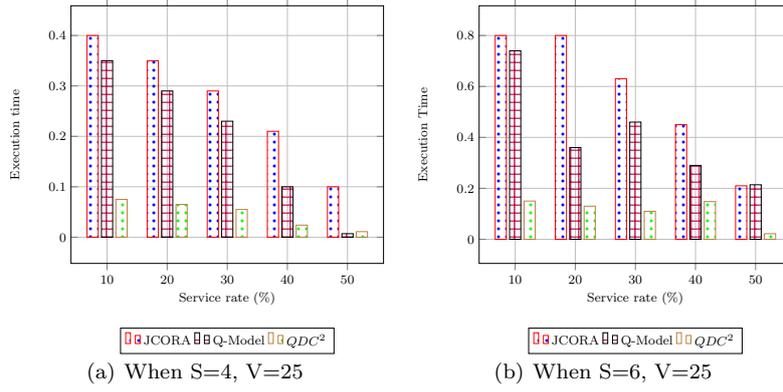


Fig. 6: Execution time analysis based on service arrival rate

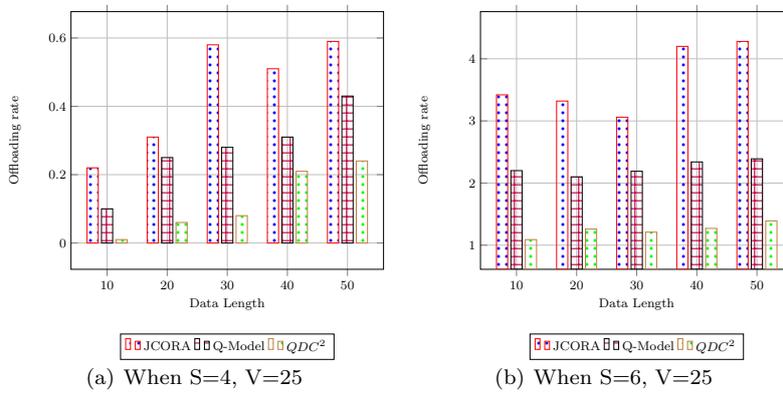


Fig. 7: Offloading analysis based on data size

Fog-RSU node by considering each layer outcome based on parameters state or neuron state as per the normalized condition described above. Fig. 6(b) illustrates the execution time of arrived services, but Fog-RSU nodes are 6 interconnected with LAN followed by 25 vehicles connected through a single router. QDC^2 approach has achieved a moderately low execution time of arrived services than the state-of-art approaches.

Fig. 7 illustrates service offloading analysis among state-of-art approaches and our proposed approach based on service data length. Our approach has achieved low offloading rate than state-of-art approaches. The offloading rate is slightly increased by increasing the service data length in scenario-1, which can be observed in Fig. 7(a). However, the offloading rate is optimal due to selecting the feasible node. In some cases, the node is potential, but the service queue length is high, which may lead to execution delay, and it has been resolved by prognosticating the device-centric parameters through individual qubits. Fig. 7(b) illustrates the second scenario-2, which comprises 6-RSU nodes and 25-vehicles. In this case, the QDC^2 has achieved a constant offloading rate because of 6-RSU nodes.

Fig. 8 illustrates QDC^2 approach performance analysis based on three measurement metrics such as Root Square Error rate, cost rate, latency rate, respectively. Fig. 8(a) illustrates offloading rate with and without the presence of QDC^2 approach. The offloading rate is notably high compared to the consideration of the QDC^2 approach. Fig. 8(b) illustrates the measurement metrics analysis concerning the service data length. RSE rate is low in all cases due to quantum computing and prognostic node-centric parameter measurements. While service offloading, the potential node selection is carried out based on several parameters and parameter weights. The DCN has implied to assess the state of each node, and this mechanism is neglected in state-of-art approaches. Subsequently, the cost is relatively high, but on average, it is low than other approaches and the latency also. Table 3 shows the FogSim simulator parameters such as service load, size, vehicle

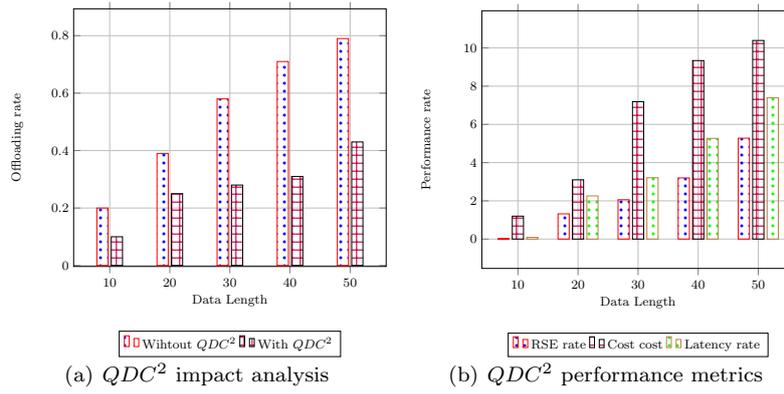


Fig. 8: QDC^2 performance analysis

Table 3: Fogsim simulation parameters

Service load	worth			
Size	≥ 9000			
Device density	25			
Length	$\leq 500 - 50000$			
Individual load	RAM Gb	Bandwidth (Mbps)	PE count	MIPS
Fog-RSU level	16	1000	32	30000
Vehicle level	8	100	16	3000

density. Additionally, the Fog-RSU and vehicle level load parameters are listed in terms of RAM capacity, bandwidth, Processing element count, MIPS count for services. Fig. 9 illustrates FogSim simulation outcomes, and Fig. 10 illustrates Rasbarry pi 4 Model-B setup, which has been used to assess the resource parameters. Subsequently, MATLAB simulation-link is employed to assess the performance of the proposed system.

```

m-1-2
m-0-7
m-1-3
m-0-8
On device m-0-0
{AppEdge [source=EEG, destination=client, tupleType=EEG]=0.1, AppEdge [source=connector, destination=client, tupleType=GLOBAL_GAME_STATE]=0.01,
-----
_SENSOR
AppEdge [source=client, destination=classifier, tupleType=_SENSOR]
client---->classifier : 0.1
GLOBAL_STATE_UPDATE
AppEdge [source=client, destination=DISPLAY, tupleType=GLOBAL_STATE_UPDATE]
client---->DISPLAY : 0.01
_SENSOR
GLOBAL_STATE_UPDATE
CLASSIFICATION
AppEdge [source=classifier, destination=client, tupleType=CLASSIFICATION]
classifier---->client : 0.1
_SENSOR
GLOBAL_STATE_UPDATE
SELF_STATE_UPDATE
AppEdge [source=client, destination=DISPLAY, tupleType=SELF_STATE_UPDATE]
client---->DISPLAY : 0.1
CLASSIFICATION
_SENSOR
GLOBAL_STATE_UPDATE
SELF_STATE_UPDATE
CLASSIFICATION
-----2
    
```

Fig. 9: Simulation outcome



Fig. 10: Raspberry Pi setup

5 Conclusion

The designed Quantum-based D2D Computation, Communication QDC^2 approach process the data generated from LiDAR sensor, millimetre-wave (mmWave) sensor at Fog-based RSU for efficient service reliability. The QDC^2 approach resolved the coated issues in two steps. The bandwidth allocation strategy has used the load and resource capacities of RSU or vehicle to decide on resource allocation. The adaptive equilibrium service offloading strategy effectively assessed the computation capacity and performance rate of Fog-node to streamline computation consistency issues. The data transmission delay has been reduced by 23% reduces with the efficient bandwidth allocation as per data importance. The adaptive equilibrium load distribution strategy assesses device-centric measurements to balance the computation resources and has minimized 21.54% average latency delay. Simulation results show that our approach achieved a 79.56% service reliability rate, 0.9% average error rate, and an execution delay is 22.5 s for 15 devices than state-of-art approaches.

Acknowledgments

This work was supported in part by Basic Science Research Programs of the Ministry of Education (*Grant NRF – 2018R1A2B6005105*) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A5A8080290).

Declarations

Data Availability

Data available on request from the authors.

Code Availability

Code available on request from the authors.

Conflict of Interest

The authors have no Conflicts of Interest to declare for this manuscript.

Authors' contributions

MS Mekala: Conceptualization, Methodology, Software, Data curation, Validation, Investigation, Visualization, Writing – original draft. **Gautam Srivastava:** Supervision, Conceptualization, Methodology, Investigation, Writing – review & editing. **Jerry Chun-Wei Lin:** Supervision, Methodology, Validation, Writing – review & editing. **Gaurav Dhiman:** Methodology, Writing – review & editing. **Ju H. Park:** Methodology, Validation, Writing – review & editing. **Ho-Youl Jung:** Methodology, Writing – review & editing.

References

1. Quoc-Viet Pham, Fang Fang, Vu Nguyen Ha, Md Jalil Piran, Mai Le, Long Bao Le, Won-Joo Hwang, and Zhiguo Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8:116974–117017, 2020.
2. Daojing He, Sammy Chan, and Mohsen Guizani. Security in the internet of things supported by mobile edge computing. *IEEE Communications Magazine*, 56(8):56–61, 2018.
3. Tie Qiu, Jiancheng Chi, Xiaobo Zhou, Zhaolong Ning, Mohammed Atiquzzaman, and Dapeng Oliver Wu. Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys & Tutorials*, 22(4):2462–2488, 2020.
4. Daniel Kudrow, Kenneth Bier, Zhaoxia Deng, Diana Franklin, Yu Tomita, Kenneth R Brown, and Frederic T Chong. Quantum rotations: a case study in static and dynamic machine-code generation for quantum computers. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 166–176, 2013.

5. Munish Bhatia and Sandeep K. Sood. Quantum computing-inspired network optimization for iot applications. *IEEE Internet of Things Journal*, 7(6):5590–5598, 2020.
6. Qi Liu, Weidong Cai, Jian Shen, Zhangjie Fu, Xiaodong Liu, and Nigel Linge. A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment. *Security and Communication Networks*, 9(17):4002–4012, 2016.
7. Yangyang Li, Zhenghan Chen, Yang Wang, Licheng Jiao, and Yu Xue. A novel distributed quantum-behaved particle swarm optimization. *Journal of Optimization*, 2017, 2017.
8. Shijie Dai, Minghui Liwang, Yang Liu, Zhibin Gao, Lianfen Huang, and Xiaojiang Du. Hybrid quantum-behaved particle swarm optimization for mobile-edge computation offloading in internet of things. In *International Conference on Mobile Ad-Hoc and Sensor Networks*, pages 350–364. Springer, 2017.
9. Sheetal N. Ghorpade, Marco Zennaro, Bharat S. Chaudhari, Rashid A. Saeed, Hesham Alhumyani, and S. Abdel-Khalek. Enhanced differential crossover and quantum particle swarm optimization for iot applications. *IEEE Access*, 9:93831–93846, 2021.
10. Sushil Kumar, Omprakash Kaiwartya, Manisha Rathee, Neeraj Kumar, and Jaime Lloret. Toward energy-oriented optimization for green communication in sensor enabled iot environments. *IEEE Systems Journal*, 14(4):4663–4673, 2020.
11. Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE transactions on evolutionary computation*, 6(6):580–593, 2002.
12. Erik P DeBenedictis. A future with quantum machine learning. *Computer*, 51(2):68–71, 2018.
13. Hung Viet Nguyen, Zunaira Babar, Dimitrios Alanis, Panagiotis Botsinis, Daryus Chandra, Mohd Azri Mohd Izhar, Soon Xin Ng, and Lajos Hanzo. Towards the quantum internet: Generalised quantum network coding for large-scale quantum communication networks. *IEEE Access*, 5:17288–17308, 2017.
14. Ahmed A Abd El-Latif, Bassem Abd-El-Atty, M Shamim Hossain, Samir Elmougy, and Ahmed Ghoneim. Secure quantum steganography protocol for fog cloud internet of things. *IEEE access*, 6:10332–10340, 2018.
15. Q. Wu, Y. Zeng, and R. Zhang. Joint trajectory and communication design for multi-uav enabled wireless networks. *IEEE Transactions on Wireless Communications*, 17(3):2109–2121, 2018.
16. M S Mekala and P. Viswanathan. Equilibrium transmission bi-level energy efficient node selection approach for internet of things. *Wireless Personal Communications*, 108(3):1635–1663, Oct 2019.
17. F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian. Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems. *IEEE Journal on Selected Areas in Communications*, 36(9):1927–1941, 2018.
18. S. Bi and Y. J. Zhang. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. *IEEE Transactions on Wireless Communications*, 17(6):4177–4190, 2018.
19. M S Mekala and P Viswanathan. Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for iot. *Computers & Electrical Engineering*, 73:227–244, 2019.
20. MS Mekala and Patan. Dawn: Cost-aware asset claim analysis approach on big data analytic computation model for cloud data centre. *Security and Communication Networks*, 2021, 2021.
21. M. S. Mekala, Alireza Jolfaei, Gautam Srivastava, Xi Zheng, Amjad Anvari-Moghaddam, and P. Viswanathan. Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pages 1–10, 2020.