International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

Energy Harvesting based Mobile Cloud Network in Latency and QoS Improvement using 5G Systems by Energy Routing Optimization

Kajol Khatri
MS- Computer Science,
University of Texas at Arlington, Texas
kxk7899@mavs.uta.edu

Dr. Anand Sharma
Asst. Prof., Department of CSE,
Mody University of Science and Technology, Lakshmangarh
https://orcid.org/0000-0002-9995-6226
anand_glee@yahoo.co.in

Abstract:
D2D communication technology enables the User Equipment (UE) in 5G networks to instantly connect with other UEs, with or without partial infrastructure involvement. In a Cloud Assisted energy harvesting system, it has improved user numbers and data transmission between mobile nodes. This research propose energy harvesting for mobile cloud computing in enhancing the QoS and latency of the network. The main aim of this research is to enhance energy optimization using discrete energy efficient offloading algorithm. The routing has been optimized using fuzzy logic cognitive Bellman-Ford routing algorithm. To identify the failing node and find an alternative node to deliver the seamless services, an unique weight-based approach has been presented. The method relies on two working node parameters: execution time and failure rate. Threshold values are specified for the parameters of the chosen master node. By contrasting the values with the threshold values, the alternative node is chosen. The experimental results shows comparative analysis in terms of throughput of 96%, QoS of 96%, latency of 28%, energy consumption of 51%, end-end delay of 41%, average power consumption of 41% and PDR of 85%

Keywords: 5G networks, energy harvesting, mobile cloud computing, energy optimization, routing.

1. Introduction:

By standardising the storage and processing of data, cloud computing (CC) is a technological development that increases the potential of IT systems [1]. It enhances access to private data stored in the cloud and enables users to access requests without first authenticating them [2]. Additionally, CC has a lower cost need for building IT infrastructures and acquiring cutting-edge assets. By maintaining a single application, the computers utilised in cloud computing services gain the benefit of multitenant structure. It depends on the ability to access computer resources through mobile devices. Similar to how regular desktop computers traditionally complete tasks, mobile computing does the same. In general, key concepts like software, hardware, and communication are used to support mobile computing [3]. Devices that are typically used as personal devices, such as PCs, tablets, and smartphones, are considered hardware. Software is made up of the programmes that are modelled after and created for mobile platforms. Communication encompasses networks and protocols that comprise concepts linked to communication techniques, to sum up [4]. The following are elements of the mobile computing strategy. The first component is mobility, which enables fixed or mobile nodes to link with nodes of other devices in a mobile computing environment utilising a mobile support station (MSS). Diversity of access network types is the next factor that enables communication between mobile and MSS nodes despite the presence of various access networks with differing bandwidths and overhead. Third, frequent network disconnects suggest that the mobile nodes lack the capacity to establish dependable connections because of their constrained resources, such as battery power and communication bandwidth. Finally, the signals of mobile nodes are vulnerable to interference in mobile networks because of the issue of poor consistency and safety [5]. The majority of outsourcing models for computing and hosting resources are included in the CC model. CC as defined by NIST [6], is the practise of efficiently accessing network resources while minimising management effort and time. Mobile CC, which enables clients to access limitless computing resources and storage space, is the combination of wireless communications, CC, mobile devices, mobile web, and other technologies for the effective transport of data. Data transport is significantly hampered by the

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

problem that affects CC and mobile devices both. In traditional cloud computing, cost and length are the most critical factors to take into account when transferring data, however in MCC, energy consumption and communication are crucial factors. In the case of a small device and while transporting an increased rate of information, a backup device is a useful service. By reducing power consumption while maintaining the highest level of quality for mobile services and applications, mobile device uptime can be increased. Context in which the mobile device runs, including network connection quality, location, and potential time and cost of executing application or service, should be taken into consideration when implementing such optimization, which can be done at the hardware or software levels. To increase quality metrics and optimise power consumption, usage of context information may enable mobile services and applications to adapt to the current conditions. The technique of adaptation can be used to both the mobile device itself as well as mobile services and applications used by device in order to optimise power consumption. These types of adaptations are frequently made using techniques that make use of remote resources, such as the cloud or a different mobile device with the necessary resources, to which applications, services, or their constituent parts are loaded in improve performance of mobile device. During software development process, the decision of when and what should be offloaded from mobile device can be made offline. It can also be made dynamically while the device is in use. Through adaptation, mobile devices' application/service execution times and costs can be decreased, and their online power consumption can be optimised.

The contribution of this paper is as follows,

1. To propose energy harvesting for mobile cloud computing in enhancing the QoS and latency of the network
2. To improve energy optimization using discrete energy efficient offloading algorithm
3. To enhance network routing using fuzzy logic cognitive Bellman-Ford routing algorithm and weight based method is proposed to detect failed node and discover the alternate node to provide seamless services

2. Review of literature:

Power management concerns are becoming more and more important, particularly in context of contemporary distributed systems, such as those that use virtualization as well as CC. [7] presents an examination of power-saving methods and looks at machine learning's potential in automatic power management systems. The paper does not, however, go into great detail about machine learning algorithms and features of potential adaptability. Mobile devices, which have grown to be a significant component of contemporary distributed systems, are not covered in the research, which solely takes into account desktop and similar systems. Since the inception of mobile gadgets, particularly mobile phones, the aspect of power consumption has been of utmost importance. Numerous articles on the subject have been published, including [8, 9] and some more recent studies where writers provide an overview of power management in context of mobile devices [10] and energy-saving techniques in context of mobile device apps [11]. The authors of [12] offer a variety of strategies that have been proposed in the literature for improving energy efficiency of mobile devices at software and hardware levels, including operating system-level power management, sensor and communication interface management, and cloud computing. The tactics that the creator of mobile applications can use are presented in [13] and include GUI design, sequential programmes, and mobile computation offloading. The potential for power savings on mobile devices employing CC is examined in further detail in certain publications, such [14]. The authors analyse the power consumption needed while offloading calculations to cloud utilizing mobile devices' network interfaces. A job power-aware scheduling approach, for instance, was suggested by Xu et al. [15] to lower HPC's electricity costs without lowering system utilisation. Our study, however, focuses on scheduling algorithms for mobile devices that are energy-conscious. Santinelli et al. [16] investigated ways to effectively lower real-time application power usage when resources are limited. Energy-Aware Scheduling by Minimizing Duplication [17], a novel approach put out by Mei and Li, takes into account both energy usage as well as makespan of programmes. In addition to the energy constraint, we also take performance and time constraints into account in our study. To fully take use of the advantages, Wang et al. [18] presented a method for variable partitioning as well as instruction scheduling. Their method, which aims to capture both performance as well as power demands, is based on a graph model. However, this method relied on a multiple memory architecture, but modern mobile devices only have a single memory. Other researchers [19] who wanted to optimise memory used genetic algorithms. A variety of runtime and compilation approaches were jointly disclosed in [20] by the authors to hide the heterogeneity of cellphones from developers.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

3. Proposed model:

This section discuss energy harvesting for mobile cloud computing in enhancing the QoS and latency of the network. The main aim of this research is to enhance energy optimization using discrete energy efficient offloading algorithm. The routing has been optimized using fuzzy logic cognitive Bellman-Ford routing algorithm. The proposed mobile architecture is shown in figure-1.

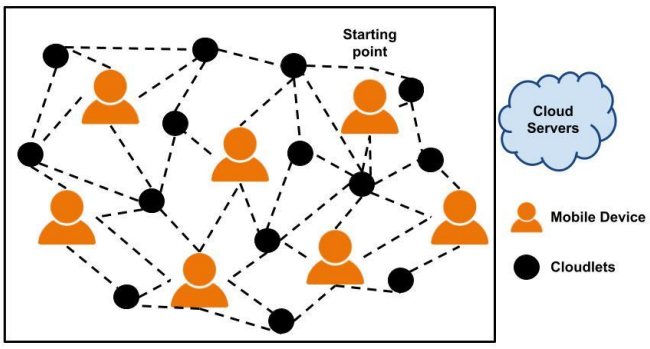


Figure-1 Proposed mobile architecture

In the MCC environment, a task of the mobile cloud is a relatively autonomous computation component that is raised inside a work flow and is executed on a cloud server or a mobile device. Every task requires an input that will be processed to provide an output. In this study, the mobile device's task is modelled as in equation (1),

$$T_{mob} = (A, Y, l) \tag{1}$$

Where l is the size of the task to be completed, A stands for the input, Y for the outcome, and A task flow in the mobile cloud is a process made up of a number of tasks that must be completed in a specific order in order to fulfil request of the mobile user. Equation(2) can be used to model how the mobile cloud's work flow operates,

$$D = (W, R) \tag{2}$$

W = {w1 ,w2 ,...wS ,...,wx} specifies the set of x mobile tasks, and R = {r(ws ,wt) ;ws ,wt ∈w} implies the relation between the tasks ws and wt with the requirement that the job ws must be finished before the work wt can be started. A mobile is typically a portable device, which can be a phone, laptop, or any other movable device with internet access and the ability to access cloud data sources. Equation (3) in this study models the mobile device using four sets of values,

$$S = \{u_m, p_i, p_l, p_{dl}\}; p_l \gg p_{dl} \tag{3}$$

Where I p denotes the mobile device's idle power consumption when it is in an idle state and m u denotes the frequency of operation when doing a task. Similar to this, the terms l p and pdl refer to the power needed to upload and download data from mobile devices, respectively. Equation (4) can be used to express the mobile device's power consumption,

$$P_c = \beta u_m^{\eta} \tag{4}$$

where the constants associated with the mobile device are β and η. The aforementioned expression shows that, under the condition that the operating voltage is proportional to frequency, power consumption is directly proportional to square of operating voltage and frequency. Communication time is 0 when the tasks, ws and wt, are equivalent and completed on a mobile device or in the cloud. When a work is completed on a mobile device while another task is being completed immediately after it on the cloud, the communication time is calculated as l st U l J /. Similar to this, the communication time is derived as dl st U dl J / when the job ws is completed on the cloud and the task immediately following it is completed on a mobile device. Equation (5) can be used to summarise this condition connected to communication time,

$$C_{ks}^{cm}(k_s, k_t) = \begin{cases} 0, & k_s = k_t = 0 \\ J_{st}^{l}/U_l, & k_s = 0, k_t = 1 \\ J_{st}/U_{dl}, & k_s = 1, k_t = 0 \\ 1, & k_s = k_t = 1 \end{cases} \tag{5}$$

In this equation, st l J stands for data transferred from mobile device to cloud, st dl J for data received from the cloud, and Ul and Udl, respectively, are uploading and downloading rates.

Energy optimization using discrete energy efficient offloading algorithm:

There are numerous single-antenna UEs and numerous MEC servers, as seen in Fig. 1. To get the necessary data and deliver the optimised local processing intensity to the serviced UEs, all MEC servers have access to etcd. The energy required to upload and retrieve the hardware information is negligible, and just a very little amount of hardware information is required for this paper.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

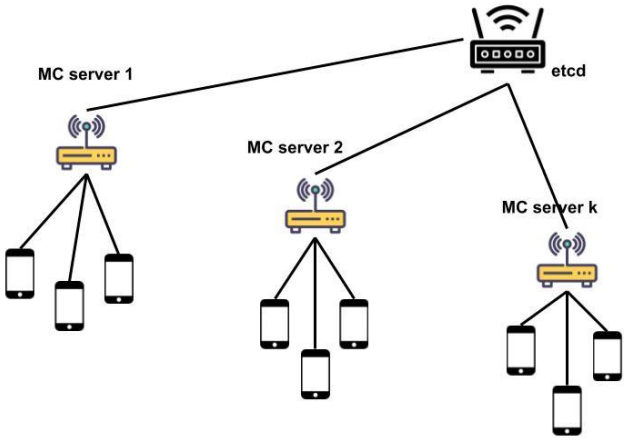


Figure 2. Multi-UE based MC network.

The sub-channels used by each UE cannot be shared, and the set of sub-channels is denoted by N = {1, 2, ..., N}. i,e,. by eqn (6)

$$C3: \sum_{i\in\mathcal{I}} w_{i,n} \leq 1 \; \forall n \in \mathcal{N} \tag{6}$$

The sub-channel n assigned to the UEi is shown by W = {Wi,n | wi,n ∈ {0, 1} , ∀i ∈ I, ∀n ∈ N }. K MC (mobile computing) servers in this case can only provide computing services to UEs that are physically within a certain radius. They must handle the computational tasks that the UEs have uploaded before returning the information and energy via N sub-channels. Consider that a single MC handles all of the calculations for each UE. Consequently, we have by eqn (7)

$$C4: \sum_{k\in\mathcal{K}} b_{i,k} \leq 1 \; \forall i \in \mathcal{I}$$

$$C5: 0 \leq \sum_{i\in\mathcal{I}} F_{i,k}^{\mathrm{UC}} \leq F_k \; \forall k \in \mathcal{K} \tag{7}$$

where $F^{\mathrm{UC}} = \left\{F_{i,k}^{\mathrm{UC}} \mid \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\right\}$ represents frequency that MECk spends in evaluating tasks of UEi . Let $B = \left\{b_{i,k} \mid b_{i,k} \in \{0,1\}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\right\}$ represents UE's offloading strategy. bi,k = 1 represents that UEi offloads computation task to MECk; otherwise, bi,k = 0. We assume that each UE in the MEC can continually receive energy from the MEC in order to guarantee that there is enough energy for each UE. Additionally, interference and noise are poor for ID but helpful for EH. So it makes sense and is legal to transmit energy to nearby computing UE. Let hi,n stand for sub-channel n gain with UEi receiving channel estimation error, and let σ 2 stand for the white Gaussian noise power. Further, we presumptively assume that channel gain stays constant throughout time [28]. Given that there are other energy-intensive activities besides computing, the received energy by UE must be greater than estimated energy consumption. This paper transmits energy to UE using maximum downlink power in order to prevent the UE's power from running out. The maximum energy that the UEi receives in uplink and downlink is denoted by eqn (8),

$$E_i^{\mathrm{EH}} = \zeta_i\left(\sum_{k\in\mathcal{K}} P_{i,k}^{\mathrm{DL}} t_i^{\text{tolerant}} h_{i,n,k}^{\mathrm{EH}} + \sigma^2\right) \tag{8}$$

Where $\zeta = \{\zeta_1, \zeta_2, \ldots, \zeta_i, \ldots, \zeta_I\}$ denotes energy absorption rate of all UEs. $P^{\mathrm{DL}} = \left\{P_{i,k}^{\mathrm{DL}} \mid \forall i \in \mathcal{I}; \forall k \in \mathcal{K}\right\}$ represents downlink transmission power between MECk and UEi . Additionally, gain of sub-channel employed for energy transfer between UEi and MECk is represented by h EH I n, k. Since every UE continuously receives energy, these channels are fixed. The latest time at which the UEi receives the calculation result is indicated by t tolerant i. It is expected that a polyphase bandpass filter is used by the UEi to receive signals from various subcarriers. The entire computation time when UEi chooses the local computing mode is given by eqn (9)

$$T_i^{\mathrm{LC}} = \frac{D_i X_i}{F_i^{\mathrm{LC}}} \tag{9}$$

where F LC I stands for the CPU's clock speed in UEi. For greater practicality, we established Fi as the top bound for the CPU's calculating frequency, which is comparable to equation (10)

$$C6: 0 \leq F_i^{\mathrm{LC}} \leq F_i \; \forall i \in \mathcal{I} \tag{10}$$

During local computing, each CPU cycle is proportional to square of F LC I and it may be expressed as eqn (11)

$$p_i^{\mathrm{LC}} = k_0\left(F_i^{\mathrm{LC}}\right)^2 \; [k_0 = 1 \times 10^{-24}] \tag{11}$$

amount of energy used by the UEi to execute a task can be expressed by eqn (12)

$$E_i^{\mathrm{LC}} = \left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right) p_i^{\mathrm{LC}} D_i X_i \tag{12}$$

Here, $\left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right)$ has the physical meaning of indicating whether UEi chooses local computing. Therefore, it is possible to define the number of sub-channels UEi uses to transmit computing jobs as $\left(\sum_{n\in\mathcal{N}^w} w_{i,n} - 1\right) R_{i,n}^{\mathrm{UL}}$ where denotes the sub-channel n's transmission rate and is given by equation (13)

$$R_{i,n}^{\mathrm{UL}} = B_N \log_2\left(1 + \frac{\sum_{n\in\mathcal{K}} b_{i,k} h_{i,n,k} P_{i,n}^{U\,\mathrm{L}}}{\sigma^2}\right) \tag{13}$$

Equation(14) can then be used to express the constraint

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

$$\text{C7: } \sum_{n\in\mathcal{N}} P_{i,n}^{\text{UL}} \le P_i^{\text{Max}} \; \forall i \in \mathcal{I} \tag{14}$$

The time it takes for the MECk to receive the computing work from the UEi can be stated as T UL I based on the premise that $R_{i,n}^{\text{UL}} \neq 0$, and the time it takes for the MECk calculation task is T UC i. They are each given as an equation (15)

$$T_i^{\text{UL}} = \frac{D_i}{\sum_{n\in\mathcal{N}} \left(w_{i,n} - \frac{1}{N}\right) R_{i,n}^{\text{UL}}}$$
$$T_i^{\text{UC}} = \frac{D_i X_i}{\sum_{k\in\mathcal{K}} F_{i,k}^{\text{UC}}} \tag{15}$$

Assuming t tolerable I is maximum computation time for both local computing as well as computation offloading, which is represented as equation(16), we can simplify the model,

$$C8: T_i^{\text{LC}} \le t_i^{\text{tolerant}} \; \forall i \in \mathcal{I}$$
$$C9: T_i^{\text{UC}} + T_i^{\text{UL}} \le t_i^{\text{tolerant}} \; \forall i \in \mathcal{I} \tag{16}$$

Equation(17) can be used to represent the uplink energy usage

$$E_i^{\text{UL}} = \sum_{n\in\mathcal{N}} P_{i,n}^{\text{UL}} T_i^{\text{UL}} \tag{17}$$

For the sake of simplicity, we make the assumption that computational processes consume a lot more energy than calculations that are optimised, and we disregard the energy consumption of sending and receiving devices. According to the analysis above, the objective function is written as P1. Since network is binary offloading, equation states that tasks offloaded by the UE equal the entire quantity of data flowing $\sum_{k\in\mathcal{K}} b_{i,k} D_i$ in the network (18).

$$C1: b_{i,k} \in \{0,1\} \; \forall i \in \mathcal{I}, \forall k \in \mathcal{K}$$
$$C2: w_{i,n} \in \{0,1\} \; \forall i \in \mathcal{I}, \forall n \in \mathcal{N}$$
$$\text{C3: } \sum_{i\in I} w_{i,n} \le 1 \; \forall n \in \mathcal{N}$$

$$\text{C4: } \sum_{k\in\mathcal{K}} b_{i,k} \le 1 \; \forall i \in \mathcal{I}$$

$$C5: 0 \le \sum_{i\in I} F_{i,k}^{\text{UC}} \le F_k \; \forall k \in \mathcal{K}$$
$$C6: 0 \le F_i^{LC} \le F_i \; \forall i \in \mathcal{I}$$
$$\text{C7: } \sum_{n\in\mathcal{N}} P_{i,n}^{\text{UL}} \le P_i^{\text{Max}} \; \forall i \in \mathcal{I}$$

$$C8: T_i^{\perp\text{C}} \le t_i^{\text{tolerant}} \; \forall i \in \mathcal{I}$$
$$C9: T_i^{\text{UC}} + T_i^{\text{UL}} \le t_i^{\text{ollerant}} \; \forall i \in \mathcal{I}$$
$$C10: E_i^{\text{UL}} + E_i^{\text{LLC}} \le E_i^{\text{EH}} \forall i \in \mathcal{I}$$
$$C11: \left(1 - \sum_{k\in K} b_{i,k}\right) \ln\left\{\sum_{n\in\mathcal{N}} w_{i,n}\right\} = 0 \; \forall i \in \mathcal{I}$$

$$C12: 0 \le \sum_{k\in\mathcal{K}} b_{i,k} \ln\left\{\frac{\sum_{n\in\mathcal{N}} w_{i,n}}{2}\right\} \; \forall i \in \mathcal{I} \tag{18}$$

The MINLP and non-convex energy efficiency optimization model that was suggested in the preceding section. To make the issue solvable and reduce complexity, the original model will be mathematically simplified in this part. We divide numerator as well as denominator by T UL to reduce complexity of the objective function. P1 becomes P2 by equation(19) following the operation.

$$\frac{E_i^{\text{LC}}}{T_i^{\text{UL}}} = \left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right) k_0 \left(F_i^{L\,\text{L}}\right)^2 X_i \sum_{n\in\mathcal{N}} \left(w_{i,n} - \frac{1}{N}\right) R_{i,n}^{\text{UL}} \left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right) \sum_{n\in N}^{\infty} \left(w_{i,n} - \frac{1}{N}\right) \sum_{k\in\mathcal{K}^L} b_{i,k} \tag{19}$$

In case of local computing, term $\left(1 - \sum_{k\in\mathcal{K}} b_{i,k}\right)$ is always equal to zero. Else, $\sum_{\text{n}\in\mathcal{N}} \left(w_{i,n} \cdot \frac{1}{N}\right)$ is always 0. The$\sum_{k\in\mathcal{K}} b_{i,k}$ can likewise be eliminated in a similar manner. At present time, independent of UE of local computing, objective function's practical meaning is to maximise energy efficiency of UE of computation offloading. UEs will become unbalanced as a result. As stated in equation(20), we incorporated the energy efficiency of the UE of local computing to objective function in order to give the best services to all UEs

$$\text{P3: } \max_{P_{i,n}, w_{i,n}, b, k, F_i} \sum_{i\in I} \left(\frac{\sum_{n\in N} \left(w_{i,n} - \frac{1}{N}\right) R_{i,n}^{\text{UZ}}}{\sum_{n\in N} P_{i,n}^{\text{UL}}} + \frac{\left(1 - \sum_{k\in K'} b_{i,k}\right) D_i}{E_i^{LC}} \right)$$
$$\text{s.t. } C1 \sim 12 \;) \tag{20}$$

In P3, t tolerant I T UC I influences the value of T UL I and t tolerant I T UL I limits T UC i. C12 is obviously a constraint that has physical significance but cannot be solved. To limit minimum value of RUL, we therefore employ additional constraints in place of C12. C12 is obviously a constraint that has physical significance but cannot be solved. To limit minimum value of RUL, we therefore employ additional constraints in place of C12.

$$C1 \sim 8{,}10 \sim 12$$

$$C9: R_{i,n}^{\min} \le R_{i,n}^{\text{UL}} \; \forall i \in \mathcal{I}, \forall n \in \mathcal{N}$$

$$\text{s.t. } C1 \sim 8{,}10 \sim 12 \tag{21}$$

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

where the minimum transmission rate for each subchannel is Rmin i,n. Because each sub-channel in this paper will be fully utilised by equation (22), there is not need to limit the overall rate of UE in this article.

$$\text{P4a}: \max_{F_i \in} \sum_{i \in I} \left[\left(1 - \sum_{k \in \mathcal{K}} \bar{b}_{i,k}\right) - q_i^F k_0 X (F_i^{LC})^2 \right]$$

$$\text{s.t. } C6, C8, C10$$

$$\text{P4b}: \max_{P_{t,n}^{\text{a}}} \sum_{i \in \mathcal{I}} \left[\sum_{n \in \mathcal{N}} \left(\tilde{w}_{i,n} - \frac{1}{N} \right) R_{i,n}^{\text{LH}} - q_i^{\text{p}} \sum_{n \in \mathcal{N}} P_{i,n}^{\text{UU}} \right] \quad (22)$$

$$\text{s.t. } C7, C9 \sim 10$$

The auxiliary variables q P I and q F I are introduced by the method and tend to expand progressively as iteration progresses. By using equation(23), we specify the error judgement functions of P4a and P4b as $F(F_i^{LC})$ and $F\left(P_{i,n}^{UL}\right)$

$$F(F_i^{LC}) = \sum_{i \in \mathcal{I}} (q_i^P[j] - q_i^F[j-1]) \quad (23)$$

This is employed to assess how accurate the computation result was. Their values will gradually drop until the accuracy standards are satisfied. Similar to before, equation(24) allows model P4b to become P5b.

$$\text{P5b.1 } \min_{P_{t,-\infty}} \sum_{i \in \mathcal{I}} \left(-\sum_{n \in \mathcal{N}} \left(\bar{w}_{i,n} - \frac{1}{N} \right) R_{i,n}^{UL} + q_i^{\text{p}} \sum_{n \in \mathcal{N}} P_{i,n}^{UL} \right.$$

$$\left. + \lambda_i^{\text{p}} \sum_{n \in N} \left(P_{i,n}^{\text{LI}} - P_{i,n}^{i\,\text{L}} \right) + \frac{\rho_P}{2} \sum_{n \in N} \left\| P_{i,n}^{\text{UL}} - P_{i,n}^{i\,\text{L}} \right\| \right)$$

$$\text{s.t. } C9: R_{i,n}^{\min} \le R_{i,n}^{\text{il}} \; \forall i \in \mathcal{I}, \forall n \in \mathcal{N}$$

$$C10: E_i^{\text{TI}} \le E_i^{\text{iH}} \forall i \in \mathcal{I}$$

$$\text{P5b.2 } \min_{P_{i,n}^{iL}} \sum_{i \in \mathbb{I}} \left(\lambda_i^{\text{p}} \sum_{n \in N} \left(P_{i,n}^{\text{PL}} - P_{i,n}^{iL} \right) + \frac{\rho_P}{2} \sum_{n \in \mathcal{N}} \left\| P_{i,n}^{UL\cdot} - P_{i,n}^{iU} \right\| \right) \quad (24)$$

Allocation of the remaining funds to P UL i,n using the same process after letting P UL i,n satisfy C7. P is the penalty function's coefficient in computation offloading model. λ P i is Lagrangian coefficient related to update equation (25) and the constraint P UL i,n = P UL i,n

$$\lambda_i^{\text{p}}[t+1] = \lambda_i^{\text{p}}[t] + \rho_P \sum_{n \in N} \left(P_{i,n}^{\text{UL}} - P_{i,n}^{\text{iL}} \right) \quad (25)$$

The Lagrangian functions L 1 P (P UL i,n) and L 2 P (P UL i,n), which correspond to P5b.1 and P5b.2, are shown in formula (26).

$$L_P^1\left(P_{i,n}^{UL}\right) = \sum_{i \in I} \left(-\sum_{n \in N} \left(w_{i,n} - \frac{1}{N} \right) R_{i,n}^{UL} + q_i^{\text{p}} \sum_{n \in \mathcal{N}} P_{i,n}^{U\,\text{L}} + \lambda_i \sum_{n \in N} \left(P_{i,n}^{U\,\text{L}} - P_{i,n}^{\text{UL}\cdot} \right) + \frac{\rho_p}{2} \sum_{n \in N} \left\| P_{i,n}^{UU\cdot} - P_{i,n}^{\text{UL}} \right\| \right) \quad (26)$$

The update formula of P UL i,n and P UL i,n can be written as equation(27) by computing the partial derivative,

$$P_{i,n}^{\text{UL}} = \left[\sqrt{\frac{X^2}{4} - \frac{G}{\rho_P \sum_{n \in \mathcal{K}} b_{i,k} h_{i,n,k}}} - \frac{X}{2} \right]^+$$

$$P_{i,n}^{\text{UL}} = \left[P_{i,n}^{\text{UL}} + \frac{\lambda_i - \gamma_{i,n}^{\text{PI}}}{\rho_P} \right]^+ \quad (27)$$

where by equation (28)

$$X = \frac{\lambda_i + q_i^{\text{P}} + \gamma_i^{\text{P3}}}{\rho_P} + \frac{\sigma^2}{\sum_{n \in \mathcal{K}_{i,k}} h_{i,n,k} N} - P_{i,n}^{\text{J\v{L}}}$$

$$G = \left(1 + \gamma_{i,n}^{\text{P}} + \frac{\gamma_i^{\text{P}} E_i^{\text{EH}}}{D_i}\right) \sum_{n \in \mathcal{K}} \bar{b}_{i,k} h_{i,n,k} \left(\bar{w}_{i,n} - \frac{1}{N} \right) \frac{B_N}{\ln 2} + \frac{\sigma^2 (\lambda_i + q_i^{\text{p}} - P_{i,n} \rho_p + \gamma_i^{\text{P3}})}{N} \quad (28)$$

$\gamma_{i,n}^{\text{P1}}, \gamma_{i,n}^{\text{P2}}$ and γ_i^{P3} represent Lagrangian coefficient of constraint $C7, C9 \sim 10$, and iterative formula is written by equation (29)

$$\gamma_{i,n}^{\text{P1}} = \left[\gamma_{i,n}^{\text{P1}} + \nabla_{P1} \left(P_{i,n}^{U,L} - P_{i,n}^{\text{Max}} \right) \right]^+$$

$$\gamma_{i,n}^{\text{p2}} = \left[\gamma_{i,n}^{\text{p2}} + \nabla_{P_2} \left(R_{i,n}^{\min} - R_{i,n}^{\text{UL}} \right) \right]^+$$

$$\gamma_i^{\text{P3}} = \left[\gamma_i^{\text{P3}} + \nabla_{P3} \left(\sum_{n \in \mathcal{N}} P_{i,n}^{\text{UL}} - \frac{\sum_{n \in \mathcal{N}} \left(\bar{w}_{i,n} - \frac{1}{N} \right) R_{i,n}^{UL} E_i^{\text{EH}}}{D_i} \right) \right]_{(41)}^+ \quad (29)$$

also, $\nabla_{P1}, \nabla_{P2}, \nabla_{P3}$ is weight of update step.

Fuzzy logic cognitive Bellman-Ford routing algorithm:

Path delay and cost (path distance) are the input parameters (average end-to-end delay). Knowledge base, neurofuzzy engine, and decision support system make up the architecture. As shown in figure 3, the knowledge base is made up of a knowledge repository, fuzzy logic, and neural networks.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

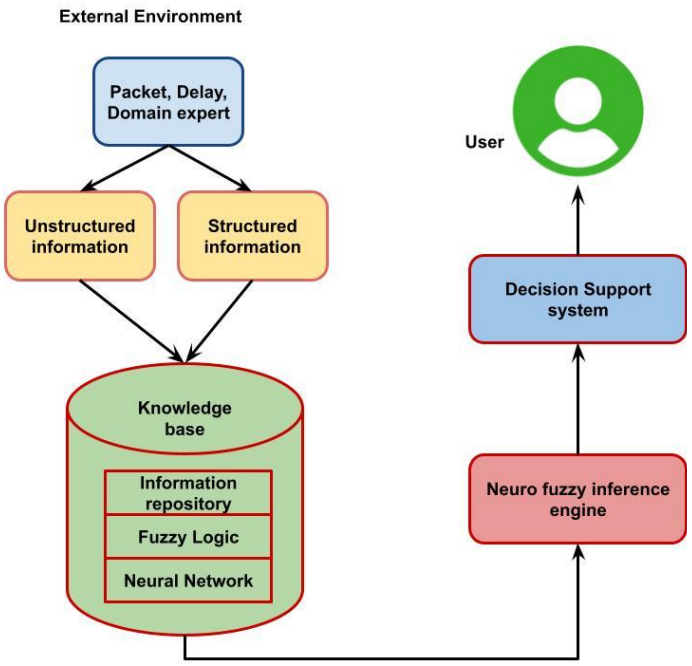


Figure-3 Architecture of Neuro-Fuzzy System

According to equation (30), a route is a path or paths that a message takes to get from its source to its destination.

$$\text{Route}_{\ i} = \sum_{i=1}^{n} \text{path}_{\ i}$$

$$\text{where i} = 1,2, \ldots, \text{n.} \tag{30}$$

The packet (message) and delay are represented in the knowledge base by equation (31):

$$\text{Packet}_{\ S,D} = \sum_{i=2,j=1}^{n,m} \text{message\_path}_{\ i,j} \tag{31}$$

where m is the maximum number of routes j that can be taken from the source (S) to the destination and n is the size of the message that needs to be routed through any potential path(s) between each node (D). Equation (32) is the condition we establish for packet routing.

$$\text{Packet} = \begin{cases} 0 \text{ if } a < M_i < b \nabla \{BR_i\} \in a \cup \{BR_i\} \in b; a, b > 0 \\ 1 \\ \text{Otherwise} \end{cases} \tag{32}$$

where Packet represents the message on a certain route and BRj denotes the amount of bandwidth on each Route. Subtracting the new delay from the previous value of the delay for the path given by equation(33) yields the increase or decrease in the delay that occurred.

$$\text{Delay path}_{\ k,j} = \sum_{i,k,j=1}^{n,m} \text{Delay}_{\ path_{i-k,j}} \tag{33}$$

where i-k is the time difference between path I and path k. Equation (34) represents the routing of delay circumstances.

$$\text{Delay} = \begin{cases} 0 \text{ if Delay}_{\ R_{im}} > \text{Delay}_{\ R_{j,m}} \\ 1 \qquad \text{Otherwise} \end{cases} \tag{34}$$

There won't be any routing if condition is 0, but there will be if it's not 0. The likelihood that at least one node in candidate set Ci will receive the data supplied by node I is determined by equation (35) if probability of successful transmission between nodes I and j is peij.

$$pc_i = 1 - \prod_{j \in C_i} (1 - pc_{ij}) \tag{35}$$

We define the transmission cost as follows using the significance model. According to equation (36): The more crucial a node is to preserving network's transmission performance, greater transmission cost.

$$\cos\ t_i = \frac{\lambda_i + 1}{\lambda_{\max}} \tag{36}$$

where λi is the node's algebraic connection. Additionally, λmax stands for the network's maximal algebraic connectedness. Ti transmissions are required for a single anycast transmission from node I to candidate set Ci. The anticipated transmission count is ti. Equation (37) gives the transmission cost of node I transmitting Ti times.

$$Ct_i c_i = \frac{1}{T_i} \times \cos\ f_i = \frac{1}{\left(1 - \prod_{j \in C_i} (1 - pe_{ij})\right)} \times \frac{(\lambda_i + 1)}{\lambda_{\max}} \tag{37}$$

Assume that the ascending order of candidate set for node i Ci = {ci1, ci2, . . . , cin}, which is dependent on the transmission cost CQ1 < CQ2 < . . . < CQn. Node ci1 will be considered the next hop to forward data by equation if it receives the data packet before any other candidate nodes, regardless of whether they also receive the data packet by eqn (38).

$$\begin{aligned} Cq_{ic1} &= \alpha \times \left(E_{TX} + E_{RX}(i, c_{i1})\right) \times \frac{1}{T} \times \text{cost}_i \\ &= \alpha \times \left(E_{TX} + E_{RX}(i, c_{i1})\right) \times \frac{\beta e_{i1}}{\left(1 - \prod_{j \in c_i} (1 - pe_{ij})\right)} CQ_1 \end{aligned} \tag{38}$$

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

where, $\frac{pe_{i1}}{\left(1-\prod_{j<c_{ij}}(1-p_{ij})\right)}$ is probability that data are successfully received from node i to node ci1. α is weight parameter. If it is not properly received, second-priority node is chosen to transmit. The remaining opportunity routing transmission cost is determined by equation (39) if the candidate node $c_{i1} \sim c_{iH-1}$did not successfully receive packet until data are received and transmitted by node u:

$$Cq_{icu} = \alpha\left(E_{TX} + E_{RX}(i, c_{i1})\right) \times \frac{\prod_{l=1}^{u-1}(1-pe_{il})pe_{iu}}{\left(1-\prod_{j\in C_i}(1-pe_{ij})\right)} CQ_u \quad (39)$$

$\frac{\prod_{l=1}^{u-1}(1-pe_{il})pe_{iu}}{\left(1-\prod_{j\in C_i}(1-pe_{ij})\right)}$ is probability that data are sent from node i to $c_{i1} \sim c_{iu-1}$failed and received successfully by ciu. Each forwarding node in opportunistic routing performs opportunistic forwarding by default. Packets may be sent along the wrong path if forwarding priority is only based on cost of a single node's transmission. This study uses the transmission cost of candidate set associated with present candidate node as an influencing element, taking into account the interaction between links. As a result, equation (40) provides the transmission cost CqiCi of the remaining opportunistic routing:

$$\begin{aligned} Cq_{ic_i} &= \sum_{j=1}^{n} Cq_{icj} \\ &= \frac{1}{\left(1-\prod_{p\in C_i}(1-pp_{ij})\right)}(\alpha \\ &\times (E_{TX} + E_{RX}(i,c1)) \times pe_{ic1} CQ_1 \\ &+\sum_{j=2}^{n} \alpha \times (E_{TX} + E_{RX}(i,cj)) \times pe_{icj} CQ_j \prod_{l=1}^{j-1}(1-pe_{idl})\Big) \end{aligned} \quad (40)$$

According to equation (41): CQi = CtiCi + CqiCi, the projected transmission cost from node I to sink is total of anycast transmission cost and remaining opportunistic routing transmission cost.

$$\begin{aligned} CQ_i &= \mathrm{Ct}_{ic_i} + \mathrm{Cq}_{ic_i} \\ &= \frac{1}{\left(1-\prod_{jc_i}(1-pc_{ij})\right)} \times \left(\frac{\lambda_i + 1}{\lambda_{\max}}\right) + \\ &\frac{1}{\left(1-\prod_{ic_i}(1-pe_{ip})\right)}(\alpha \times E_{TX} + E_{R\times}(i,c1)) \times pe_{ic1} Dc_i^1 + \\ &\sum_{j=2}^{n} \alpha \times (E_{TX} + E_{R\times}(i,cj))\Big) \times pe_{icj} Dc_i \prod_{i=1}^{j-1}(1- \\ &pe_{idl})\Big) \end{aligned} \quad (41)$$

The nodes in set A are those whose transmission costs decreased in the previous iteration, while the nodes in set B are those whose transmission costs decreased in the current iteration. The following is the improved algorithm for creating the candidate set:

(1) Let $\mathrm{CQ}_{\mathrm{sink}} = 0, CQ_i = +\infty, 2 \le I \le N$,and add sink to set A. Start method parameters, $h = 1, \mathrm{CQ}_1{}^0 = 0, \mathrm{CQ}_i{}^0 = +\infty, 2 \le i \le N$.

(2) Transmission cost of node I in A is computed in h-th iteration. Node j is included in candidate set if its neighbour node j joins it and reduces the transmission cost CQj h. Node I is simultaneously evaluated to see if it belongs to set B; if not, node I is added to set B.

(3) Method ends when candidate set of all nodes stops changing; otherwise, next step is carried out.

(4) Set B should be empty before set A is filled with set B's components. Let CQ1 0 = CQh 0 for each node I and then let h = h + 1 to proceed to step 2.

As a discrete time index in descending order, we denote t. If st bits are communicated during time slot t, transmission energy cost is $\mathcal{E}_t(s_t, g_t) = \lambda \frac{s_t^n}{g_t}$. For the best data-transmission schedule, optimization issue in Eq. (42) is recast as,

$$\begin{aligned} &\min_{s_t}: \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{E}_t(s_t, g_t)\right] \\ &\text{s.t.: } \sum_{t=1}^{T} s_t = Ls_t \ge 0, \forall t. \end{aligned} \quad (42)$$

Whether the channel is in the G or B condition at time t = T + 1 determines the least predicted energy. We integrate this requirement into notation of ideal amount of data bits transferred in every time slot s∗ t and least energy E∗ c in order to make the condition gT +1 = G and gT +1 = B easier

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

to convey. lt is the quantity of incomplete bits at time slot t. Optimal data scheduling vector is given by equation (19) for optimal data transmission scheduling issue (43)

$$s_t^*(l_t, g_t; G) = \begin{cases} l_t \left[\dfrac{(g_t)^{\frac{1}{n-1}}}{(g_t)^{\frac{1}{n-1}} + \left(\frac{1}{\zeta_{t-1;G}}\right)^{\frac{1}{n-1}}} \right], & t \geq 2, \\ l_1, & t = 1, \end{cases} \tag{43}$$

if $g_{T+1} = G$, where by equation (44)
and

$$s_t^*(l_t, g_t; B) = \begin{cases} l_t \left(\dfrac{(g_t)^{\frac{1}{n-1}}}{(g_t)^{\frac{1}{n-1}} + \left(\frac{1}{t_{t-1;B}}\right)^{\frac{1}{n-1}}} \right), & t \geq 2, \\ l_1, & t = 1, \end{cases}$$

if $g_{T+1} = B$, where

$$\zeta_{t;B} = \begin{cases} p_{BB} \left[\left(\dfrac{1}{(g_B)^{\frac{1}{n-1}} + \left(\frac{1}{\zeta_{t-1;B}}\right)^{\frac{1}{n-1}}} \right)^{n-1} \right] \\ + p_{BG} \left[\left(\dfrac{1}{(g_G)^{\frac{1}{n-1}} + \left(\frac{1}{t_{t-1;B}}\right)^{\frac{1}{n-1}}} \right)^{n-1} \right], \\ p_{BB} \left[\frac{1}{g_B}\right] + p_{BG} \left[\frac{1}{g_G}\right], \quad t \geq 2, \\ t = 1. \end{cases} \tag{44}$$

In accordance with this, the minimal transmission energy indicated by equation (45)

$$\mathcal{E}_c^*(L, T; G) = \lambda L^n \zeta_{T;G},$$

$$\mathcal{E}_c^*(L, T; B) = \lambda L^n \zeta_{T;B} \tag{45}$$

Equation (46) thus gives the minimal predicted transmission energy E*c.

$$\mathcal{E}_c^*(L, T) = \frac{T_G}{T_G + T_B} \mathcal{E}_c^*(L, T; G) + \frac{T_B}{T_G + T_B} \mathcal{E}_c^*(L, T; B) \tag{46}$$

4. Experimental analysis:

In order to evaluate the effectiveness of both sleep scheduling policies, numerical findings are presented in this section. As part of our benchmark designs, we also utilize three multiple access methods without sleep scheduling. The following are some of the technological features of the computer used during the implementation phase: Operating System: Windows 7 Pro 64-bit; Central Processing Unit: Intel(R) Core(TM) i7-4610M CPU @ 3.00GHz 3.00GHz; Random Access Memory: 8 GB.

Table 1. WSN simulation parameters

Parameter	Value
Number of clusters	5
Number of nodes	100 nodes
Network area	100mx100m
Size of data packet	500 bytes
Size of packet header	25 bytes
Sink location	(50,175)
Routing protocol	LEACH
Simulation time	3600s

Parameters Settings: The distance between each MTCD and BS or power station is assumed to be same12 for the nth CD in our simulation scenario, which is depicted in table-1, i.e., d PM n = d MB n. The distance set is denoted by the formula d MB = d MB 1, , dMB N, which represents a linearly spaced vector in interval [3, 5]m and meets the condition $d\ M{-}B\ 1 < \cdots < dM{-}B\ N$. TGn path-loss factor is utilised, and the usable bandwidth is 5MHz with a carrier centre frequency of 470MHz. N0 = 60dBm/Hz is the noise power spectral density. EH has an efficiency of ηp = 0.9. The EH time is a constant 10ms, while the maximum transmission time is τb, = 40ms. Additionally, values of Φr, and $\Phi t,max$ are 4W and 400mW, respectively. An active CD in an EH or transmission mode has a probability of $\rho = \lambda n \tau b$. As a result, we define average power consumption for sleep-scheduling-based protocols as follows:

$$\bar{\Phi} = \rho \left(\frac{\tau_b - \tau_p}{\tau_b} \Phi_t + \Phi_w \right) + \left(\frac{1 - \rho}{\tau_s} \right) (\Phi_s \tau_i + \Phi_{sc} \tau_{sc})$$

where Φw = 100mW, Φs = 10mW and Φsc = 100mW, respectively, stand for active mode, idle mode, and switching cost power consumption.

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

Table- 2 Comparative analysis of Energy Consumption

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	81	75	65
50	78	71	62
75	75	68	61
100	72	65	58
125	71	64	55
150	69	61	54
175	67	58	52
200	64	55	51

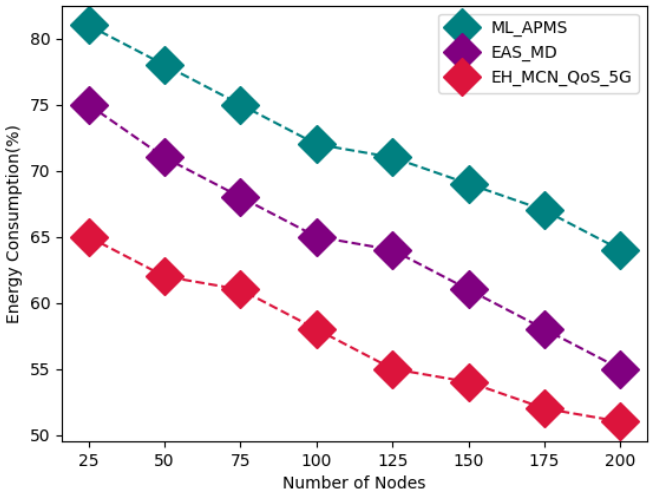


Figure-4 Comparative analysis of energy consumption

The above table- 2 shows comparative analysis between proposed and existing technique in terms of energy consumption based on number of nodes. Energy consumption refers to the overall amount of energy used by the network to carry out data aggregation, transmission, and reception. The comparisons made between the various methods were based on how much energy was used by the cluster member and cluster head sensor nodes. Here the proposed technique attained energy consumption of 51%, while existing ML_APMS attained 64% and EAS_MD obtained 55% of energy consumption of the network for 200 nodes as shown in figure-4.

Table-3 Comparative analysis of Throughput

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	71	75	79
50	75	79	83
75	77	81	85
100	79	83	86
125	81	85	89
150	83	88	91
175	85	89	93
200	88	91	96

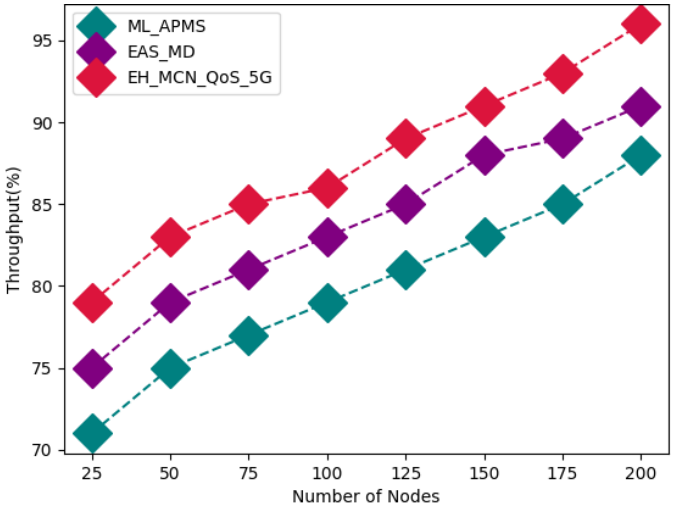


Figure-5 Comparative analysis of throughput

Above table-3 shows comparative analysis of throughput between proposed and existing technique based on number of nodes. Here proposed technique is compared with existing ML_APMS and EAS_MD. Network throughput in data transmission is the amount of data successfully transferred from one location to another in a predetermined amount of time. It is commonly measured in bits per second (bps), such as megabits per second (Mbps) or gigabits per second (Gbps). Proposed technique attained throughput of 96% for 200 nodes while ML_APMS attained 86% and EAS_MD obtained 91% for 200 nodes as shown in figure-5.

Table- 4 Comparative analysis of End-End delay

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	75	68	65
50	74	65	62
75	71	62	58
100	68	60	55
125	65	59	51
150	62	57	45
175	61	55	43
200	59	51	41

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

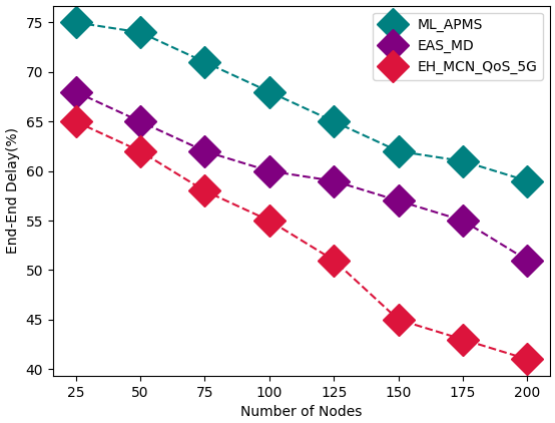


Figure- 6 Comparative analysis of End- end delay

The above table-4 represents comparative analysis of end-end delay between proposed and existing technique. When store-and-forward packet switches are employed, the formula for end-to-end delay for sending a single packet of length L over N connections, each with a transmission rate R, is d = N*L/R. (ignoring queuing, propagation delay, and processing time). The proposed technique attained end-end delay of 41%, while existing ML_APMS protocol attained 59% and EAS_MD obtained 51% for 200 nodes as shown in figure-6

Table- 5 Comparative analysis of Packet delivery ratio

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	55	59	62
50	59	63	65
75	61	65	69
100	63	68	72
125	65	71	75
150	68	73	79
175	71	75	83
200	73	79	85

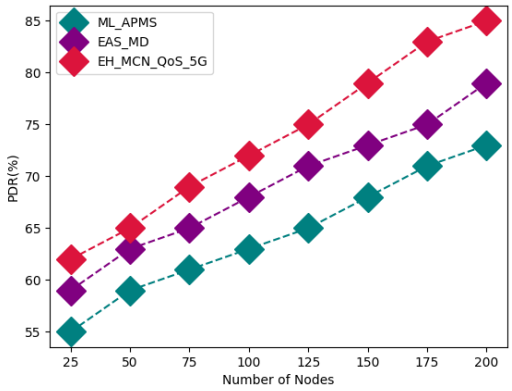
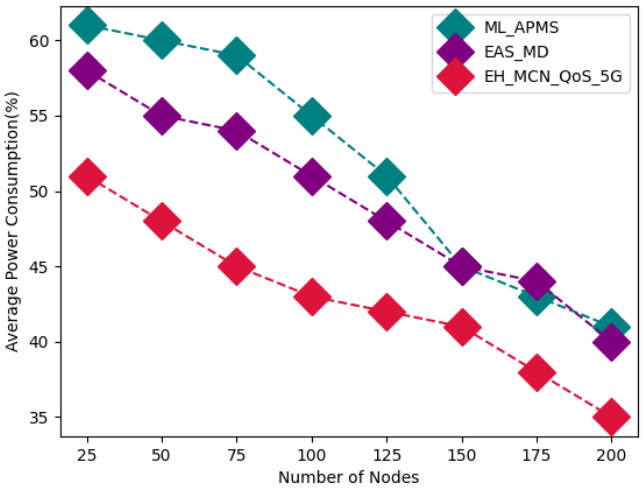


Figure-7 Comparative analysis of Packet delivery ratio

Above table-5 shows comparative analysis of PDR between proposed and existing technique. here the analysis has been carried out based on number of nodes for existing technique ML_APMS and EAS_MD. PDR can be calculated by dividing the total number of data packets that have reached their destinations by the total number of packets that have been delivered from sources. In other words, PDR is proportion of packets delivered from source to those received at the destination. Proposed technique attained PDR of 85%, whereas existing ML_APMS attained PDR of 75% and EAS_MD obtained PDR of 79% as shown in figure-7.

Table-6 Comparative analysis of Average power consumption

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	61	58	51
50	60	55	48
75	59	54	45
100	55	51	43
125	51	48	42
150	45	45	41
175	43	44	38
200	41	40	35



Figutre-8 Comparative analysis of Average power consumption

Table-6 shows comparative analysis of average power consumption between proposed and existing technique. To obtain the entire current The amount of time you will spend in each of the two states is multiplied by the two currents. where n is how many ranges occur each second. Therefore, the average current at 1Hz (n=1) will be 0.64mA. The value

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

for 10Hz (n=10) is 6.28mA. Power consumption is the entire amount of energy used by the network for data aggregation, transmission, and reception. The comparisons made between the various methods were based on how much energy was used by the cluster member and cluster head sensor nodes. The proposed technique attained average power consumption of 35%, existing ML_APMS attained average power consumption of 41% and EAS_MD obtained average power consumption of 40% for 200 nodes as shown in figure-8.

Table- 7 Comparative analysis of QoS

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	58	62	79
50	59	65	82
75	61	69	85
100	63	71	88
125	65	73	91
150	68	75	93
175	71	79	95
200	73	81	96

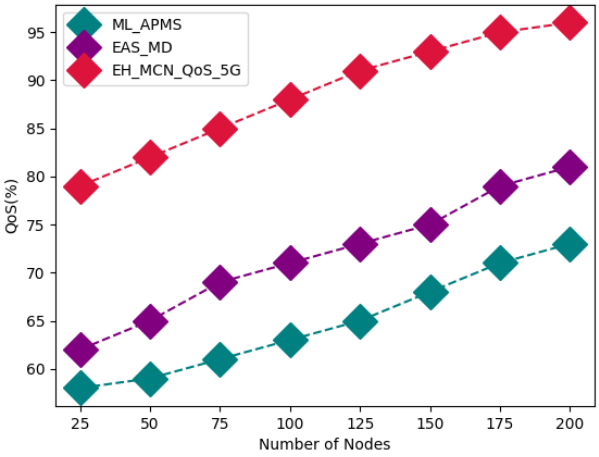


Figure-9 Comparative analysis of QoS

The above table-7 shows comparative analysis of QoS between proposed and existing technique. Design complexity of a WSN depends on certain application requirements such as number of nodes, power consumption, life span of sensors, data to be sensed as well as its timing, geography of where sensors are placed, environment, and context. Proposed technique attained QoS of 96%, whereas existing ML_APMS attained QoS of 73% and EAS_MD obtained QoS of 81% for 200 nodes as shown in figure-9.

Table-8 Comparative analysis of Latency

Number of nodes	ML_APMS	EAS_MD	EH_MCN_QoS_5G
25	65	61	55
50	63	58	52
75	64	55	51
100	61	53	45
125	55	51	44
150	54	45	32
175	51	44	31
200	50	42	28

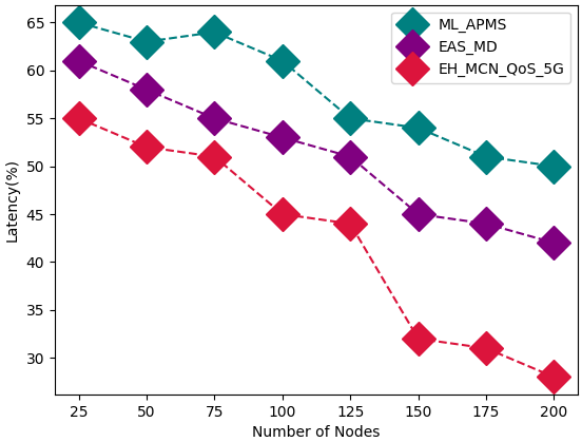


Figure-10 Comparative analysis of latency

The table-8 shows comparative analysis of latency between proposed and existing technique based on number of nodes. The term "latency" describes the interval of time between the occurrences of noticing something and the point at which the necessary action occurs. Low latency is a requirement for the majority of WSN deployments. For human rated systems, such as radiation level sensors [6] in nuclear power plants as well as temperature sensors in thermal power plants, a high latency requirement is essential. IoT systems and apps cannot tolerate any latency above a particular threshold. The proposed technique latency of 28%, whereas existing ML_APMS attained latency of 50% and EAS_MD obtained latency of 42% for 200 nodes as shown in figure-10.

5. Conclusion:

In proposed framework of this research aim to enhance network QoS and latency based on energy optimization and optimal routing. The energy optimization enhanced using discrete energy efficient offloading algorithm. The routing has been optimized using fuzzy logic cognitive Bellman-Ford routing algorithm. To identify the failing node and find

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
***Article Received**: 15 March 2022 **Revised**: 18 May 2022 **Accepted**: 25 May 2022 **Publication**: 30 June 2022*

an alternative node to deliver the seamless services, a unique weight-based approach has been presented. Energy harvesting (EH) devices are used to reduce the amount of energy and computational resources used in the MC system while still ensuring that IoT devices receive the quality of service they demand. We transform this stochastic optimization problem into a deterministic optimization problem and solve it. The experimental analysis has been carried out in terms of where proposed technique attained throughput of 96%, QoS of 96%, latency of 28%, energy consumption of 51%, end-end delay of 41%, average power consumption of 41% and PDR of 85%. To enhance EE performance in D2D communications, we will concentrate on how to jointly optimise power splitting ratio, power control, and partner selection in next works.

Reference:

[1]. Xuefei, E., Ma, Z., & Yu, K. (2020). Energy-efficient computation offloading and resource allocation in SWIPT-based MEC Networks. *IEEE Access*.

[2]. Wu, H., Han, X., Zhu, H., Chen, C., & Yang, B. (2022). An Efficient Opportunistic Routing Protocol with Low Latency for Farm Wireless Sensor Networks. *Electronics*, *11*(13), 1936.

[3]. Zhao, F., Chen, Y., Zhang, Y., Liu, Z., & Chen, X. (2021). Dynamic offloading and resource scheduling for mobile-edge computing with energy harvesting devices. *IEEE Transactions on Network and Service Management*, *18*(2), 2154-2165.

[4]. Xia, S., Yao, Z., Li, Y., & Mao, S. (2021). Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT. *IEEE Transactions on Wireless Communications*, *20*(10), 6743-6757.

[5]. Ben Ammar, M., Ben Dhaou, I., El Houssaini, D., Sahnoun, S., Fakhfakh, A., & Kanoun, O. (2022). Requirements for Energy-Harvesting-Driven Edge Devices Using Task-Offloading Approaches. *Electronics*, *11*(3), 383.

[6]. Guo, M., Li, Q., Peng, Z., Liu, X., & Cui, D. (2022). Energy harvesting computation offloading game towards minimizing delay for mobile edge computing. *Computer Networks*, *204*, 108678.

[7]. Hu, H., Wang, Q., Hu, R. Q., & Zhu, H. (2021). Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting. *IEEE Internet of Things Journal*, *8*(24), 17541-17556.

[8]. Rahmani, A. M., Mohammadi, M., Mohammed, A. H., Karim, S. H. T., Majeed, M. K., Masdari, M., & Hosseinzadeh, M. (2021). Towards data and computation offloading in mobile cloud computing: taxonomy, overview, and future directions. *Wireless Personal Communications*, *119*(1), 147-185.

[9]. Famitafreshi, G., Afaqui, M. S., & Melià-Seguí, J. (2022). Enabling Energy Harvesting-Based Wi-Fi System for an e-Health Application: A MAC Layer Perspective. *Sensors*, *22*(10), 3831.

[10]. Song, J., Song, Q., Wang, Y., & Lin, P. (2021). Energy–Delay Tradeoff in Adaptive Cooperative Caching for Energy-Harvesting Ultradense Networks. *IEEE Transactions on Computational Social Systems*, *9*(1), 218-229.

[11]. Mahmood, A., Ahmed, A., Naeem, M., Amirzada, M. R., & Al-Dweik, A. (2022). Weighted utility aware computational overhead minimization of wireless power mobile edge cloud. *Computer Communications*, *190*, 178-189.

[12]. Bi, H., Shang, W. L., Chen, Y., & Wang, K. (2022). Joint Optimization for Pedestrian, Information and Energy Flows in Emergency Response Systems With Energy Harvesting and Energy Sharing. *IEEE Transactions on Intelligent Transportation Systems*.

[13]. Rani, G. E., Sukumar, G. A., Chandra, T. U., Reddy, K. A., & Sakthimohan, M. (2021, August). Load Allocation as Quality and secured in Mobile Cloud Networking Location. In *Journal of Physics: Conference Series* (Vol. 1979, No. 1, p. 012045). IOP Publishing.

[14]. Abbasi, M., Mohammadi-Pasand, E., & Khosravi, M. R. (2021). Intelligent workload allocation in IoT–Fog–cloud architecture towards mobile edge computing. *Computer Communications*, *169*, 71-80.

[15]. Maray, M., & Shuja, J. (2022). Computation Offloading in Mobile Cloud Computing and Mobile Edge Computing: Survey, Taxonomy, and Open Issues. *Mobile Information Systems*, *2022*.

[16]. Hamdi, M., Hamed, A. B., Yuan, D., & Zaied, M. (2021). Energy-Efficient Joint Task Assignment and Power Control in Energy-Harvesting D2D Offloading Communications. *IEEE Internet of Things Journal*, *9*(8), 6018-6031.

[17]. Xie, Z., Poovendran, P., & Premalatha, R. (2021). Retention based energy harvesting technique for efficient internet of things aided edge devices. *Sustainable Energy Technologies and Assessments*, *47*, 101424.

[18]. Shukla, A. K., Upadhyay, P. K., Srivastava, A., & Moualeu, J. M. (2021). Enabling Co-Existence of Cognitive Sensor Nodes with Energy Harvesting in Body Area Networks. *IEEE Sensors Journal*, *21*(9), 11213-11223.

[19]. Li, S., Zhang, N., Jiang, R., Zhou, Z., Zheng, F., & Yang, G. (2022). Joint task offloading and resource allocation in mobile edge computing with energy harvesting. *Journal of Cloud Computing*, *11*(1), 1-14.

[20]. Kaliappan, V. K., Gnanamurthy, S., Kumar, C. S., Thangaraj, R., & Mohanasundaram, K. (2021). Reduced power consumption by resource scheduling in mobile

International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)
ISSN: 2454-4248 Volume: 8 Issue: 2
DOI: https://doi.org/10.17762/ijfrcsce.v8i2.2099
Article Received: 15 March 2022 Revised: 18 May 2022 Accepted: 25 May 2022 Publication: 30 June 2022

cloud using optimized neural network. *Materials Today: Proceedings*, *46*, 6453-6458.