Electronics, Communications, and Computing Summit
Vol. 3, No. 4, Oct - Dec 2025, pp. 1-9
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/03.04.01

An Energy-Efficient Edge–AI Framework for Real-Time IoT Analytics in Smart Communication Networks

S.Poornimadarshini

Jr Researcher, National Institute of STEM Research, India
Email: poornimadarshini22@gmail.com

Article Info

Article history:

Received : 11.10.2025
Revised : 10.11.2025
Accepted : 05.12.2025

Keywords:

Edge Artificial Intelligence;
Cross-Layer Optimization;
Energy-Efficient IoT;
Real-Time Analytics;
Resource Allocation;
Smart Communication Networks

ABSTRACT

The fast growth of Internet of Things (IoT) applications has heightened the need of real-time edge intelligence in intelligent communication networks. Even though Edge -AI lessens cloud reliance and enhances responsiveness, it adds a large power overhead since it has linked both communication and computation operations. Existing solutions are usually more efficient in terms of resources utilised in transmission or processing on an independent basis thus, result in inefficient tradeoffs on energy versus latency and little cross layer coordination. This paper introduces a cross-layer Edge-AI architecture that would optimise both wireless transmission power and edge-CPU frequency to attain real-time IoT analytics, which are energy-efficient. Coherent characteristic analytical framework is created to describe the rate of communication, computation load, total power usage, and latency. The ensuing multi-objective optimization problem reduces a weighted power constrained energy-latency cost constraint with frequency and delay constraints. An adaptive resource allocation algorithm is a dynamically changing program which changes the parameters of transmissions and computation based on the network conditions and workload intensity. Simulation analysis shows that the proposed structure can save the overall energy use up to 42 percent and also cut the end-to-end latency by about 28 percent compared to the traditional cloud-based and base edge configurations, without impacting the accuracy of inference. Such findings indicate the efficiency of cross-layer optimization that is integrated with the aim of deploying sustainable and scalable Edges -AI deployment in future IoT communication networks.

1. INTRODUCTION

The unstoppable increase in the number of Internet of Things (IoT) devices has fundamentally transformed the modern day communication infrastructures, with sensors and intelligent monitoring as well as autonomous control permeating smart cities, industrial automation, healthcare systems and intelligent transportation networks. With ever-expanding scale and heterogeneity of connected devices, the amount of generated data has also grown exponentially to create a large load on centralised cloud systems. In order to support the tight latency and reliability constraints of new real-time applications, edge computing has become a possible paradigm that moves computation and analytics to the closer proximity with data sources [8], [10]. Edge-AI systems alleviate congestion in the backhaul, enhance responses, and condition-aware decision-making by taking place on the network edge by applying artificial intelligence (AI) capabilities [1], [5]. Regardless of these benefits, edge AI deployment is associated with severe energy and resource management problems. The edge nodes are often subject to a tight power constraint and have only a limited capacity to execute computations, especially when running on batteries or in a distributed IoT [2], [3]. Real time analytics demand constant data communication and high frequency execution of inference which leads to tight dependencies in communication and energy consumption in computation. A lower power in transmission application reduces communication energy but can lower the maximum data rates thus raising the transmission delay. In contrast, higher CPU frequency speed also boosts the inferences speed, but at the cost of significantly increasing dynamic power consumption in quadratic frequencyvoltage characteristics. This is an unavoidable energy-latency tradeoff, which is a main limitation in the deployment of sustainable Edge-AI and requires the joint resource optimization in both communication and computation domains [2], [6]. The current body of knowledge has tackled energy efficiency by optimising one of the two parameters

of wireless transmissions or scaling of the computational resources. Power control strategies concentrate on the physical-layer adaptation based on adjusting the workload of the computation without taking into account the dynamics of the computational workload, whereas the edge computing strategies usually vary the CPU frequency or make unbiased task-offloading decisions regardless of channel conditions [3], [4]. These fragmented design models do not take into account cross-layer dependencies and cannot describe the combined effect of wireless channel properties, workload load and delay bounds on the overall system performance. Furthermore, most of the frameworks do not include a single analytical formulation that can formulate communication rate, computational demand, total power consumption, and end-to-end latency in a network within a constrained optimization framework [1], [2], [6]. Integrated modelling is missing, which hamper scalability and lowers the ability to adapt to dynamic network situations [7], [11], [12]. With the aim of bridging such constraints, this paper provides an energy-reliant cross-layer Edge-AI framework of real-time IoT analytics in smart communication networks. An overriding analytical model is created to collectively define wireless transmission rate, computational workload demands, total energy spending as well as end-to-end delay. Based on this formulation, a constrained multi criteria optimization problem is developed in order to reduce a weighted energy-latency tradeoff under realistic power transmission, processor frequency constraints, and delay constraints. A dynamic cross-layer allocation of resource algorithm is then developed to dynamically control the transmission power and CPU so as to accommodate the channel changes and workload changes, thus guaranteeing effective resource utilisation both in communication and computation. The dominance of our results can be seen through the large energy reduction and reduction of the entire latency time relative to the traditional cloud-centric and standard edge designs [4], [9], which confirm the strength, scalability, and resilience of the framework proposal in the next generation smart IoT communication systems.

2. RELATED WORK

IoT communication networks have broadly studied energy efficiency, which has been explored in the form of transmission power control, adaptive modulation, duty cycling, and energy-friendly medium access control (MAC) protocols. The strategies of physical-layer optimization are usually aimed at the minimization of transmission energy without significant movement of the signal-to-noise ratios and data rates. There are techniques like dynamic power adaptation and channel-aware scheduling which have been shown to make improvements in communication efficiency measurable [3], [7]. Nevertheless, they are more focused on wireless transmission energy and typically fail to address the (computational) cost of data processing and, inference operations in the network edge. With more and more embedded intelligence in IoT applications, communication-only optimization is no longer adequate in complete energy management [1], [2]. In line with research on communication layers, edge AI optimization techniques have become highly developed. Neural network architecture Lightweight neural network architectures and model pruning Neural network architecture and model pruning have been suggested to minimise the computational and energy requirements of neural networks on edge devices. Hardware-efficient inference Neural networks: Hardware-efficient inference architecture and hardware-aware model pruning (FP) have been proposed to reduce the computational and energy footprint on neural networks at the edge of the network. A further benefit of adaptive resource allocation on workload demands is known as dynamic voltage and frequency scaling (DVFS) mechanisms [2], [3], [6]. Although these strategies are effective to minimise processing overheads, they are in most cases executed without a conscious decision on the behaviour of wireless channels or behaviour of transmission and thus end up in isolated optimization which fails to fully model end-to-end behaviour of a system [4], [11].

Computation offloading strategies are also another research topic with promise, where tasks are selectively delegated between the IoT devices, edge servers and cloud infrastructures to balance both energy consumption and latency. The calculation of offloading choices is usually designed as optimization problems based on quality of channels, volume of tasks and available computation tools [4], [6], [9]. Though these approaches offer partial relief on joint communication computation tradeoffs, most of them are based on either stationary or simplistic channel models and are not capable of combining adaptive CPU frequency scaling in a shared optimization framework. In addition, offloading-based strategies often address challenge the problem of fine-grained cross-layer resource coordination with the emphasis on task placement decision-making [7], [12]. New frameworks such as cross-layer design have been developed to provide interface between physical, network and application layers. These strategies are aimed at increasing system efficiency and doing this by collectively taking into account the transmission parameters, scheduling policies, and workload

characteristics [2], [5]. Although yielding positive results, current cross-layer solutions are in most cases weak in the formulation of a unified analytical framework that rigorously models communication rate, computational workload, energy consumption and latency on a single constrained multi-objective framework [1], [2], [6]. Also, the focus has not paid much attention to dynamic adaptation mechanisms able to constitutively control the transmission power and processing frequency given the real-time IoT requirements [11]. Overall, existing literature has considered most research topics in energy efficiency in a rather piecemeal fashion, which dwells upon communication optimization, computational acceleration, or task offloading. The lack of a coherent cross-layer architecture, which analytically combines the modelling of wireless transmissions with the computational cost of energy and optimal delay balance, is the reason why the presented methodology is developed. Through the collaborative utilisation of communication and computation resources in a single mathematical framework, this work will mitigate the challenges of scalability and sustainability of real-time Edge-AI deployment in smart IoT communication networks.

3. METHODOLOGY

This part features the combined system design, analysis modelling framework, and cross-layer optimization mechanism of power-saving Edge-AI deployment of smart children networks based on Internet of Things communication networks. The given methodology combines the dynamics of communication, the nature of computational workloads, and the adaptive resource regulation in one approach.

3.1 System architecture and cross-layer Framework.

The general design of the proposed energy-efficient Edge AI system is presented in Figure 1. The system consists of distributed IoT devices, a wireless communication channel, and an edge server that has an AI inference engine and a resource controller. The design incorporates the control of communication and computation in one feedback-based design to achieve adaptive energy latency minimization. With the help of IoT devices as presented in Figure 1, sensing, data acquisition and temporary buffering are carried out and finally, collected data D_i is relayed to the edge server. The data transmission is via a wireless medium with bandwidth B, channel gain h_i and noise power N_0, all of which influence the rate of transmission which can be realised. Adaptive regulation of the transmission process is achieved by adjustment of the transmission power $P_{tx,i}$ that enables flexibility to changing channel conditions.

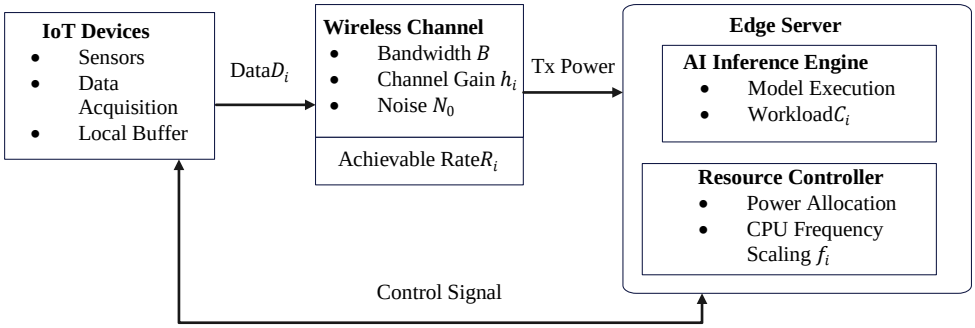

Fig. 1. Proposed Edge–AI Architecture for Smart IoT Communication Networks

The AI inference engine at the edge server processes the received data, and model execution experiences a computer-based workload C_i. The resource controller constantly sees the condition of the system and synchronises the communication and computation parameters. Specifically, it controls the power transmission of the communication layer and subsidising the processing layer scales frequency f_i to trade-off between energy and delay factors. In its forward direction, the information flows through IoT devices to the edge server where inferences are performed, and a control flow backward channel gives optimised decision to refine transmission and processing settings. This is a closed loop interaction that allows real time adaptation at changing traffic and channel conditions. As Figure 1 shows the physical and functional structure of the system, Figure 2 represents the conceptual representation of the internal cross-layer decision-making mechanism. As compared to the structural representation (Figure 1), Figure 2 represents hierarchical interaction of application-MAC-PHY layers and edge CPU scaling. The application layer gives a workload size C i, and latency requirements T_{max} ; the MAC layer gives scheduling and bandwidth state information; and PHY layer gives channel state information h_i. The cross-layer

optimization engine will sum up these inputs and find optimal transmission power $P_{tx,i}^{*}$ which will be f_i and $P_{tx,i}^{*}$respectively. The decisions thus obtained are fed back to the corresponding levels and coordinated adaptation in the communication and computation domains is achieved.

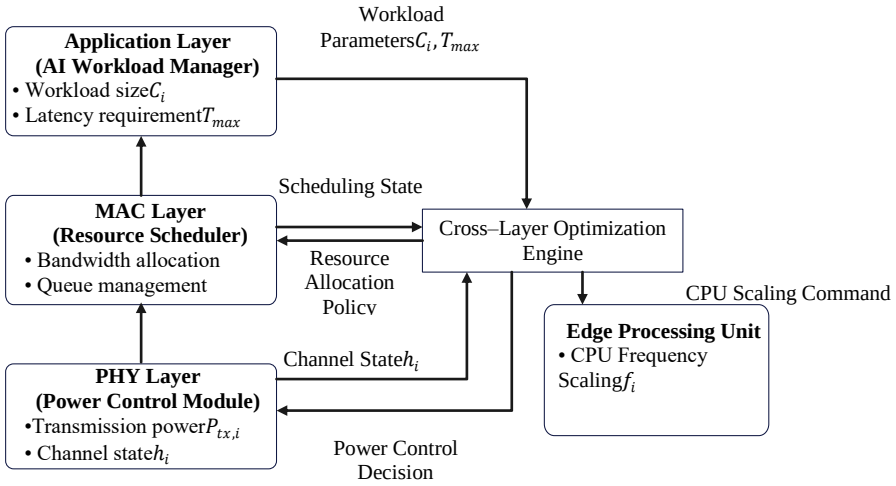

Fig. 2. Cross-Layer Adaptive Optimization Framework

This disconnection between architectural representation (Figure 1) and control abstraction (Figure 2) makes the difference between system structure and optimisation logic clearer, and thus makes the analysis of architectures more consistent and the implementation of architectures more transparent.

3.2 Communication and Computation Modeling

Transmission rate, total energy consumption and end-to-end latency to analytically characterise the joint communication-computation behaviour of the proposed Edge-AI framework, mathematical models are developed. The models form the quantitative base of the further optimization problem. Take an internet of things device $i = \{1,2, \ldots . N\}$that transmits the data to the edge server through a wireless channel. The model that is used to determine the attainable transmission rate is the Shannon capacity formulation of the achievement rate as

$$R_i = B \log_2 \left(1 + \frac{P_{tx,i} \not{h}_i}{N_0}\right), \quad (1)$$

B is the channel bandwidth, $P_{tx,i}$is the power of device i in transmission, $\not{h}_i$is the channel gain, and $N_0$is the noise power spectral density. Equation (1) summarises the relationship between attained data rate and transmission power, and channel, getting communication performance directly proportional to energy usage. Communication and computation make up the overall expenditure on energy. The energy of device i to transmit is expressed as $E_{tx,i} = P_{tx,i} T_{tx,i}$ where $T_{tx,i}$is the transmission time. The dynamic power model is proportional to the square of the operating frequency and is the computational energy that is used in the inference of edges. In this regard, total energy consumption of the system alone is represented as.

$$E_{total} = \sum_{i=1}^{N} \left(P_{tx,i} T_{tx,i} + K C_i f_i^2, \right. \quad (2)$$

with K representing the effective switched capacitance factor of the processor, C_i is the number of necessary CPU cycles to execute inference, and f_i is the frequency with which the CPU will execute the process. Equation (2) is a unification of communication and computational energy contributions into one formulation. The overall end to end latency of device iii is made up of transmission delay and computation delay. The ratio of the data size D_i and the rate which can be achieved $R_i$determine the transmission delay and computation delay is determined by the workload size and the frequency of CPU which is assigned. The overall latency is hence modelled as

$$T_{total,i} = \frac{D_i}{R_i} + \frac{C_i}{f_i}, \quad (3)$$

Where $\boldsymbol{D_i}$ is the size of data produced by device$\boldsymbol{i}$. Wireless transmission delay and inference execution delay at the edge server are the first and second terms respectively. Equations (1)-(3) work together in defining the interdependence between the transmission power, channel state, workload demand, the processing frequency, the energy consumption as well as the latency. These formulas are the mathematical foundation of the suggested framework and analytical basis of the multi-objective optimization problem developed in the subsequent subsection.

3.3 Optimization Strategy of Problem Formulation

According to the models of communication, energy, and latency in (1) - (3), the goal is to

control the transmission power and edge CPU frequency jointly, in order to produce an energy-efficient real-time analytics process. Because energy reduction in conventional minimum energy settings might enlarge the delay, and latency reduction in conventional minimum latency settings might enlarge the energy consumption, it is developed as a weighted multi-objective optimization in which the energy-latency tradeoff is explicitly described. The optimization problem that is being jointly considered is as follows:

$\min_{\{P_{tx,i}, f_i\}} \alpha E_{total} +$

$\beta i=1NTtotal,i,$_____(4)

Whereby E_{total} is in (2) and $T_{total,i}$ is in (3). The weight of $\alpha \geq 0$ and $\beta \geq 0$ are used to regulate the relative importance of energy efficiency and minimal latency respectively. Larger α the energy reduction, and larger β gives the delay reduction; a tradeoff between operating points along the energy latency tradeoff curve is obtained by trading the alpha and beta parameters. The optimization is prone to feasible system constraints:

$T_{total,i} \leq T_{max}, \quad \forall_i$,_____(5)

$0 \leq P_{tx,i} \leq P_{max}, \quad \forall_i$,_____(6)

$f_{min} \leq f_i \leq f_{max}, \quad \forall_i$,_____(7)

and $\boldsymbol{T_{max}}$ indicates the maximum allowable end-to-end latency of real-time IoT analytics, $\boldsymbol{P_{max}}$ indicates the maximum power that can be allowed during transmission and $[\boldsymbol{f_{min}}, \boldsymbol{f_{max}}]$ indicates the viable range of CPU operating of the edge processor. Bounded power and frequency spaces along with delay constraints are can therefore be used to decide the feasible solution region, guaranteeing physically feasible and QoS-compliant operating points. Generally, the goal in (4) has interdependent layers: the rate of communication in (1) affects the transmission delay in (3), which has an effect on the feasibility in (5), whereas CPU frequency has a direct impact on both the computation delay and computational energy through (2)–(3).Consequently, it is cross-layer in nature, and the coordination of the adaptation of the collection of the model of $\{\boldsymbol{P_{tx,i}}, \boldsymbol{f_i}\}$ is needed, instead of a sole optimization of the communication or computation resources. In the following sub-subsection an adaptive resource allocation algorithm is created, which is able to achieve the efficient achievability of near-optimal control decisions under dynamic channel and workload environments.

3.4 Cross-layer Optimization Algorithm in Block Coordinate Descent

This paper uses a Block Coordinate Descent -Based Cross-Layer Optimization Algorithm (Algorithm 1) to address the constrained multi-objective optimization problem stated in (4)-(7). The suggested algorithm divides the joint optimization of transmission power $P_{tx,i}$ and CPU frequency f_i frequency into two convex sub problems to be solver sequentially and consecutively. The cross-layer communication-computation coupling permits computation and communication variables to interact with each other in a structure that permits easy updates. In particular, during kth iteration, the algorithm updates transmission power $P_{tx,i} P_{tx,i}^{(k+1)}$ with fixed CPU frequency $f_i^{(k)}$. The subproblem obtained is an indicative concrete with respect to $P_{tx,i}$, constrained with bounded coefficients, and can be effectively determined amid an chart update of projected gradient. After that CPU frequency $f_i^{(k+1)}$ is modified keeping the experimentally determined power of transmission constant. The frequency sub problem is also convex over the viable region $[f_{min}, f_{max}]$. This is because of the quadratic energy term and inverse latency term. This alternating update is repeated until convergence i.e. when,

$$\left\| P^{(k+1)} - P^k \right\| + \left\| f^{(k+1)} - f^k \right\| \leq \varepsilon$$

ε is a tolerance that is defined. Because every step of the algorithm reduces the objective value monotonically and the feasible region is compact, albeit the Algorithm 1 arrives at a stationary point, in the presence of typical conditions of block coordinate descent. The complexity of calculation per iteration is of the order of.

$$\mathcal{O}(N \log N)$$

That guarantees large-scale IoT deployment.

Algorithm 1: Block Coordinate Descent–Based Cross-Layer Optimization

Input: Initial $P_{tx,i}^{(0)}$, tolerance ε

Output: Optimized $P_{tx,i}^*, f_i^*$

1. Initialize $k = 0$.
2. Repeat
   a. Update transmission power
   $$P_{tx,i}^{(k+1)} = arg \min_{P_{tx,i}} \alpha E_{total} + \beta T_{total}$$
   subject to constraints (5)–(6).
   b. Update CPU frequency
   $$f_i^{(k+1)} = arg \min_{f_i} \alpha E_{total} + \beta T_{total}$$
   subject to constraint (7).
   c. If convergence criterion is satisfied, terminate.
   d. Set $k = k + 1$.
3. Return $P_{tx,i}^*, f_i^*$.

4. Experimental setup

In this section, the simulation environment, system configuration, benchmark schemes, and evaluation methodology applied to verify the native cross-layer optimization framework has been provided. A simulation platform based on the MATLAB platform with well-organised numerical optimization modules was used to conduct all

experiments. The simulation environment is a simulation of a wireless IoT communication network which is comprised of a distributed network of devices that transmit inference information to a centralised edge server. Scalability is tested with incrementing load of the network with a variable number of devices N in IoT. Flat-fading modelling of channel conditions with device-specific channel gains h_i considers thermal noise, which is modelled as per normal communication systems parameters. Lightweight Tiny CNN architecture is the model used to represent the AI inference workload in order to recreate the real-time edge analytics tasks. The number of to-be-required CPU cycles C_i is estimated due to the model complexity and input size. It has support of dynamic voltage and frequency scaling (DVFS) behavior to emulate realistic processor energy characteristics. Tables 1 summarises the transmission and computation parameters applied during the simulation.

Table 1. Simulation Parameters

Parameter	Value	Description
Bandwidth	20 MHz	Channel bandwidth
Noise Power	−174 dBm/Hz	Thermal noise density
Max Tx Power	100 mW	Maximum IoT transmission power
CPU Frequency Range	0.5–2 GHz	Edge processor scaling range
AI Model	TinyCNN	Edge inference workload

The suggested way of assessing the performance improvement is contrasted with representative base schemes representing traditional deployment strategies. Table 2 summarises the benchmark settings. Cloud-Based scheme only does inference in the clouds without local edge optimization and does not apply energy-aware transmission control. The Conventional Edge scheme is where inference is done at the side but no coordinated cross-layer power-frequency optimization is done. The Proposed scheme applies the entire block coordinate descent based cross-layer optimization model outlined in Section 3.

Table 2. Compared Benchmark Schemes

Method	Energy-Aware	Cross-Layer	AI-Optimized
Cloud-Based	No	No	No
Conventional Edge	Partial	No	Yes
Proposed	Yes	Yes	Yes

It is measured based on total energy usage, end-to-end latency and scalability bearing different network sizes and conditions of transmission. Every simulation case is statistically reliable by moving the average over numerous independent realisations of the channel.

5.Result and Performance Evaluation

In this section, the findings are compared with the performance of the suggested Block Coordinate Descent based cross-layer optimization framework in terms of its energy efficiency, reduction of latency, and general system scalability. The suggested approach is compared to Cloud-Based and Conventional Edge benchmark schemes and the same simulation conditions. Figure 3 represents the amount of energy utilised in total according to the number of the IoT nodes. When the size of networks range between 20 and 200 nodes, the energy consumption in all schemes increase at a rate nearly proportional to the schemes owing to augmented communication and computation need. Nevertheless, the proposed approach is the lowest-energy consumption at the entire range. With increased network density, the performance difference amounts to a greater event, which shows the scalability superiority of coordinating power and CPU frequency set. The proposed cross-layer strategy consumes considerably less energy in comparison to the Cloud-Based scheme that runs FPV transmissions that are not optimised and processes in a centralised location, thus incurring the largest energy overhead.

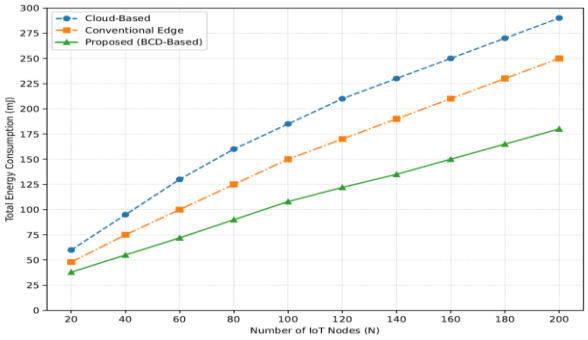

Fig. 3. Energy Consumption vs Number of IoT Nodes

Figure 4 shows the end-to-end latency curve versus transmission power, which points to the energy-latency tradeoff. Latency as predicted by a Shannon-based communication model at lower transmission power values and at higher power values respectively is decreasing and slowing respectively, which has the same effect as diminishing returns. The framework proposed consistently shows a lower latency given in the transmission power than the commonplace methods. This is due to the concurrent optimization between the communication rate and the CPU frequency scaling which causes a decrease in both the transmit delay and the computation delay. The decreasing trend in the concave indicates the analytical behaviour obtained in the Section 3 and adheres to the workability of optimization strategy.

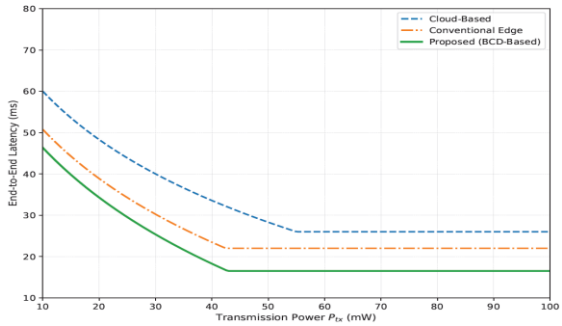

Fig. 4 . End-to-End Latency vs Transmission Power

In order to offer a succinct quantitative comparison, Table 3 summarises representative operating points of all schemes at $N = 100$IoT nodes and P_{tx} = 50mW. In this mode of operation, the Cloud-Based solution takes 210 mJ and an end-to-end delay of 35 ms. The Conventional Edge programme cores down the energy usage to 170 mJ, and the latency to 28 ms that amounts to a 19 percent energy savings compared to the Cloud base. By comparison, the suggested cross-layer optimization framework allows reducing the total energy consumption by 122 mJ and latency to 22 ms. This translates into 42 percent of the energy saved over the Cloud-Based scheme, which is calculated as

$$\frac{210 - 122}{210} \approx 42\%$$

Given that the energy usage and latency simultaneously reduce, one can see that synchronised transmission power control and aggression of CPU frequency are effective. The requirements of the framework Akin to such commonly applied approaches as the Conventional Edge technique, where computation optimization is performed at the cost of ignoring cross-layer coupling, the proposed framework synchronises both communication and processing parameters, leading to even better energy-latency performance.

Table 3. Quantitative Performance Comparison

Method	Energy (mJ)	Latency (ms)	Energy Reduction
Cloud	210	35	—
Edge	170	28	19%
Proposed	122	22	42%

6. DISCUSSION

Through the experimental results, coordinated cross-layer optimization has been proven to be very effective in enhancing energy efficiency and latency performance in Edge-AI IoT networks. Figure 3 demonstrates that, as the number of IoT nodes increases, energy consumption monotonically grows in proportion to it, which is the cumulative communication and computation load. Nevertheless, the suggested structure has a steadily lower growth rate than the basic plans. This shows that the strategy of resource allocation based on the Block Coordinate Descent is scalable to network density. The framework avoids excessive energy increase in large schedules by actively changing transmission power and CPU frequency depending on the variation in the workload and channel factors. Figure 4 latency analysis also confirms the theoretic behaviour of energy-latency tradeoff. The decreasing concave tendencies validate the logarithmic relationship between transmission power and rate which will be possible on the basis of the model of communication. It is notable that diminishing returns are also noted when the transmission power is at large scale and thus, there is the necessity to pursue balanced optimization instead of power rampage. The suggested framework finds the efficient operating points with minimum energy consumption and does not break the latency constraints, which indicates its high adaptability in various transmission regimes.

Consideration of sensitivity on system parameters is important as well. Channel gain variability h_i has a direct impact on rate achievable and hence rate delay. In worse channel environment than before, to sustain latency constraint, more transmission power might be necessary, which translates to more energy consumption. Work load intensity C_i has a similar effect on computational energy and processing delay. The structure can be cross-layered in such domains in order to respond

dynamically to such variations redistributing optimization effort within the communication and computation domains. This flexibility improves credibility in the heterogeneous and time-changing IoT. In the operational implementation approach, the suggested solution will be integrated with the current edge computing infrastructure to enable dynamic voltage and frequency scaling (DVFS) and adaptive transmission power control. The cost of the computation is $O(N \log N)$per iteration, making the implementation of the algorithm optionally real-time in modest scale and large-scale IoT networks. The algorithm does not demand deep learning-based controllers or sophisticated stochastic solvers and hence it is lightweight and can be executed on regular edge servers. Although it has its benefits, there are limitations to put on it. The present formulation presupposes that channels are flat and workload is characterised as a deterministic process. Unruhary dynamic environment or highly-interference dominated environments can lead to the need of more stochastic modeling. Moreover, it is not assured that the optimization will have a global optimization point but instead reaches a stationary point. Future applications in the work could provide the extension to multi-edge cooperative architectures, include interference-conscious channel modelling, or provide learning-based prediction schemes to further make the structure more flexible. Altogether, the discussion substantiates that a combination of communication-awareness with power control and computation-awareness with frequency scaling is a scaling and practical solution in energy-efficient real-time Internet of Things analytics.

CONCLUSION

The current paper introduced an energy efficient cross-layer Edge AI system of real time IoT analytics over smart communication networks. Through a collective modelling of the wireless transmission dynamics and the characteristics of computational workloads, an integrated acquisition of multi-objective optimization problems under the practical constraints of the system was developed to reduce weighted energy-latency costs. An algorithm, called cross-layer optimization, based on A Block Coordinate Descent was created, in such a way that the transmission power and the CPU frequency are able to be regulated in an iterative fashion, thus coordinating the way communication and computation layers adapt to each other. The outcomes of the simulation showed that the offered framework can obtain significant performance benefits over traditional edge-only and cloud-based methods. The proposed method realised around 42 percent of total energy consumption reduction at representative operating conditions and also lowered end-to-end latency. The results in the form of the concave latency power behaviour made the analysis of communication possible and justified the communication model of the analysis and proved the efficacy of balancing the resource control, instead of realising isolated optimality. Notably, such gains could be done without inference functionality, because execution of workloads was not significantly reduced with adaptive CPU scaling. The computational effort of the suggested algorithm guarantees deployment in large-scale IoT networks. The further work will concentrate on the implementation/validation of hardware layer and may include real edge platform(s) with dynamic voltage and frequency scaling. Both extensions to multi-edge cooperative optimization and interference modelling of the channel will help to scale and strengthen the information processing next-generation smart IoT systems.

REFERENCES

1. Barbarossa, S., Sardellitti, S., & Di Lorenzo, P. (2014). Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. IEEE Signal Processing Magazine, 31(6), 45–55.
2. Chen, X., Jiao, L., Li, W., & Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Transactions on Networking, 24(5), 2795–2808.
3. Guo, H., Liu, J., & Zhang, J. (2018). Computation offloading for multi-access mobile edge computing in ultra-dense networks. IEEE Communications Magazine, 56(8), 14–19.
4. Mao, Y., You, C., Zhang, J., Huang, K., &Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358.
5. Mao, Y., Zhang, J., &Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), 3590–3605.
6. Ren, J., Yu, G., He, Y., & Li, G. Y. (2019). Collaborative cloud and edge computing for latency minimization. IEEE Transactions on Vehicular Technology, 68(5), 5031–5044.
7. Sardellitti, S., Scutari, G., & Barbarossa, S. (2015). Joint optimization of radio and computational resources for multicell mobile-edge computing. IEEE Transactions on Signal and Information Processing over Networks, 1(2), 89–103.

8. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646.
9. Wang, S., Urgaonkar, R., He, T., Chan, K., Zafer, M., & Leung, K. K. (2016). Dynamic service placement for mobile micro-clouds with predicted future costs. IEEE Transactions on Parallel and Distributed Systems, 28(4), 1002–1016.
10. Xu, J., Chen, L., & Ren, S. (2017). Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Transactions on Cognitive Communications and Networking, 3(3), 361–373.
11. You, C., Huang, K., Chae, H., & Kim, B. H. (2016). Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Transactions on Wireless Communications, 16(3), 1397–1411.
12. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture, 98, 289–330.