International Journal of Engineering in Computer Science

E-ISSN: 2663-3590
P-ISSN: 2663-3582
IJECS 2022; 4(2): 59-66
Received: 10-09-2022
Accepted: 14-10-2022

Mithilesh Kumar Singh
Ph.D. Scholar, Department of Business Management, IAR, Gandhinagar, Gujarat, India

Rao Bhamidimarri
Professor, President, IAR, Gandhinagar, Gujarat, India

Dr. Meena Sharma
Professor and Head of Department, Department of Business Management, IAR, Gandhinagar, Gujarat, India

Corresponding Author:
Mithilesh Kumar Singh
Ph.D. Scholar, Department of Business Management, IAR, Gandhinagar, Gujarat, India

CoAP based smart industry automation using scheduling and emergency data transmission technologies

Mithilesh Kumar Singh, Rao Bhamidimarri and Dr. Meena Sharma

DOI: https://doi.org/10.33545/26633582.2022.v4.i2a.79

Abstract
The intelligent industrial is one of the Iota's most significant uses. IPv6 over time-slotted channel hopping is the new protocol for industrial telecommunications (CoAP). Scaling, speed, and sustainable energy are tough problems for smart businesses. Current studies currently provide scheduling tactics for CoAP-based businesses. These works, however, fall short in real-world settings when crises occur frequent. Therefore, the goal of this research is to improve the scale, latencies, and fuel efficiency of CoAP-based smart enterprises in order to increase their practical usefulness. The new CoAP design, which incorporates edge technologies, is composed of three layers: Industry, Edge, and Cloud. Edge computing achieves scalability and latency by working in the edge layer. Real-time scheduling of crucial data packets is made possible by fuzzy logic-based parental control, which cuts down on transmission delays. Consumers can access data from numerous industries on the cloud layer. To evaluate the performance of the CoAP networks, Networks Simulator 3.26 is used. Experiments show that the proposed work achieves its objectives in terms of energy consumption, transmission time, and economics.

Keywords: Smart industry, CoAP, dynamic scheduling, data transmission technologies, and edge computing

Introduction

The terms "Industry 4.0" and "industrial internet of things" have recently emerged as essential concepts in many modern businesses (IIoT) [1, 2]. To facilitate smart communication in the workplace, the IEEE 802.11.15.4-based CoAP network has been implemented. Industry is now exploring the concept of edge computing in an effort to decrease latency and boost energy efficiency in networks. Edge computing is a novel approach to edge computing that moves data processing to the edges of a network. Bandwidth can be reserved in real-time, on the fly (OTF), as a means of controlling the frequency with which broadcasts must be resent. However, this is the result of increasing network bandwidth utilisation by setting aside supplementary transmission slots for retransmissions [3]. To facilitate more precise management of the CoAP network's scheduling, a centralised scheduling approach was created. Centralized scheduling's biggest drawbacks are longer wait times and complicated mobility management. Grouping nodes into DODAGs with the same depth levels might decrease the time needed to complete the entire process. However, in dynamic networks, maintaining a constant depth across all DODAGs is inappropriate. An alternative method of scheduling in the CoAP network was proposed: PID control (an acronym for proportional, integral, and derivative). Emergency data management is a crucial part of industrial networks, but the PID technique can't handle it. A priority based fast forwarding scheduling function (PFFSF) algorithm was used in the priority queuing layout. The combination of the CoAP network and edge computing has been used in a wide variety of Internet of Things applications to improve latency and energy efficiency [4]. However, the network's performance is hampered in a few use cases because of inefficient scheduling and offloading methods. Processing data in the cloud causes an increase in latency for end users and network devices. Iterative search algorithms (ISAs) were created as a means of resolving the offloading problem in mobile edge computing. Using the ISA technique, you may find less-than-optimal answers to your problems. Unfortunately, the best offloading technique cannot be determined by ISA. An energy-conscious adaptive offloading method employs the Lyapunov optimization computation offloading. It is crucial to adapt this technology so that it can be utilized with large networks, as it is now only suitable for networks of a certain size.

Queuing theory was also used to the offloading phase of edge computing [5, 6, 7, 8]. Establishing and maintaining a timetable that meets the Quality of Service (QoS) requirements in the IIoT will be a formidable challenge. An original approach to scheduling IoT networks is presented in this paper. Fuzzy logic is used to ensure that data packets are delivered to the root in as few hops as feasible. After selecting a number of nodes equal to the network depth, the proposed technique significantly reduces the amount of time spent waiting for data [9, 10]. The rest of the paper may be broken down as follows: The background behind CoAP's development is discussed in Section II. In section III, we show the proposed method we recommend. The assessment of the proposed classification method is discussed in Section V below. The findings will be discussed in the final portion (section VI) of the study.

Related Work

CoAP for Industrial Applications

Limited Application Protocol, or CoAP, is a term used to refer to one of the application protocols for constrained devices. The protocols are used to communicate with the constraint devices through the internet. The identical constraint network is designed using it. This protocol is also utilised for end-to-end security, authentication, and other purposes. The primary function of CoAP is to exchange small packets between constrained devices. CoAP is comparable to the HTTP protocol, which supports the GET, POST, PUT, and DELETE requests. The CoAP protocol's structure. This protocol has a fixed header length of 4 bytes, which contains the options, token, and payload parameters. Using the servers, the CoAP client makes a request to the server in a response/request format.

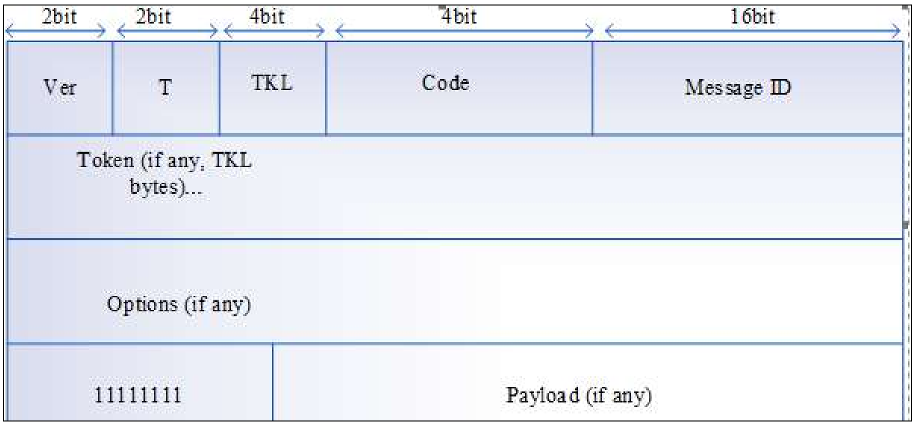

Fig 1: CoAP Architecture

Fig 1 shows the architecture for CoAP. T stands for the message type, and TKL for the variable token field length. Both dependable and non-reliable transmission modes are supported by the CoAP protocol. Four different message kinds are included in the CoAP protocol: confirmable (CON), non-confirmable (NON), acknowledge (ACK), and reset messages. A trustworthy channel of communication is used to transmit the CON message, after which the receiving devices transmit an ACK reply. A NON message receives no acknowledgement from the recipient. If the sender does not get an ACK message from the receiver after a specific amount of time, then the transmission is considered unreliable.

Background of this study

Deploying IIoT and industrial 4.0 in the edge reduces network energy usage by minimising latency for sensors and end users [13]. This study suggests an adaptive edge setup for IIoT to reduce latency and increase energy economy. In this work, we focus on optimizing edge nodes' capacity. Lyapunov-optimized and parallel adaptive edge configuration (AFC) is used in this It is recommended that edge-assisted IIoT networks implement the Gibbs sampling technique to boost performance. Mist computing is introduced in IIoT in [14] to decrease network latency and increase energy efficiency. The term "extreme edge Computing" (or mist computing) describes the next generation of edge computing. This research report argues that edge device processing of data in the IIoT, as opposed to cloud processing, results in better network performance. Therefore, the network's energy efficiency may be enhanced with the help of edge computing and mist computing. An IoT-agnostic framework is proposed in [15]. The suggested framework has two main functions: I finding and Collecting data from IoT-enabled devices, and (ii) planning the operation of these appliances in accordance with the information gathered. Data processing and analysis are also well suited to the Hadoop system. The CoAP standard is used in the network to facilitate data exchange between nodes. Since data processing is handled on the cloud layer, which is separated from the network layer, latency is high in this architecture. The offloading done by the RBQ-learning algorithm takes place within the edge layer, where latency is kept to a minimum. Energy efficiency and latency in edge computing are addressed by the authors of [16]. In this work, we define a weighting factor based on energy consumption and delay, which takes into account the remaining charge in the device's battery. Offloading and resource allocation is treated as a mixed integer nonlinear problem (MINLP) with the goal of reducing delay and increasing energy efficiency. Then, an iterative search technique is presented to solve the MINLP issue, and it is based on a combination of the interior penalty function and the difference of two convex functions/sets. The suggested iterative search strategy in this research can only produce sub-optimal answers and cannot locate an ideal solution for offloading. The RBQ-learning method chooses the best edge nodes for offloading. The authors of [17] propose an offloading technique for the cloud of things that takes into account energy consumption. The edge nodes are fueled by green energy to reduce the amount of brown energy they use. Computation offloading is done to decrease lag time in the edge layer. An approach based on

Lyapunov optimization is developed for compute offloading; it minimizes the total time and energy cost of the calculation. In a real world setting with a large number of nodes, this approach will not work. A large-scale network can efficiently aggregate data with the help of an involved DODAG structure. A computational offloading mechanism is proposed for edge computing in IoT applications by the authors [18]. The primary goal of this article is to reduce network latency and increase energy efficiency. Three queuing models are implemented for devices, edge, and cloud servers, all based on queuing theory. This study takes into account the M/M/1, M/M/C, and M/M/ queuing theories. Since the offloading nodes are chosen in a suboptimal fashion, this technique is inefficient. The RBQ-learning method prioritises the well-being of the network as a whole by selecting the best edge nodes. In this study, we present a decentralised scheduling method for CoAP networks. First, a DODAG graph is built, using the same pattern as the RPL protocol. The scheduling is done based on data shared between surrounding nodes, which helps to reduce the likelihood of collisions. In this case, scheduling is handled in a distributed fashion by each node in the DODAG graph. Each node's RPL ranking is used in the network's channel offset calculation. The DeTAS technique is a non-sequential procedure (i.e., the scheduling of a parent node is done independently of the scheduling of its child nodes). As a result, network latency will rise. In this case, scheduling is carried out in accordance with normal traffic levels rather than emergency traffic levels. As a result, there is a lot of downtime with emergency data. For highly fluid systems, this approach is inadequate. In this study, we schedule a scenario in which data will be delivered in both directions, called a "divergecast." The scheduling in this case is handled using CoAP networks, which were developed primarily for use in manufacturing. CONREQ, CON-ACK, Data, Data-ACK, and KEEP-ALIVE packets are sent between nearby nodes in the network as control traffic to coordinate scheduling. When neighbours in all directions exchange five control packets with one another, it uses more resources and takes up more bandwidth. This approach works well with preexisting networks and cannot accommodate the addition of new ones. Scheduling is performed efficiently by the suggested ECDAS algorithm without an increase in bandwidth usage or power consumption. The use of R2S-based DODAG architecture enables a flexible network environment. In this research, we introduce a priority-based scheduling technique that prioritises data according to the severity of the emergency. In the event of an emergency, the data can be classified by its flag value in the packet. In the event of an emergency, only the most reliable data transmission channel is chosen and planned. In this case, the optimum channel is chosen by calculating the smooth packet delivery ratio (PDR) using the weight value. An ideal channel is designated for urgent data transfer upon a successful PDR. Given that smooth PDR depends on the value of, using smooth PDR as the basis for optimal channel selection is ineffective. Obtaining the best results from this approach to scheduling requires fine-tuning a number of factors. In this work, we mix IoT and edge computing to reduce user latency. In order to distribute the workload fairly across the edge nodes, a new offloading approach is presented for use in Edge based IoT. Reinforcement learning is used to identify manufacturing faults, and the integrated edge-based IoT is utilised in the manufacturing area. In the event that offloading is chosen, this deep learning process is sent to the cloud. When data analysis is outsourced to the cloud, it adds a delay for the end user. If you want to do data analysis in the cloud, your edge node will have to send intermediate data to the cloud, which will increase network traffic and therefore your need for a lot of bandwidth. Here, we provide an optimum offloading strategy and an auto scaling approach for enhancing power effectiveness. In this setup, edge nodes (also known as edge nodes) communicate with the home station. Offloading data size and auto scaling strategy are established to distribute network activity evenly. To achieve this goal, we offer an algorithm that takes into account the condition the system is in after a choice has been made. However, this approach works best in networks with a single base station, such as a edge node. End-user latency rises when data is offloaded to the cloud [19].

Research Gaps

Decentralized traffic aware scheduling (DeTAS) was distributed scheduling scheme performed based on destination oriented directed acyclic graph (DODAG) [20]. But DeTAS method is a non-sequential process which limits the performance of data aggregation and increases scheduling delay. Furthermore, scheduling based on amount of traffic instead of emergency level of traffic increases delay for emergency data. A distributed divergecast scheduling algorithm was designed with five control packets [21]. Exchanging of five control packets for scheduling increase bandwidth consumption and energy consumption. In addition, both the methods are not suitable for dynamic networks and not able to support new nodes in the network. In priority based scheduling with best channel selection, smooth PDR was considered as selection criteria [22]. However, smooth PDR computation is fully depends upon weight value α which is not suitable for best channel selection. In this method, multiple parameters are needed to be optimally tuned in order to perform optimal scheduling. For minimizing latency, reinforcement learning based offloading strategy was proposed [23]. But in this method, each edge node must transmit intermediate data to cloud which increases network traffic results in high bandwidth consumption. Online learning algorithm for offloading and auto scaling was proposed in [24]. This method is only suitable for network with single base station. In addition, offloading data to cloud layer increases latency in both methods.

Proposed Work

Our suggested research study is on reducing delay and power consumption in industrial networks as a means of overcoming the aforementioned issues. We propose a new Tri-Layer Edge-Assisted CoAP structure in this study, which consists of the following levels of abstraction: There are three distinct layers: Industry Layer I Edge Layer (ii), and Cloud Layer (iii). The nodes, sensors, actuators, and other components of an information system make up the first layer. Then, the CoAP networking standards is modified to allow nodes to communicate with one another. Edge nodes are set up in the edge layer. The highest level is the cloud, which houses the public cloud. The following operations are carried out in each successive layer:

Fig 2 illustrates the proposed system architecture. In order to facilitate effective data analysis and collision-free sequencing, nodes are organised into DODAG structures at the network layer. The first step in building a DODAG is to choose a good starting point, or root node, using the roughset root technique known. The suggested approach uses residual energy, the number of nodes, and the distance to the smart gateway to determine the weighting factor. A sequential scheduled procedure is carried out in a created DODAG. A unique adaptable scheduling technique is developed, which takes into account both the volume of traffic and the priority of the information being scheduled. Optimum parent node choice is made in this case by a Fuzzy Logic algorithm taking into account distance, energy level, network quality, and anticipated transmissions (ETX). To guarantee the smooth delivery of urgent information, the Channel selection channel and parent are chosen. As a final step in the process of developing distributed DODAGs, the root node will transmit a 'Build' message to its child node. The Build message is sent down the DODAG to the very last node. The suggested ECDSA method is sequential, and it constructs schedules in a Child-to-Parent fashion, as was previously mentioned. The planning procedure is kicked off by the child node that received the Build message, which uses an FBL-based parental choice to find a suitable time to start building. Parental Preferences Determined by Fuzzy Logic Scheduling using the proposed method is carried out between the child node and the best parent node. Selecting the best parent node for data transfer increases link reliability. The Fastest Growing Graph (FBL) technique is used in the proposed algorithm to choose the best possible parent node. This new system, called the FBL algorithm, combines the best features of fuzzy logic with those of the Bayesian learning algorithm. Whereas Bayesian learning makes use of probabilities, fuzzy logic relies on IF-THEN rules to make judgments. When choosing a parent, the fused FBL method takes into account both rules and typical range. Probabilistic calculations are useful in fuzzy rescheduling.

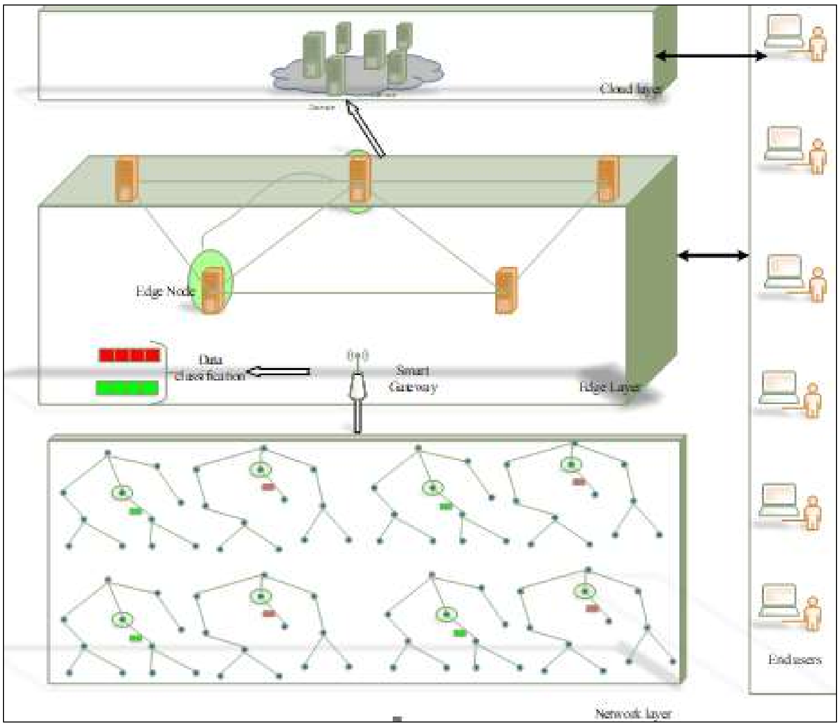

Fig 2: Proposed system architecture

The investigated characteristics are first subjected to the Bayesian learning rule, and then their values are transferred among [0, 1]. Then, the fuzzy logic is fed these data to choose the best parent node possible. Link quality (), and ETX are the variables in question (X). As an example of the use of Bayesian learning, consider the following:

$$\mathcal{P}_1 = \mathcal{P}(N_i(\mathcal{E})|PN) = \frac{\mathcal{P}(PN|N_i(\mathcal{E}))\mathcal{P}(N_i(\mathcal{E}))}{\mathcal{P}(PN)} \quad (1)$$

$$\mathcal{P}_2 = \mathcal{P}(N_i(\wp)|PN) = \frac{\mathcal{P}(PN|N_i(\wp))\mathcal{P}(N_i(\wp))}{\mathcal{P}(PN)} \quad (2)$$

$$\mathcal{P}_3 = \mathcal{P}(N_i(\mathbb{X})|PN) = \frac{\mathcal{P}(PN|N_i(\mathbb{X}))\mathcal{P}(N_i(\mathbb{X}))}{\mathcal{P}(PN)} \quad (3)$$

Here, eq. (1) calculates, from, the likelihood that a given node, N i, will become a parent node (PN). The frequency is also calculated in eq. (2) and eq. (3), but this time using and X. The classifier for each potential node is determined using these formulae. Figure 2 depicts the assumed triangular shape of the classifier.

The criteria are then used to determine which parent node will be the most suitable for passing on the data. Following are the IF-THEN conditions that govern the FBL method:

Rule 1

IF (P_1) & (P_2) & (P_3) is (>0.75) THEN (O) is (PPN)

Description- All triangular membership functions must be larger than 0.75 for a node to be called a partial parent node (PPN), which means the node has a moderate likelihood of becoming a parents node.

Rule 2

IF (P_1) & (P_2) is (>0.75) & (P_3) is (<0.5) then (O) is (PN)

Description

The nodes is chosen as the best parent if the values of P 1 and P 2 are larger than 0.75 and P 3 is less than 0.5. The best parent node is one with a high-energy state, a high node degrees, and a low ETX.

Rule 3

IF (P_1) & (P_2) is (<0.5) & (P_3) is (>0.75) then (O) is (NPN)

Description

The node is considered a non-parent node (NPN) if the product of its P 1 and P 2 probabilities is less than 0.5 and its P 3 chance is more than 0.75.

Similarly, 27 rules (because there are three factors and different membership criteria (Low: 0.5, Medium: among 0.5 and 0.75, High: > 0.75)). The FBL determines the best educator node to use for transmitting data and processing by running 27 rules.

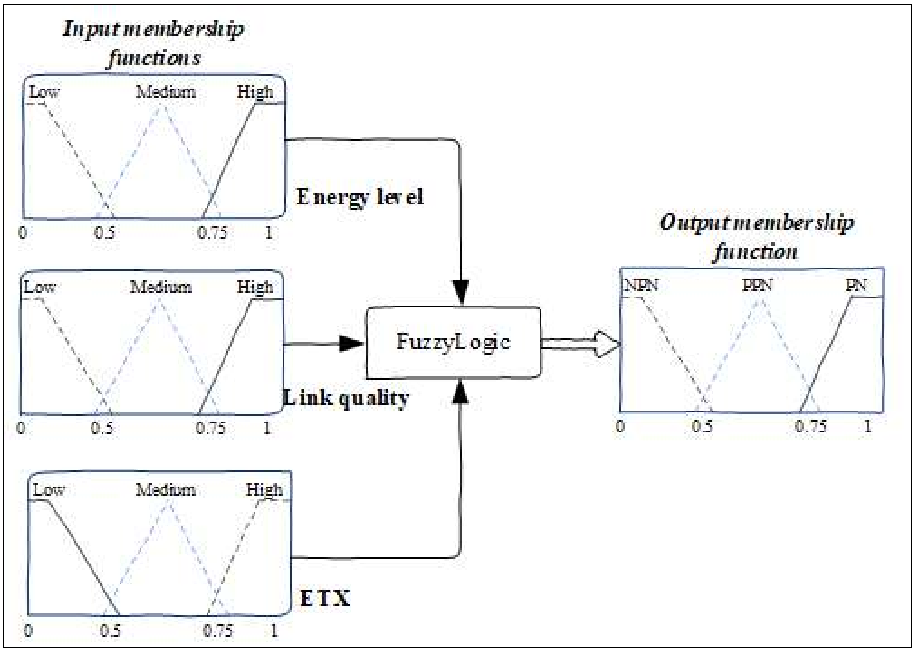

Fig 3: Membership functions in FBL algorithm

Adaptive Scheduling

- **Normal Data**
  A child node will send a parent node acknowledgement (Pn ACK) message to the selected node. The PN initiates scheduling of information from its child nodes after receiving Pn ACK. To do this, the child node will compose a request (Req) communication to its PN using the format _ ((C, S)), ((C, and S)). The required cells and slots are denoted by ((C, S)), whereas the accessible cells and slots are indicated by _ ((C, S)). PN uses this knowledge to perform a channels hop at a certain slot in order to collect data from a leaf nodes. Children's node throughput is used to dynamically adjust the desired quantity of cells and slots. Let's take a look at DODAG 1 with N=1, N=2, N=3, N=4, N=5, N=6, N=7, and R=1. In this case, N 3 and N 4 choose N 5 as their preferred parent node. Because of this, N 5 must find a way to assign cells and slots to both nodes without causing any conflicts. The demands for N 3 and N 4 are as follows: 2 cells and 1 channels and 3 cells and 2 slots, respectively. After N 5 has received both applications, it will first allocate N 4 three slots (s 1,s 2,s 3) and three cells (c 1, c 2, c 3). As a result, N 5 safely aggregates the data without any overlap. N 5 then chooses the best parent node and requests sequencing, ensuring that data is planned in a linear fashion from the children's vertices to the root node.
- **Emergency Data**
  The suggested technique aspires to enhance the data dependability without delay in the event of any data transmissions (such as data about gas leaks). If N i contains critical information (), the FBL algorithm will choose the best tree structure. Then, it decides which channel is best for PN using a number of criteria. Selecting the best route possible safeguards data integrity. Through the smart gateway, the collected data is sent on to the intelligent network layers. All of the root nodes' data is sent to the help of sophisticated, where it is sorted into two categories: routine and emergency. After then, critical information is sent to the closest edge node, while routine information is dispersed to the remaining nodes. As soon as an emergency happens, data is analysed at the edge layer and a report is sent to the appropriate parties. Users may also access regular information stored in both the cloud level and the edge layer.

Results & Discussion

This section assesses the future framework via comprehensive experiments. This part is split into two subsections: the experimental design and the analysis of results. Using the discrete event simulation ns-3.26, we simulate the suggested CoAP tri-layer design. Our computer runs Ubuntu 14.04, which we used to set up the ns-3.26 server. The code itself is in C++, but Python interfaces are provided for convenience. Here, the planned task is assessed in terms of its efficiency, delay, bandwidth, and packet delivery rate. Fig 3 displays an analysis of the industrial layer's energy use. Based on the results of the study, the suggested work uses much less energy than the baseline.

The graphical analysis demonstrates that the suggested work has reduced energy usage compared with current works. Fig 4, 5, 6, and 7 illustrates the performance of the resource consumption, latency, PDR, and throughput.

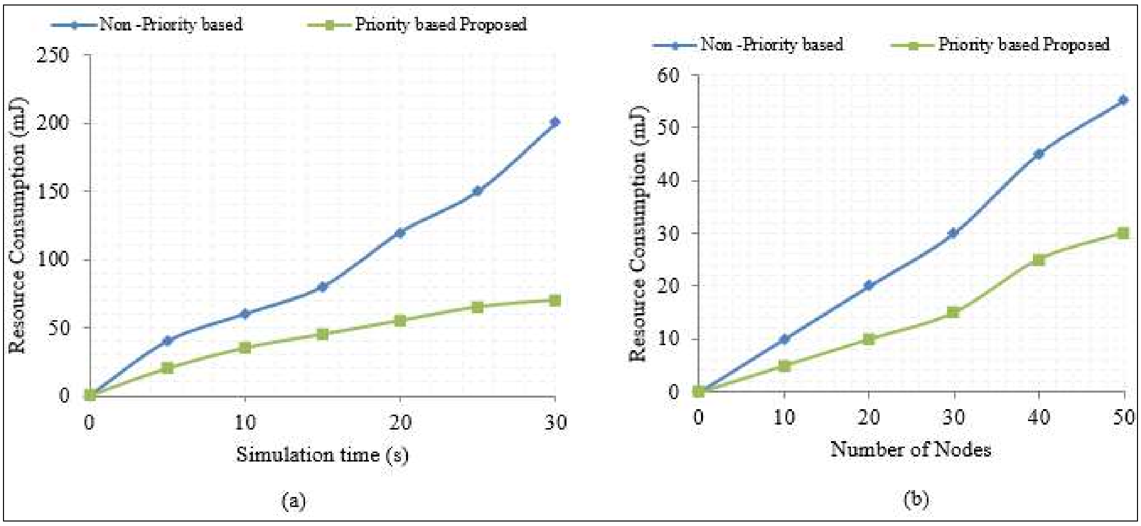

Fig 4: (a). Resource Consumption vs. Simulation Time, and (b). Number of Nodes

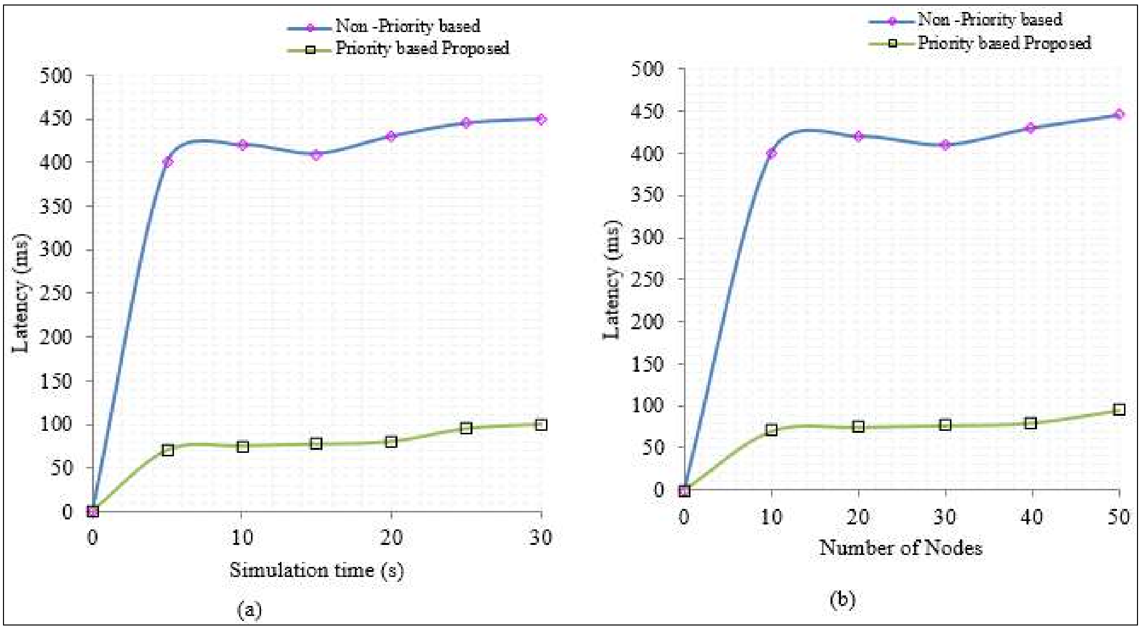

Fig 5: (a). Latency vs. simulation time, and (b). Number of nodes

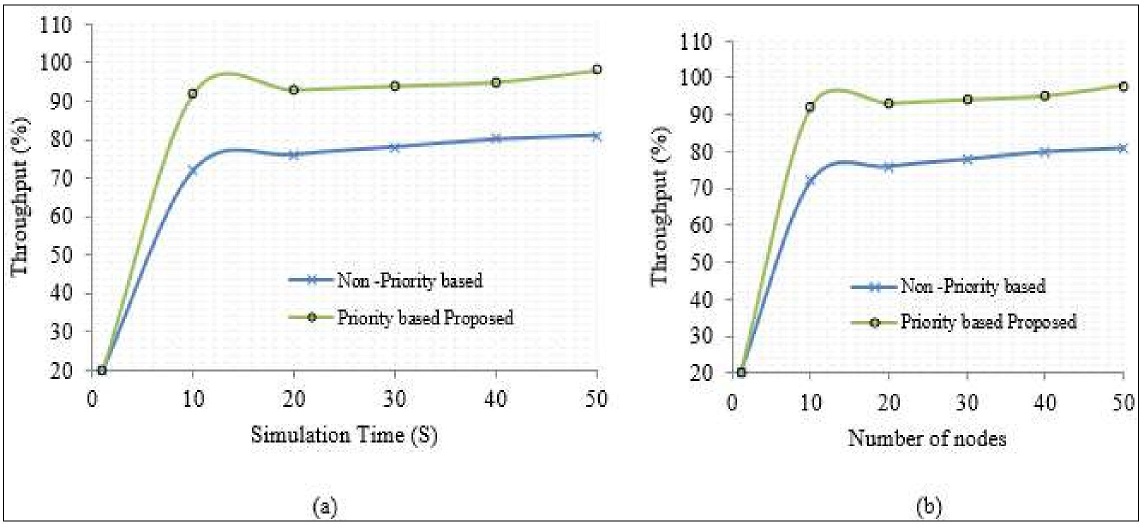

Fig 6: (a). Throughput vs. simulation time, and (b). Number of nodes

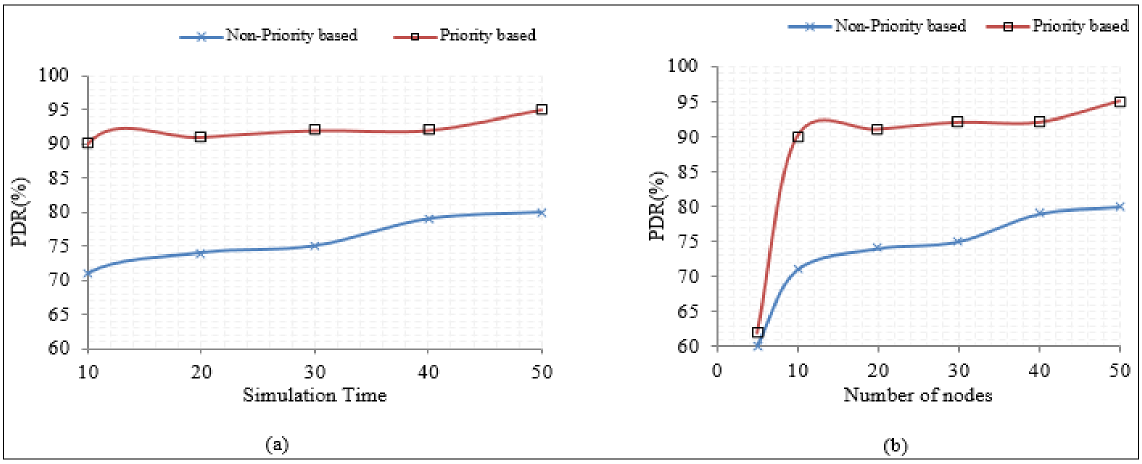

Fig 7: (a). PDR vs. Simulation time, and (b). Number of nodes

Conclusion

CoAP is a promising new technology that has emerged as the new benchmark for industrial telecommunications. The smart business world is now confronting significant challenges in the areas of scaling, latency, and fuel efficiency. Existing research has provided business with CoAP-based schedulers. But in the real world, when surprises are the norm, these works are useless. As such, our research aims to improve the scaling, delay, and fuel efficiency of CoAP-based smart businesses in the real world. A new cloud layer, an industrial layer, and a edge layer make up the edge-assisted CoAP design. Computing offers the possibility of addressing sustainability and speed issues at the edge layer. Utilizing a parent selection algorithm informed by fuzzy logic, this dynamic scheduling system prioritises the transfer of time-sensitive data packets. Because it is stored in the cloud, information on manufacturing processes is easily accessible to end users. The efficiency of the CoAP network may be evaluated with the help of a model created in network simulator-3.26. Findings from simulations reveal that the given method achieves desirable levels of signal quality, latency, and energy consumption.

References

1. Dighriri M. Data traffic modelling in mobile networks for heterogeneous types of IoT services; c2020.
2. He Y, Yin Z. Research and Design of Manufacturing Execution System in a Bicycle Smart Factory. DEStech Transactions on Computer Science and Engineering; c2019.
3. Sujatha M, Priya N, Benő A, Blesslin Sheeba T, Manikandan M, Tresa IM, *et al*. IoT a nd Machine Learning-Based Smart Automation System for Industry 4.0 Using Robotics and Sensors. Journal of Nanomaterials; c2022.
4. Serrano-Magaña H, González-Potes A, Ibarra-Junquera V, Balbastre P, Martínez-Castro D, Simó J. Software Components for Smart Industry Based on Microservices: A Case Study in pH Control Process for the Beverage Industry. Electronics; 2021.
5. Khillare P. IoT based smart home Automation System using ARM7 LPC2148 & GSM/GPRS; c2021.
6. Flórez CA, Rosário JM, Hurtado DA. Application of Automation and Manufacture techniques oriented to a service-based business using the Internet of Things (IoT) and Industry 4.0 concepts. Case study: Smart Hospital; c2020.
7. Sodhro AH, Pirbhulal S, Luo Z, Muhammad K, Zahid N. Toward 6G Architecture for Energy-Efficient Communication in IoT-Enabled Smart Automation Systems. IEEE Internet of Things Journal. 2021;8:5141-5148.
8. Katuk N, Ku-Mahamud KR, Zakaria NH, Maarof MA. Implementation and recent progress in cloud-based smart home automation systems. 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE); c2018.p. 71-77.
9. Iglesias-Urkia M, Orive A, Urbieta A. Analysis of CoAP implementations for industrial Internet of Things: a survey. Journal of Ambient Intelligence and Humanized Computing. 2017;10:2505-2518.
10. Brasilino LR, Swany DM. Low-Latency CoAP Processing in FPGA for the Internet of Things. 2019 International Conference on Internet of Things (I Things) and IEEE Green Computing and Communications (Green Com) and IEEE Cyber, Physical and Social Computing (CPS Com) and IEEE Smart Data (Smart Data). c2019. p. 1057-1064.
11. Klymash M, Maksymyuk T, Dumych S, Yaremko O. Designing the Industrial and Environmental Monitoring System based on the Internet of Things Architecture. 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS); c2018.p.187-190.
12. Hasan S, Ahmed S, Khan Marwat SN. Realization of a Novel, Lightweight and Secured CoAP Based Monitoring System in Smart Logistics. 2021 6th International Multi-Topic ICT Conference (IMTIC); c2021.p. 1-6.
13. Aazam M, Zeadally S, Harras KA. Deploying Edge Computing in Industrial Internet of Things and Industry 4.0. IEEE Transactions on Industrial Informatics; c2018. p. 1–1.
14. Chen L, Zhou P, GAO L, Xu J. Adaptive Edge Configuration for the Industrial Internet of Things. IEEE Transactions on Industrial Informatics; c2018. p. 1-1.
15. Dogo EM, Salami AF, Aigbavboa CO, Nkonyana T. Taking Cloud Computing to the Extreme Edge: A Review of Mist Computing for Smart Cities and Industry 4.0 in Africa. EAI/Springer Innovations in Communication and Computing; c2018.p. 107-132.

16. Gardašević G, Fotouhi H, Tomasic I, Vahabi M, Björkman M, Lindén M. A Heterogeneous IoT-Based Architecture for Remote Monitoring of Physiological and Environmental Parameters. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; c2018. p. 48-53.
17. Khan M, Han K, Karthik S. Designing Smart Control Systems Based on Internet of Things and Big Data Analytics. Wireless Personal Communications. 2018;99(4):1683-1697.
18. Zhang J, Hu X, Ning Z, Ngai ECH, Zhou L, Wei J, *et al*. Energy-Latency Tradeoff for Energy-Aware Offloading in Mobile Edge Computing Networks. IEEE Internet of Things Journal. 2018;5(4):2633-2645.
19. Nan Y, Li W, Bao W, Delicato FC, Pires PF, Dou Y, *et al*. Adaptive Energy-Aware Computation Offloading for Cloud of Things Systems. IEEE Access. 2017;5:23947-23957.
20. Liu L, Chang Z, Guo X, Mao S, Ristaniemi T. Multi-objective Optimization for Computation Offloading in Edge Computing. IEEE Internet of Things Journal. 2018;5(1):283-294.
21. Accettura N, Vogli E, Palattella MR, Grieco LA, Boggia G, Dohler M. Decentralized Traffic Aware Scheduling in CoAP Networks: Design and Experimental Evaluation. IEEE Internet of Things Journal. 2015;2(6):455-470.
22. Demir AK, Bilgili S. DIVA: A distributed divergecast scheduling algorithm for IEEE 802.15.4e TSCH networks. Wireless Networks; c2017.
23. Lee TH, Chang LH, Liu YW, Liaw JJ, Chu HC. Priority-based scheduling using best channel in CoAP networks. Cluster Computing; c2017.
24. Li H, Ota K, Dong M. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. IEEE Network. 2018;32(1):96-101.