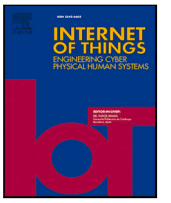
Internet of Things 27 (2024) 101312

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Self-adaptive and content-based scheduling for reducing idle listening and overhearing in securing quantum IoT sensors

Muhammad Nawaz Khan [a], Irshad Khalil [b], Inam Ullah [c,*], Sushil Kumar Singh [d], Sami Dhahbi [e], Habib Khan [f], Abdullah Alwabli [g], Mahmoud Ahmad Al-Khasawneh [h,i]

[a] Department of Computer Science, Government College of Management Sciences Talash Pakistan, Dir Lower, Chakdara 18800, Pakistan
[b] Department of Biomedical Engineering, Gachon University, Incheon, 406-799, Republic of Korea
[c] Department of Computer Engineering, Gachon University, Seongnam 13120, Republic of Korea
[d] Department of Computer Engineering, Marwadi University, Rajkot, Gujarat, India
[e] Applied College of Mahail Aseer, King Khalid University, Muhayil Aseer, 62529, Saudi Arabia
[f] Sejong University, Republic of Korea, Seoul 143-747, South Korea
[g] Department of Electrical Engineering, College of Engineering and Computing in Al-Qunfudhah, Umm al-Qura University, Mecca, Saudi Arabia
[h] School of Computing, Skyline University College, University City Sharjah, 1797, Sharjah, United Arab Emirates
[i] Jadara University Research Center, Jadara University, 21110, Irbid, Jordan

ARTICLE INFO

Keywords:
Quantum computing
Post quantum cryptography
Internet of Things
Security analysis
IoT security
Delay
Overhearing
Idle listening

ABSTRACT

Today is the age of superconductivity where each object connects in a cascading manner to other objects, allowing for seamless integration of real-world objects into the digital domain of the Internet of Things (IoT). These objects collaborate to deliver ubiquitous services based on the user mode and context. For more real-time applications, IoT is integrated with quantum computing technologies and tools for enhancing the conventional structure into more different aspects, revolutionizing the processing speed, enhancing communication, and increasing security features. All these objects are equipped with sensors that collect real-time data from their surroundings and share it with neighboring objects. This data is then broadcast into the environment, enabling users to access services without understanding the underlying complex and hybrid IoT infrastructure of heterogeneous devices. These minute and plugable sensors are capable of data collection and are always busy handling data management. However, these sensors often have limited resources, creating significant issues when dealing with massive and repetitive operations. Most of the time, these low-energy sensors are busy with excessive sensing and broadcasting, resulting in overhearing and passive listening. These factors not only create congestion on communication channels but also increase delays in data transmission and adversely affect system performance. To assess the network traffic for securing the IoT resources in the quantum computing environment, in this research work, we have proposed a novel scheme called "Self-Adaptive and Content-Based Scheduling (CACS) for Reducing Idle Listening and Overhearing in Securing the Quantum IoT Sensors". This scheme reduces idle listening and minimizes overhearing by adaptively configuring network conditions according to the contents of sensed data packets. It minimizes extensive sensing, decreases over-cost processing, and reduces frequent communication that lessens the overall system traffic and secures the resources from being overwhelmed. The simulation results demonstrate a 0.80% increase in delay across various baud rates, resulting in a general increase of 0.44 s. Moreover,

* Corresponding author.
E-mail address: inam@gachon.ac.kr (I. Ullah).

https://doi.org/10.1016/j.iot.2024.101312
Received 17 May 2024; Received in revised form 2 July 2024; Accepted 29 July 2024
Available online 31 July 2024

it ensures a notable 22.23% reduction in BER and lowers energy consumption by approximately 20%, which is actual energy enhancement in the connected system.

1. Introduction

The Internet of Things (IoT) is an inter-network of things that connects everything in the surroundings, like physical devices, objects, and entities [1–3]. This may include devices like smartphones, smart home appliances, and wearable devices, but nowadays IoT also connects other physical objects, infrastructure, and even living beings [4–6]. These devices, objects, and other entities are used to collect data about the surroundings and disseminate this information to other objects and to the central gateway for connecting to the global internet. Embedded sensors, software, and hybrid network infrastructure are used to collect and transmit data. The main goal of IoT is to create a hybrid network of resources where they provide services in a uniform layer [7,8]. All devices and objects are cascading and sharing information and providing services according to the context and the user's preferences. Most of these things share information and perform different actions without human intervention. There are many application domains of IoT and are used in many practical scenarios, like smart homes [9], industrial automation [10], healthcare [11], agriculture [12], transportation [13], etc. Commonly used objects in the surroundings are smart thermostats, connected appliances, wearable health trackers, industrial sensors, and smart city infrastructure [14,15]. In short, we can say that the main aim of IoT technologies is to create a fully connected and pervasive system in which different devices, objects, and appliances support heterogeneity and a variety of other tasks in the machine-to-machine ($M2M$) paradigm [16,17].

With their growing popularity, IoT devices and services are becoming more vulnerable as computational power increases, allowing intruders to easily predict the symmetric and asymmetric mechanisms employed to secure them. Quantum computing is a new emerging area where quantum algorithms solve those problems that classical computers have not processed [18,19]. Due to the enormous speed and complex number-solving algorithms, quantum computing can easily break up the logic behind the security algorithms. Using the quantum bit ($QuBit$), these algorithms can solve any combination problem with greater speed and accuracy. In the global infrastructure, IoT devices are connected in any way to customer data in routine applications, from financial transactions to health-related critical information. Large-scale IoT data sets may be processed more quickly and effectively by quantum algorithms, facilitating better decision-making and quick responses that can compromise the combination logic used in any security mechanism. IoT security in post-quantum computing mainly focuses on cryptographic techniques with specific goals [20,21].

There are many challenges in IoT devices due to this new quantum computing because traditional mechanisms will be invisible and need to be checked with new directions of $QuBit$. For more scalable and secure IoT infrastructure, the quantum techniques are used not only to check and find vulnerabilities but also for checking the long turn procedure of combination logic. The legacy and classical approaches of securities need to be revisited according to the new era of $QuBit$. There are many security techniques recommended for IoT infrastructure to be used, like Quantum-resistant Cryptography [22], Quantum Key Distribution (QKD) [23], Post-Quantum Hybrid Cryptography [24], and device authentication-based $QuBit$ [25], but there are many other approaches used for securing the network, like continuous monitoring and regulatory compliance [26,27]. Using the last two mechanisms, we also ensure the security of IoT devices by regulating traffic and adjusting the flow of bits towards the communication links [28]. Controlling and monitoring of packets aligns with system security, and it provides a better security policy, including scheduling of IoT devices and services [29].

IoT devices are objects that collect information from their surroundings using different types of sensors. Sensors are small autonomous gadgets embedded with other appliances for seamless connectivity and data collection. They are used for collecting data from the physical world and enabling the communication and intelligence of IoT objects. Sensor networks [30–32] are the core of all IoT infrastructures for collecting information from real-world scenarios and sharing it with other smart objects in the surroundings. These networks are accomplished networks of small autonomous sensors embedded with all devices [33,34]. These sensors have limited capabilities in terms of energy, processing, memory, and low-range communication models. The process of data collection by these sensors is a continuous process where, most of the time, the same data is collected again and again. Collecting a huge volume of data with redundant bits and repeating patterns is meaningless, while a good amount of energy is consumed in each sensing cycle. Continuous sensing of the same data packets not only degrades system performance but also affects the availability of resources for other tasks. There is always a need for a mechanism that minimizes idle listening [31,35] and overhearing [36] to minimize redundant sensing. Regulating sensing and communication not only enhances the energy level of these networks but also minimizes congestion and delay caused by overhearing and idle listening. In traditional approaches, embedded sensors in IoT devices always engage in idle listening, continuously monitoring the network conditions even when there are no incoming packets [35,37].

IoT is growing day to day with emerging new devices and providing seeming-less connectivity. Quantum computing is one such area that creates many challenges to all areas of computing, especially for IoT security which results in severe consequences. The adversaries in quantum IoT, have enormous processing powers and quicker decision-making facilities which allows them to penetrate systems that were previously thought to be safe [38,39]. The security algorithms employed in IoT devices today, particularly those for digital signatures and key exchange, are susceptible to new kinds of attacks brought about by the advancement of quantum computing. It improves processing speed by parallelism which enables quick data analysis and better decision making in IoT devices. The main problem with idle listening is that these sensors are not aware of what is being sensed (the content of data) or what the frequency or pattern of the sensed data packets is. However, they are always busy sensing the same and redundant packets,

and many useful sensing cycles are wasted. To minimize overhearing and idle listening, a strategy that can manage active-sleep techniques in the scheduling of sensors is constantly required. In this research work, we have proposed a "Self-Adaptive and Content-Based Scheduling (CACS) for Reducing Idle Listening and Overhearing in Securing the Quantum IoT Sensors". The proposed scheme optimizes the sensing flows towards the microprocessor and minimizes the frequent communication on radio links, which ultimately results in enhanced energy efficiency and increases the lifetimes of low-energy sensors. CACS makes the system more resilient and secure by mitigating quantum techniques and $QuBit$. The main contributions of this work are as follows:

- Evaluate quantum IoT technologies that challenge traditional security mechanisms for each object in the environment. Quantum IoT technologies not only provide opportunities but also offer challenges in using the traditional approaches to security. Post-quantum techniques on IoT systems greatly affect the resources and a new mechanism is always desired.
- Evaluated the contents of sensed data packets for redundancy regularly for matching bit streams, patterns, or frequency. It is possible by applying continuous monitoring and updates, and regulatory compliance in IoT-based systems. While each device follows different states and ensures that each sensor can adapt to any other state based on a four-state model. Which thwarts the possible resource starvation attack in IoT by applying quantum algorithms.
- Implement the model with defined parameters and validate the model with different metric values. Also, verify the efficiency of the proposed scheme by testing it in different scenarios and changing environments. The overall system is tested with some criteria for various types of attacks in IoT quantum systems.

The rest of the paper is structured as follows: Section 2 is Related Work, Section 3 is the proposed solution (Self-Adaptive and Content-Based Scheduling (CACS) for Reducing Idle Listening and Overhearing in Securing the Quantum IoT Sensors) in detail, Section 4 is the Working Procedure of the CACS system in Distributed IoT Environments, Section 5 is the Healthcare scenario for CACS, Section 6 is the Evaluation and Section 7 CACS has compared with state-of-the-art schemes, and finally we have concluded the paper into Section 8.

2. Related work

IoT devices are mostly designated for providing automated services with minimal human intervention. These devices are always optimized for efficient use of energy for long periods that enable real-time data collection and continuous monitoring of various crucial processes. Continuous monitoring and regular compliance are necessary for reliable services. Many techniques have been used for the officiant of the use of resources, including processing capabilities and communication over radio links. Some common techniques are duty cycling [7], sleep modes power managements [40], data compression and aggregations [2,41,42], adaptive power samplings [43], dynamic voltage scaling [44], context awareness and task management [45] and many others.

The First scheme that has ensured a quantum approach for securing IoT is **FM-PQC** [27] and it is a framework developed to help businesses with this transition. The framework is easy to connect with various corporate frameworks and makes it possible to identify cryptographic assets efficiently. **DT-IoT** [35] is a duty cycling mechanism suggested for efficient energy usage and enhancing energy efficiency in IoT devices. They have discussed many directions like MAC-based adaptive duty-cycling for synchronization. Another approach, **EE-IoT** [46] is used for enhancing energy by optimizing power consumption using ANN. It explores the recent techniques for duty cycling. It also calculates the minimum eigenvalue and SNR values for comparison and evaluation. **EERS** [47] has proposed a routing mechanism for embedded-duty cycling. The sleeping/wake-up is managed by sink by considering remaining energy, convergence area, and the number of active nodes.

GCN [1,48] is based on a generic random pattern for duty cycling using a graph derived from the sensing range. The range is based on the Euclidean distance between nodes. It improves energy consumption through game theory and deep learning techniques. **LOS** [49] is suggested for lightweight duty cycling operations and scheduling based on periodic updates of the accumulator's state of charge. It improves energy enhancement by using periodic provisions. **DCGR** [50] is based on a prediction system utilizing the Kalman filter algorithm. Node scheduling is determined by employing covariance values. It minimizes hop count, delay, and latency at an enormous level. **OREM** [24] is used to minimize sleep leakage and restore the power lost due to sleeping schedules. It uses a fully digital Eloss optimizer and an ARM M0 processor to achieve high results compared with other approaches in duty cycling. **OSSEA** [51] is mainly designed for optimizing data freshness by using the Markov decision technique. It ensures the optimal cyclic scheduling policy. The Markov process proves the switching of packets with time constants. **PROA** [52] is proposed for lowering energy consumption, minimizing latency, and improving the duty cycling mechanism. This approach uses an anycast communication pattern for data transmissions. It uses high simulations and proves better results in latency and delay. **LTDCEEP** [53] has proposed a cooperative self-scheduling routing protocol based on SDN-controlled embedded networks. It calculates the shortest path using NDDP and creates a multi-path cooperative self-scheduling scheme for calculating transmission delay, sensor absorption rate, power consumption, node response rate, and packet loss rate.

QX-MAC [54] has found out the main drawback of X-MAC protocol design and suggests a new method to overcome that problem. It combines Q-learning and a bit-streaming process for activating sensors. It reduces latency and delays by using an active-duty cycling mechanism. **CDC** [55] has proposed for adjusting work cycles and planning. It uses cloud controllers and IoT sensors for the optimization of network resources in different years and calculates the actual data set from these experiments. Different parameters are energy storage status, device cycle performance, and predicted charging values. **SBS** [56] has improved DBSCAN by grouping the same sensors in each zone. SBS is used to implement adaptive dynamic sleep scheduling. It also uses a balancing technique that further utilizes Macrocells. It achieves higher success probability and power efficiency and proves that reduces energy consumption.

Table 1
Summary of literature of scheduling with quantum IoT sensors.

Scheme	Main idea	Main drawback	Pros and Cons
FM-PQC	Efficient management, ensures resilience	Delay and packet loss	PQC-based, adaptive duty-cycling
DT-IoT	enhanced energy efficiency, reliable	packet loss only	MAC-based adaptive duty-cycling
EE-IoT	Optimizing resources using ANN	Delay and Congestion	minimum eigenvalue and SNR values
EERS	embedded duty cycling	Routing Overload	Efficient sleeping/wake-up management
GCN	Generic random pattern for duty-cycling	Calculation overhead	game theory and deep learning
LOS	Lightweight duty cycling	Congestion and overhead	periodic provision
DCGR	Prediction system utilizing the Kalman filter	overhead	Minimized hop count, delay, and latency
OREM	Minimizing sleep leakage and restoring power loss	Delay and latency	Fully digital Eloss optimizer
OSSEA	Markov decision	latency and delay	Optimal cyclic scheduling policy
PROA	Lower energy consumption, minimized latency	communication overhead	Anycast, better results in latency
LTDCEEP	Cooperative self-scheduling	calculation overhead	Shortest path calculation
QX-MAC	Overcoming drawbacks of X-MAC protocol	X-MAC overhead	Q-learning and bit streaming process, reduced latency
CDC	Adjusting work cycles and planning	delay	Optimized network resources
SBS	Improved DBSCAN, adaptive sleep scheduling	energy wastage	Higher success probability and power efficiency
DSS	Feasibility of remote check-points for energy	calculation overhead	Architecture-based security, saving packet contents
LIAA	Increasing energy efficiency and minimizing delay	memory needed	BALI, CL, and RALI approaches, uniform resources consumption
SEMFI	Software-based energy management tool	complexity in processing	Analyzing power cycles, back-end and front-end components
D-TDMA	Dynamic TDMA scheme for reduced message exchanges	TDMA congestion	Efficient resource management reduced latency and duty cycle
DEMS	Optimizing sleeping schedules using game theory	congestion and Latency	Wake-up probability and latency consideration, mobile app
RAST	Efficient Radio duty-cycled network formation	Packet Loss	Reduced energy consumption, active-scan procedures
QC-IOT	accuracy, speed, and security functions	QC-based system implementation	quantum digital, smart locks,

DSS [57] has proposed the feasibility of remote checkpoints for energy consumption and memory. It also suggests an architecture-based security technique for saving the contents of packets. **LIAA** [58] is primarily designed to increase energy efficiency and minimize delay. It combines BALI CL, and RALI approaches for achieving uniform resource consumption. The results from the simulation prove the usefulness of the scheme. **SEMFI** [59] has introduced software-based energy management tool. Which are used for analysis of the power cycles of IoT devices without any other tool. It uses back-end and front-end modules for storing and checking the data. **D-TDMA** [60] has proposed a dynamic TDMA scheme that is used to minimize message exchanges. It further elaborates dynamic slot allocation for excluding central authority from controlling all the network activities. The results prove that the performance evaluation of this approach is more energy officiant and it reduces latency, and duty cycle in an officiant manner.

DEMS [61] is based on game theory for optimizing the sleeping schedules of sensors. This approach is mainly focused on wake-up probability, latency, and traffic conditions. It also visualizes the process by creating a mobile app for sleep duration and displaying sensor readings. **RAST** [62] has proposed an efficient Radio duty-cycled which is based on network formation. It reduces energy consumption by using joining delays and uses active-scan procedures. It improves the association of time and energy consumption with other schemes. **QC-IoT** [63] have been used to amalgamate both IoT and QC for increasing accuracy, speed, and security functions. It ensures the security of all the devices by using quantum digital marketing, quantum-secured smart locks, and quicker processing at IoT endpoints. This mechanism also used some enhanced techniques like securing IoT with QC, and network optimization in IoT using QC. **Res-QCNN** [64] is a deep learning for IoT platform analysis training method. Which combines the quantum neural network using residual structural block. It works like backpropagation within conventional neural networks and it proves resilience against noisy data and completes learning a unitary function. **QIQW** [65] is based on a blockchain framework for safe data transfer between Internet of Things devices. It uses quantum hash functions instead of traditional cryptographic hash functions. Its major goal is it allow IoT nodes to fully control their records and exchange their data with other nodes in an efficient manner. It resists impersonation and messaging attacks. **TFQ-IoT** provides an analysis of the fundamentals of quantum computing and Identifies a variety of attacks on quantum IoT devices. It uses and combines different security mechanisms for enhancing Internet of Things security and guaranteeing data consistency and security in quantum cryptography.

QPSO-SP [66] is a new quantum-inspired particle swarm optimization-based service placement technique. It increases the system's throughput, energy consumption, latency, and computing load while achieving the desired service placement. It also provides QP for an all-inclusive solution for the installation of IoT services in an EC setting. **PEPC-IoT** [67] is a modular Reduction and anti-circular Rotation Column-based Multiplication. It is a column-based multiplication method that rotates one cycle at a time. The concept was put into practice on FPGA and TSMC platforms for better results and improved system performance. **PQD-IoT** [68]

is based on distributed ledger (DL) technology and combining these two technologies presents certain difficulties. DL-for-IoT, a quantum-secure DL for IoT contexts. It proves an efficient, and compact one-time signature (OTS) system called DL-OTS. **CAQ-IoT** [69] is a safeguard of patient privacy in the Internet of Things-based healthcare systems. It is based on quantum walks and there are two stages of the suggested cryptosystem technique substitution and permutation. It ensures better results in the numerical analysis of data and performers better in health-related environments. **NQPSOOM** [70] Utilizing the logistic chaos perturbation technique, the intelligent optimization algorithm's ability to escape the local optimal solution performed well. Using the five perceptual task computing offloading techniques in the last three years, it takes into account the convergence of the perceptual offloading model. **QC-6G** [71] uses the 6G technology and it enters a new era of digital communication that promises seamless and ubiquitous connections. Combining QC into these networks is seen as a potential way to handle the complex computational and security needs of this revolution. **BQC** [72] is an effective, scalable, and secure technique for securing Quantum Cloud. It emerged from Blockchain technology and QTM to improve the viability and security of the suggested architecture.

All these mechanisms are useful in terms of scheduling with quantum security mechanisms and adaptation of scheduling policies for secure IoT system resources. The summary of all these schemes is presented in Table 1 with prose and cons.

Problem Statement: Existing resource consumption and security schemes with quantum mechanisms have primarily focused on aggregating and forwarding data with minimal resource usage, mainly targeting upper network layer policies. However, most of these schemes lack a proper method for checking real-time traffic conditions and data assessment activities within the quantum IoT sensors. To overcome this limitation and make the system more resilient in terms of proper checking and validation, through real-time content assessment and monitoring, there is always needed a novel scheme. The primary goal of the suggested strategy should be to improve network performance and efficiency by intelligently scheduling sensors in different states. It should carefully forward only important data and prevent sending redundant data packets. By dynamically adjusting quantum IoT sensors based on the content and relevance of the sensed data, it should optimize resource consumption while ensuring timely and accurate data transmission to mitigate the system attacker. In the following section, a detailed discussion of CACS is provided with the system model, energy model, and four-state model.

3. Self-adaptive and content-based scheduling (CACS) for reducing idle listening and overhearing in securing the quantum IoT sensors

Self-Adaptive, and Content-Based Scheduling (CACS) presents a new approach used for optimizing the data collection and transmission processes within quantum IoT sensors. The key concept of CACS revolves around the utilization of data packets and the implementation of four distinct sensor states. These states are determined by the energy levels of the IoT sensors, enabling them to adapt their state based on two main factors: the repeating patterns of bits detected during data collection and the remaining energy. By minimizing redundancy and promoting uniform energy usage, CACS significantly extends the lifetime of individual sensors and enhances the overall performance of quantum IoT sensors. The main rationale behind CACS is explained in the following points:

- Evaluate all methods linked to Quantum IoT sensor security and what is the concept underlying the major system vulnerabilities.
- Minimize overhearing and idle listening by checking the contents of real-time data packets produced by quantum IoT sensors. Packets inside the system are checked for similarities and each device is managed accordingly.
- Minimize the awakening period when a sensor is in sleep mode and wants to wake up to sense the surroundings. By using these dynamic state-changing policies, the overall traffic will be decreased on the communication link at quantum IoT sensors.
- Decrease congestion and delay by regulating network traffic according to the real-time traffic generated at quantum IoT sensors. It enhances system efficiency and mitigates possible system attacks which ultimately lead to the minimization of extensive sensing, reduces over-cost processing, and decreases frequent communication.

3.1. CACS system model for dispersed quantum IoT environment

Consider a system model of IoT devices that are denoted as $T_{SN-work}$, where a total of twenty devices (DN) are deployed. $T_{SN-work}$ belongs to the set N, where N represents natural numbers (1, 2, 3, ...). Within $T_{SN-work}$, there are multiple immovable homogeneous (DN), denoted as $DN1, DN2, DN3, DNk$, where k is a natural number. These DN have omnidirectional, two-way communication capabilities and establish random connections with other devices (DN_i) and a designated central device (DS_i). A typical IoT scenario with system topology and IoT sensors is shown in Fig. 1.

In $T_{SN-work}$, three types of data are transmitted: Broadcast Messages ($B_{Message}$), Sensing Values (S_{Values}), and Aggregated Data (A_{Data}). The DS broadcasts $B_{Messages}$ to all DN_s for discovering their positions within $T_{SN-work}$ and for forming a zone of devices (NT_{zone}). Upon receiving a $B_{Message}$ from the DS, an DN broadcasts necessary information, including its unique identity (IDk) and hop count (HP) to reach the DS. Through the $B_{Message}$, the DS obtains an overview of the system topology. The DS sends a D_{CH} message unique to a specific DN based on its degree and location. The DS also broadcasts a message containing the same ID_{DS} to all surrounding DN_s, informing them about its presence. Consequently, all SDN_s transmit their S_{Values} to the other DS, which aggregates the sensed data along with the corresponding $DN\ ID_s$. The DS then forwards the aggregated A_{Data} to the CS for further processing. Initially, all DNs are deployed in an active state and broadcast messages to their neighboring DNs. Consider the system topology $T_{SN-work}$, consisting of twenty SNs: $DN1, DN2, DN3, DN4, DN5, DN6, DN7, DN8, DN9, DN10, DN11, DN12, DN13, DN14, DN15, DN16, DN17, DN18, DN19$ and $DN20$, with three CN, CN_1, CN_2 and CN_3.

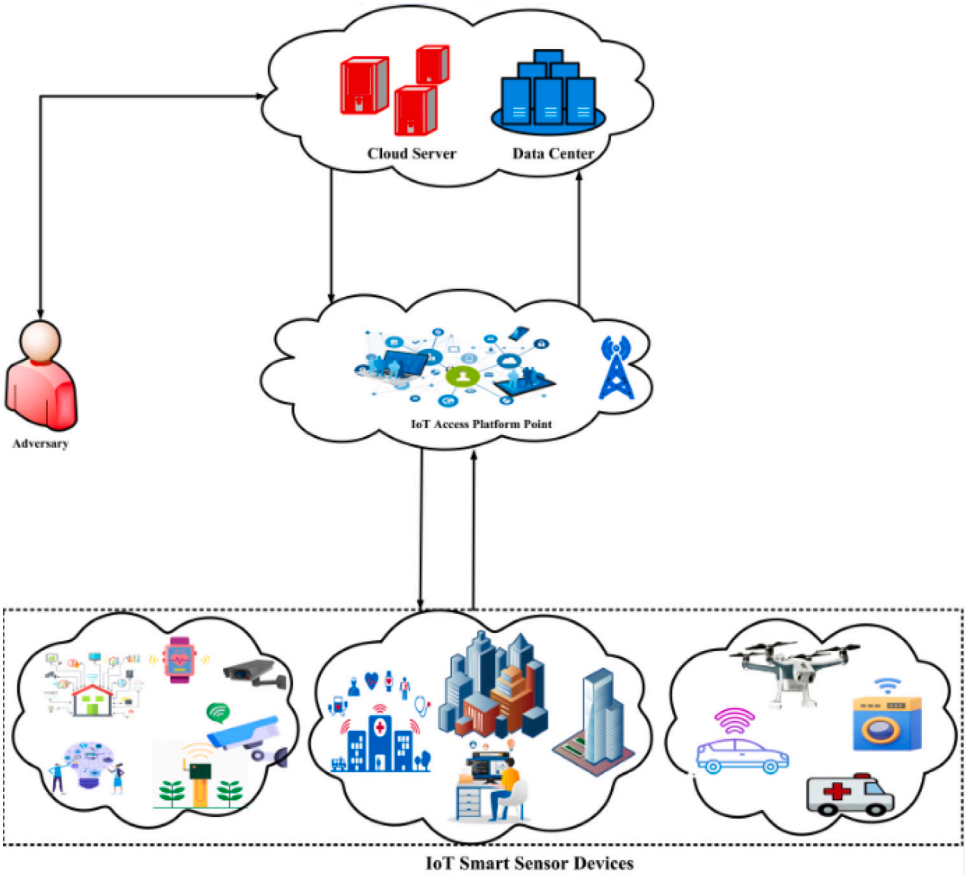


Fig. 1. Critical IoT environment with many DNs.

Some of the connections with one CN_n of these DN_n are as; $DN1, DN2, DN5, DN6, DN9$, $DN10$ and $DN14$v are connected to $CN1$, while $DN3, DN4, DN7, DN8, DN11$, $DN14$ and $DN12$ are connected to $CN1$, and in the same way $DN13, SD14, DN15$, $DN16, DN18, DN19$, and $DN20$ are connected to CN_3. Mathematically, it can be expressed as:

$$CN2 \longleftarrow DN1, DN2, DN5, DN6, DN9, DN10, DN13 \tag{1}$$

$$CN1 \longleftarrow DN3, DN4, DN7, DN8, DN11, DN12, DN14 \tag{2}$$

$$CN3 \longleftarrow DN15, DN16, DN17, DN18, DN19, DN20 \tag{3}$$

These CN_n are responsible for collecting S_{Values} from all connected DN_n. After the necessary operations like aggregations and appending ID_n, these packets are sent to DS. It can be expressed in equations as:

$$CN2 \longleftarrow ID_1 \parallel \partial_1 + ID_2 \parallel \partial_1 + ID_5 \parallel \partial_1 + ID_6 \parallel \partial_1 + ID_9 \parallel \partial_1 + ID_{10} \parallel \partial_1 + ID_{13} \parallel \partial_1 \tag{4}$$

$$CN1 \longleftarrow ID_3 \parallel \partial_1 + ID_4 \parallel \partial_1 + ID_7 \parallel \partial_1 + ID_8 \parallel \partial_1 + ID_{11} \parallel \partial_1 + ID_{12} \parallel \partial_1 + ID_{14} \parallel \partial_1 \tag{5}$$

$$CN3 \longleftarrow ID_{15} \parallel \partial_1 + ID_{16} \parallel \partial_1 + ID_{17} \parallel \partial_1 + ID_{18} \parallel \partial_1 + ID_{19} \parallel \partial_1 + ID_{20} \tag{6}$$

After selecting the appropriate DN_n as a CN based on the $B_{Message}$ in the CN selection process, CS obtained and knew about $T_{SN-work}$, including the locations of the CN_s.

3.2. Four state transition model

In a typical sensor in an IoT scenario, several components work together to enable its functionality. These components include a microprocessor SC_{MP}, a magnetic sensor SC_{MS}, a radio module SC_{RM}, and an embedded battery as a power source. The

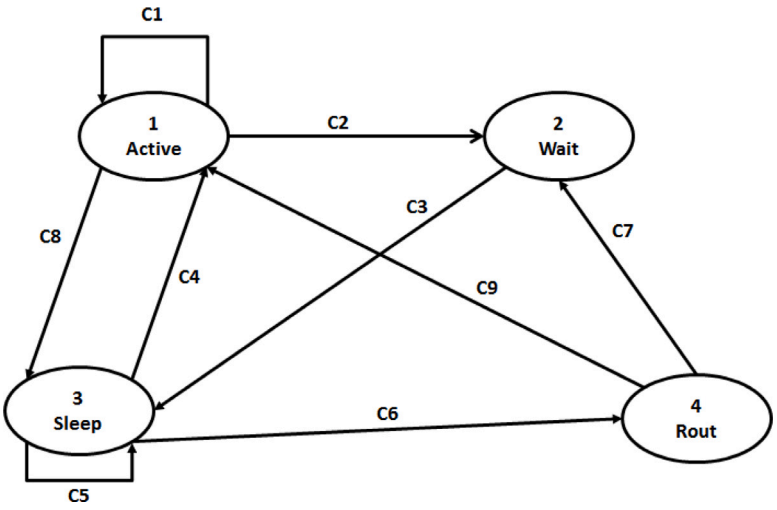


Fig. 2. Four State Transition Model.

microprocessor SC_{MP} is responsible for managing all computations within the sensor node. It processes the data collected by the sensor and controls the power consumption of the device. To optimize energy usage, the SC_{MP} has two operational modes: active and sleep. Alongside the SC_{MP}, the sensor incorporates a SC_{MS} to detect changes and collect data from real-world scenarios. The magnetic sensor has two operational modes: active and idle. When the sensor is actively collecting data, it operates in the active mode, and it switches to an idle mode when not in use. Another important module inside the IoT sensor is SC_{RM}, which is responsible for transmitting the collected data to a central location. Similar to the SC_{MP}, the SC_{RM} has two modes: active and sleep modes. When data transmission is in progress, the radio module operates in active mode, while it enters sleep mode when there is no data to be sent. To power all the components of the sensor node, an embedded battery is included as the primary power source. This battery ensures that the sensor node can function for extended periods without relying on an external power supply. The Four State Transition Model has been shown in Fig. 2.

3.3. CACS's energy model in distributed quantum IoT sensors

Using internal components or modules inside a DN_n, many operating states can be defined. These states of DN_n are the amount of energy when it processes an event. Internal components of DN_n are SC_{MP}, SC_{MS}, and SC_{RM}, however, in these components, SC_{RM} consumes more energy, and all the major operations can be linked to its use in any operation, as well as an embedded battery as a power source. We have combined these three components and derived a model for $DN-n$. Any DN_n can achieve any state defined in this model. The energy consumed by any DN_n is power consumption by each of these components combined with individual transactions. We can write it as:

$$P_{DN} = P_{MS} + P_{PM} + P_{RM} \tag{7}$$

The total power consumed in a DN_n is equal to the power consumed by all components. As power is relevant to the flow of current with resistance, it can be written as:

$$P = VI = I^2R = V^2R \tag{8}$$

For energy, and with time constraints, it can be expressed as:

$$P = V(t).I(t) = V(t).I(t) \tag{9}$$

In Eq. (9), the flow of current varies concerning time (t), and voltage also changes according to current consumption. We can write the variation in these two factors as:

$$E = \int_{ini}^{fin} V(t).I(t)dt \tag{10}$$

Eq. (10) is used for calculating the energy consumed by any DN_n concerning time. The current and voltage are directly related to power consumption in terms of energy.

Using the above equations, energy in each component inside DN_n can be identified and calculated in the following sections:

3.3.1. Energy in micro-processing unit

SC_{MP} operates in three distinct operational modes: active, idle, and sleep modes. Each of these states consumes energy at different levels. Energy consumption occurs at two levels: during actual task performance and state changes.

$$\xi_{MP} = \xi MP_{state} + \xi MP_{chang} \tag{11}$$

$$\xi_{MP} = \sum_{i}^{n} PMP_{state}(i) TMP_{state}(i) + N \sum_{j}^{m} (j) \xi MP_{change}(j) \tag{12}$$

3.3.2. Energy in radio module

The SC_{RM} operates in three distinct modes: active, idle, and sleep. In the active mode of operation, it is engaged in transmitting or receiving sensed data packets. Each of these states consumes energy at different levels, depending on the specific functions being performed. SC_{RM} serves as a transceiver, responsible for the communication of data between the DN_s and other network components. These three modes correspond to three distinct energy levels, each associated with specific operational characteristics. We can express it as:

$$\xi_{RU} = \xi_{Transmit} + \xi_{Reciev} + \xi_{idle} + \xi_{sleep} \tag{13}$$

$$\xi_{RM} = \int_{ini}^{fin} (P_{Transmit} L_i / R)(t) dt + \int_{ini}^{fin} (P_{Receiv} L_i / R)(t) dt + P_{sleep} T_{sleep} + P_{idle} T_{idle} \tag{14}$$

$$\xi_{RM} = \int_{ini}^{fin} (V_{Transmit} . I_{Transmit} L_i / R)(t) dt + \int_{ini}^{fin} (V_{Transmit} . I_{Reciev} L_i / R)(t) dt + V_{Transmit}(I_{sleep} T_{sleep} + I_{idle} T_{idle}) \tag{15}$$

Where ξ_{RM} is the amount of energy consumed in SC_{RM} and $\xi_{Trasnsmit}, \xi_{Reciev}, \xi_{idle}, \xi_{sleep}$ by SC_{RM} in different operations. $V_{Transmit}$ is the voltage for current $I_{Transmit}$. R is the rate at which data is transferred between two SN_n.

3.3.3. Energy in sensing module

The SC_{MS} is used to receive data from surroundings and send it to SC_{MP} for code fetching and further processing. The amount of energy used in this component is:

$$\xi_{MS} = N \sum_{i=1}^{j} MS_{(change)}(i) \xi_{SM(change)}(i) \tag{16}$$

$$\xi_{MS} = \int_{ini}^{fin} (P_{MS(start)}(i)) + (P_{MS(end)}(i)) dt \tag{17}$$

$$\xi_{MS} = \int_{ini}^{fin} V_{transmit}(i)(I_{MS(start)}(i)) + (I_{MS(end)}(i)) dt \tag{18}$$

4. CACS working procedure in distributed IoT environment

As an automated system, it can be formally defined, and all states can be derived from Fig. 2. Mathematically, its states, conditions, transitions, and mappings can all be expressed.

Four States are four distinct and finite states; these are $Q = \{Active, Wait, Rout, Sleep\}$. Many conditions are used to switch from one state to another; here, these conditions are marked as C_n and are the finite set of transition labels: $E = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$. Another important parameter is the transition from one state to another. Using notation, a set T represents the transitions and is defined as $T \subseteq Q \times E \times Q$.

These are defined for each transition as:

$T = \{(Active, C_1, Active), (Active, C_8, Sleep), (Active, C_2, Wait), (Wait, C_3, Sleep), (Sleep, C_5, Sleep), (Sleep, C_6, Rout), (Sleep, C_4, Active), (Rout, C_9, Active), (Rout, C_7, Wait)\}$.

While the q_o is the $Active$ state, from which each time the network starts. In the same way, l is the mapping that associates with each state of Q the finite set of elementary properties that hold in that state.

$l = Active \mapsto$ {1, 3, Active, Sleep}; $Active \mapsto$ {2, Wait}; $Wait$={3, Sleep};

$Sleep$ ={3, Sleep, 4, Sleep, 1, Active}; $Rout$={1, 2, Active, Wait}

From the above definitions and notations about the Four State Transition Model, different states can now be derived from energy consumption. All possible states are eighteen (18), but too many states also lessen system efficiency in terms of changing states from one to another. Another problem with more states is the extra overload in terms of re-configuring the states and broadcasting messages.

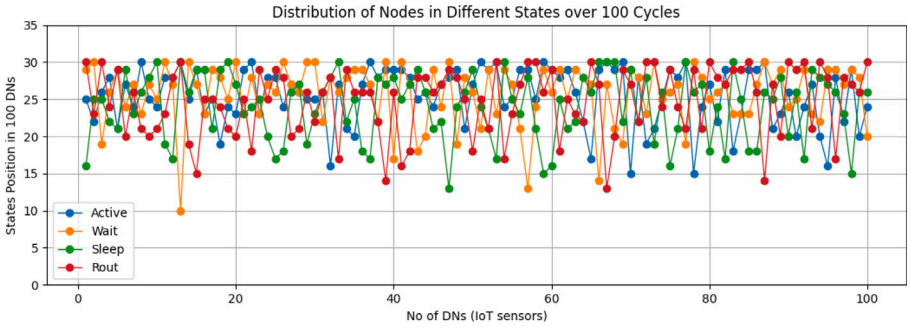


Fig. 3. CACS Four States of IoT Sensors in 100 Equal Cycles.

4.1. Four states in CACS

To optimize the number of states in the CACS, the three fundamental modules of the DN_n are combined, resulting in sixteen possible cases. However, only four optimized and useful states are used, which ensures energy efficiency and full connectivity among the DN_n. These four states are accessible to all nodes in the system, with each state theoretically allocated 30% of the total DN_n. The Route-state is specifically reserved for routing DN_n responsible for data reception, forwarding, and event sensing. 100 DN_n have experimented with a state transition model with the restriction that no state can occupy more than 40% of all the states and never gain the state less than 10%. The behavior of these 100 DN_n is mapped in Fig. 3.

4.2. CACS mechanism

In each sensing cycle, DN_n transmits data packets; A_{Data} is transmitted to CN_i. Let ϕ be the amount of inaccuracy in each sensing cycle with Real Time Clock (RTC). The idle listening can be derived for each DN_n for $B_{Messages}$ and for A_{Data}. For each C_i, idle listening for $B_{Messages}$ can be expressed as: $T_C = 2\ \phi\ T_{i-j}$ Where T_{i-j} is the time when a DN_i transmits data for another DN_j. While the idle listening for A_{Data} can expressed as: $T_{Data} = 4\ \phi\ \xi$ Where $\xi \in \{0, T_C\}$ and its value is average if $T_{Data} \leq T_C$ is the time when a DN_i transmits data for another DN_j. Energy consumed in transmitting and receiving $B_{Messages}$ is defined as $E_{Transmit}$ and $E_{Receive}$ while power consumed in the form of energy in idle listening is P_{Id}.

The energy consumed in any sensing cycle is the amount of energy consumed when a DN_i transmits data ($B_{Messages}$, S_{Values}, A_{Data}) to another DN_j in nearby configurations, while in multiple hierarchical networks, many routing DN_{Routs} are involved in receiving and forwarding these packets until they reach a central gateway. Another important thing is the time of wake-up for any SN_n and this time can be calculated as T_{idle}. With minimum values of T_{idle}, the network will be more active in time series, and DN_n will wake up in very little time. Now the total power consumed by this process can be calculated as follows:

$$P_{Total} = P_{Transmit} + P_{Receive} + P_{Idle} \tag{19}$$

Where $P_{Transmit}, P_{Receive}, P_{Idle}$ are power consumption in transmitting, receiving, and idle states inside the network respectively. We can write the $P_{Transmit}$ in time as,

$$P_{Transmit} = E_{Transmit}/T_{Transmit} \tag{20}$$

While the transmission of any data in the radio module will be,

$$P_{Receive} = (E_{Receive} + T_{Transmit} 2\phi P_{idle})/T_{Transmit} \tag{21}$$

The power consumed in wake-up SN_n will be,

$$P_{idle} = (T_{Transmit} 2\phi P_{idle})/T_{Trans} \tag{22}$$

Eq. (22) is the amount consumed in waking up the sleep DN_n. In most of the cases, it is assumed that P_{idle} is zero and for this, the equation for T_{Trans} is the first derivative concerning time is,

$$T_{Trans} = \sqrt{T_{Trans}(E_{Trans} + E_{Recv})/2\phi P_{idle}} \tag{23}$$

Now Eq. (21), (22) and (23) are the amount of energy in waking up the sleep states DN_n. These equations can be embedded with Eqs. (14), (15), and (18) for adding the amount of these energies (14), (15), and (18), we can get the actual amount of energy with wake-up time. We can write as,

$$\begin{aligned}\xi_{PM} = &\int_{ini}^{fin} (E_{Transmit}/T_{Transmit} L_i/R)(t)dt + \int_{ini}^{fin} ((E_{Receive} \\ &+ T_{Transmit} 2\phi P_{idle})/T_{Transmit} L_i/R)(t)dt + P_{sleep} T_{sleep} + (T_{Transmit} 2\phi P_{idle})/T_{Trans} T_{idle}\end{aligned} \tag{24}$$

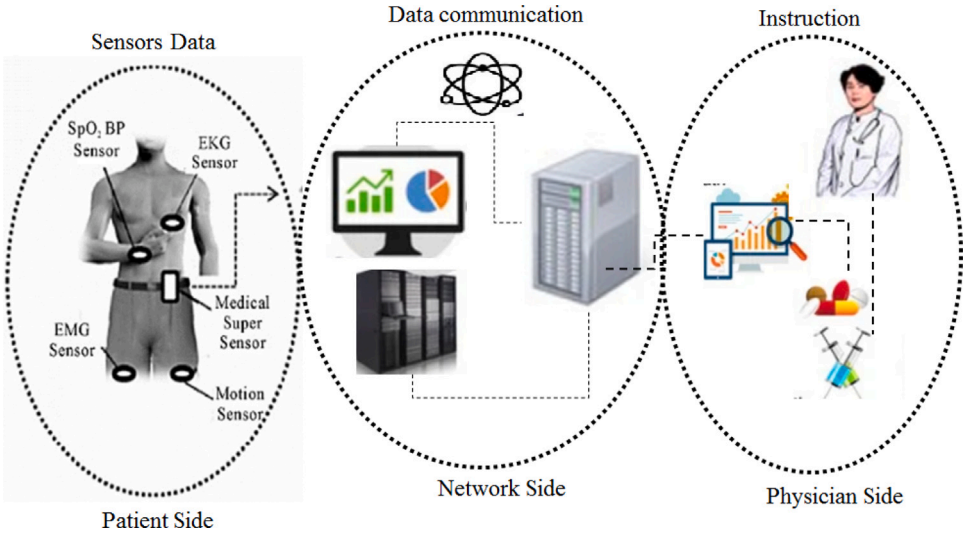


Fig. 4. CACS in Healthcare application.

Table 2
Experimental results of Bio-Sensors for different parameters.

Type of sensor	Actual value	Estimated value	Error Margin	Average Error Margin
Heart Beats	72.00	72.81, 73.31, 70.76, 73.91, 72.99, ...	1.98, 1.23, 1.75, 1.06, 1.89, ...	1.54
Blood Pressure	1.50	122.03, 119.78, 121.09, 121.88, 123.09, ...	0.53, 1.87, 3.35, 3.94, 0.83, ...	1.93
Temperature	37.00	38.03, 37.98, 37.89, 36.99, 36.89, ...	0.83, 1.08, 1.00, 0.57, 0.18, ...	0.99

$$\xi_{RM} = \int_{ini}^{fin} (V_{Transmit}.I_{Transmit}L_i/R)(t)dt + \int_{ini}^{fin} (V_{Transmit}.I_{Reciev}L_i/R)(t)dt + V_{Transmit}(I_{sleep}T_{sleep} + (T_{Transmit}2\phi P_{idle})/T_{Trans}) \quad (25)$$

$$\xi_{MS} = \int_{ini}^{fin} E_{Transmit}/T_{Transmit} + (I_{MS(end)}(i))dt \quad (26)$$

These three Eqs. (24) and (25), and (26) are the amount of energy consumed in SC_{MP}, SC_{RM}, and SC_{MS} respectively.

5. CACS used in healthcare

Healthcare IoT and bio-medical sensors, create a new era of data collection from the patients and are very useful in remote patient monitoring and telemedicine. Integrating quantum computing with these sensors can increase system efficiency and improve the security of such critical applications. There are many applications of healthcare but here we consider remote patient monitoring systems as shown in Fig. 4. In which the vital parameters are checked regularly by using these biosensors. These may include heart rate, blood pressure, glucose levels, oxygen saturation, and many more. Biosensors collect these data within time with the help of 4G/5G/6G to healthcare providers. They analyze the data and instruct the system to medicate accordingly. In this system, there are some inherent challenges of redundant data collection where these biosensors frequently produce massive data. These data need extensive processing and immediate data communication, resulting in idle listening and overhearing. Another problem is the massive data procured by IoT devices and biosensors, which needs rapid analysis of this vast dataset. CACS ensures the data distribution with proper scheduling of these sensors and also secures the resources from overwhelming and idle listening. This secure quantum communication technique provides ultra-secure data transmission between biosensors and IoT devices. For more comprehension, we have also tested this scenario with a data set and calculated some values after applying the four-state model and scheduling in quantum IoT sensors. The results of these experiments have been drafted in Table 2.

6. Performance evaluation of CACS with respect to QuBit

This section involves the examination and assessment of CACS under various scenarios, and the results have been visualized through multiple graphs. The evaluation utilized several parameters from the CC2420 [48,73] off-the-shelf product, which is a widely adopted standard for implementation in IoT. Table 3 presents the key parameters along with their corresponding metric values and symbolic representations. All these parameters are implemented in the NS-2 [74,75] simulator, allowing for the inspection of packets to check their contents and subsequently modify the state of the node accordingly. Various parameters have been pre-set for the

Table 3
Evaluation parameters, symbols & metric values.

Parameters	Symbols	Metrics
Simulation Terrain	A x A	200 x 200 m
Standard frequency	Frequency (IEEE)	2.4 GHz
Data Rate	D_{Rate}	250 Kbps
Baud Rate	B_{Rate}	50–400 kBaud/s
No of DN_n	DN_n	20–100
Energy supplied for DN_n	E_{DN}	0.5 J
Probability of CH	Pb_{CHs}	0.1
Rounds	R_{max}	1000
Buffer at transmission + Receive	M_{buffer}	256 Bytes
Size of S_{Data}	S_{Data}	512 bits (64 Bytes)

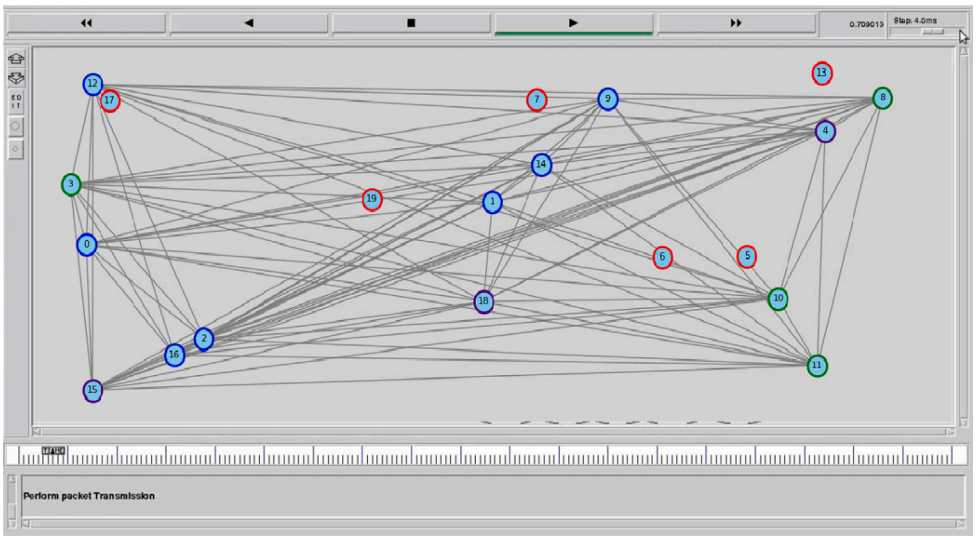


Fig. 5. CACS with 10 DNs (Nodes) connected in IoT, red DNs have no connections.

sensors within the simulation. Initially, the topology of 10 DNs was implemented with different colors. The red-color DNs are Sleep-states, green is Active, Blue is Wait and Indigo is Rout-state as shown in Fig. 5. The system has also checked for an increasing number of DNs from 20–100 and its applicability and other parameters have been checked. First, these IoT sensors are deployed randomly with no content checking and no policy implementation. The system responds with a massive data set which affects the resources of the IoT system as shown in Fig. 6. After applying the CACS policy of scheduling by eliminating the extra burden and overhead in the form of idle listening and overhearing, the quantum IoT system reshaped the structure as shown in Fig. 7.

6.1. Delay analysis in IoT sensors

The optimum delay in CACS has been calculated in two conditions, to apply with idle listening and with it. The scheme has shown a little diversion in behavior in Fig. 8. In this diagram, the delay is calculated at different data rates (50 kBaud/s, 200 kBaud/s, and 400 kBaud/s) for CC2420. The delay almost increases with the increased number of packets with time. This trend is visible for both the Idle Listening and without listening scenarios. Ignoring Idle listening, the delay is minimal while adding the factor of idle listening in total delay, the delay increases because of the time of wake-up when a DN_n is in sleep state. At the start of the simulations (0 s), the delay is also much less, and with increasing (nearly 10 s), the increase in delay becomes more pronounced. Different baud rates also affect the scenario because different baud rates produce different levels of congestion in the network. At each data rate (Baud rate), the delay for adding Idle Listening is greater than the delay for Without Listening. It proves that in the Idle Listening scenario, where the system is actively listening for incoming data, the CACS takes more time to settle compared to the scenario without listening. To find out the actual delay in each case, we have calculated as a function of delay with data rate and calculated delay values at different Baud-Rate at different times. At 50 kBaud/s, the CACS performs in Idle Listening (Delay) at 0, 1000, 2000, 3000, 4000, and 5000 ms for idle listening and without idle listening. The difference at each point is 0.79 ms. The same experiments are repeated for 200 kBaud/s, and with the same time frames of 0, 1000, 2000, 3000, 4000, and 5000 s, the difference is 0.80 ms. For 400 kBaud/s in the same scenarios, the difference is 0.80 ms. These values prove that without idle listening, the delay is higher compared to when idle listening is enabled. This makes sense, as idle listening adds some additional overhead, resulting in a longer delay for data transmission with consistency in overall experiments.

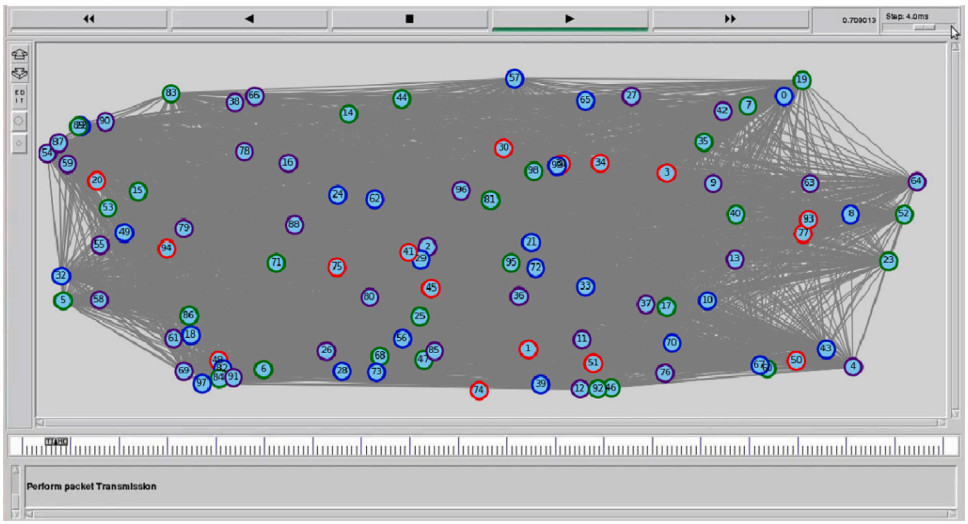


Fig. 6. CACS with 100 DNs (Nodes) connected in IoT, red DNs have no connections.

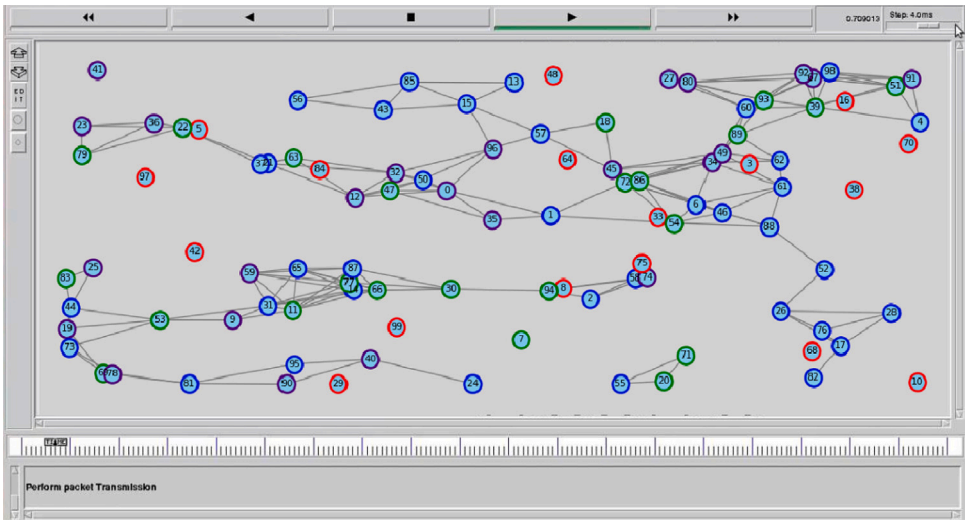


Fig. 7. CACS with 100 DNs (Nodes) connected in IoT, red DNs have no connections, after scheduling.

After faxing the Baud-Rate to 50 Baud/s with the same scenario, check the behavior of the CACS in idle listening and without idle listening. The results are quite different from the previous experiments. When idle listening is considered and added to the calculation, it affects the entire flow of data by increasing the delay. This is because of the waste of many sensing cycles in the normal flow of data. While ignoring the idle listening, the delay is much less, but still, there are 0.44 s of delay and 5000 ms of sensing cycles. On the other hand, there are 9.1 s of delay in the same cycles of 5000 ms. This delay is enormous and cannot be ignored, as shown in Fig. 9.

6.2. Bit error rate in CACS

The Bit Error Rate (BER) is calculated in both scenarios and found to have different values at different numbers of rounds. We have experimented with these scenarios when the erroneous bits received were compared to the total bits transmitted in a 100-node network with 200 × 200 terrain. The graph is mapped between some rounds versus BER, and it ranges from 0% to 10%. The results from these experiments are mapped in Fig. 10, in which the blue line represents BER with idle listening and the red line represents BER without listening. With idle listening, the BER is less compared to idle listening because, in this technique, nodes constantly listen to the channel even when they have no data to transmit and consume a good amount of energy. While applying without idle listening, they only send fewer bits as per policy, which reduces channel contention. To compare these values, there is a 22.23% difference in adding idle listening or ignoring it.

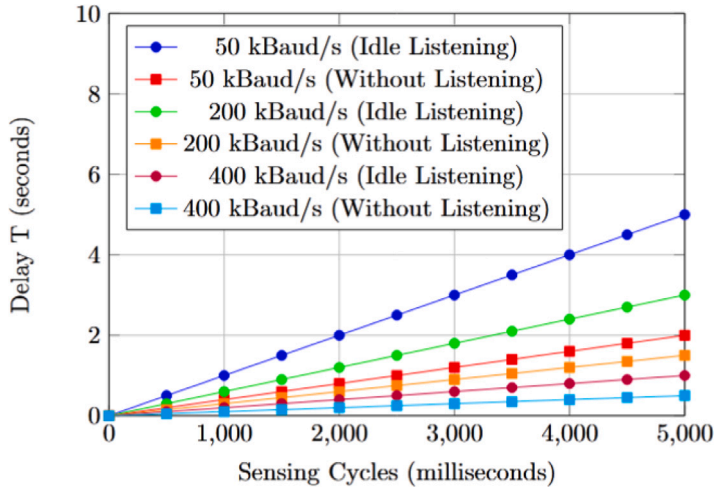


Fig. 8. CACS with different Baud Rate in CC2420.

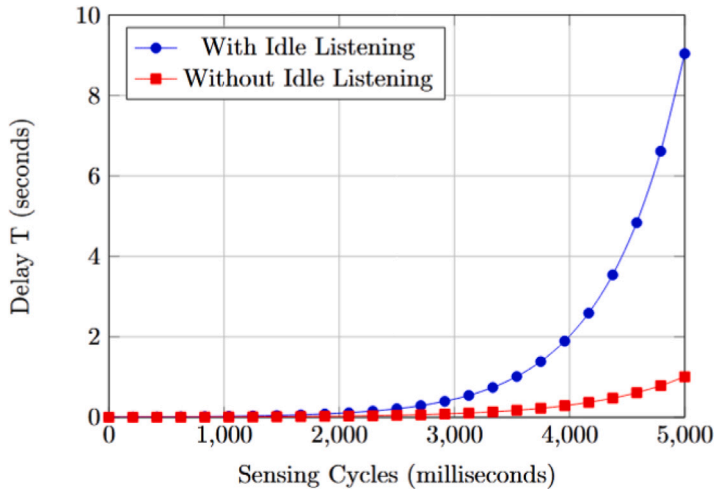


Fig. 9. CACS with same 50 Baud/s with and without Idle Listening.

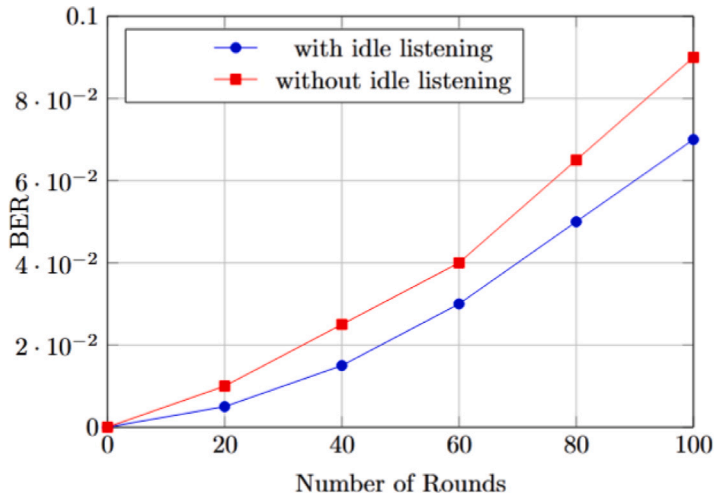


Fig. 10. BER in CACS with and without idle listening.

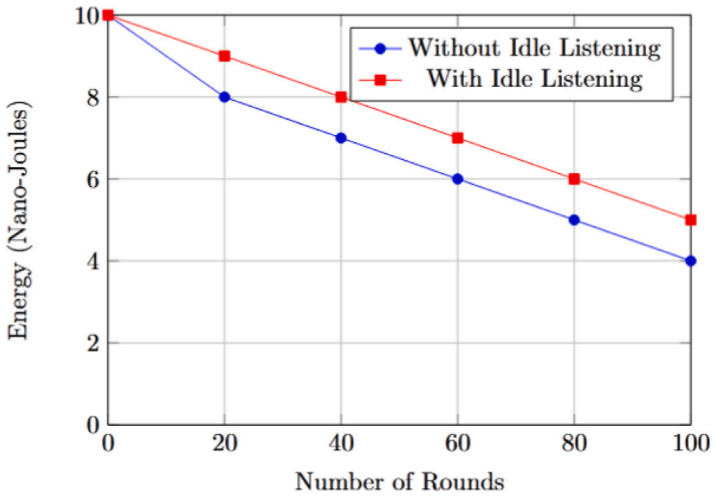


Fig. 11. Energy comparison in applying Idle listening and without idle listening.

6.3. Energy consumption in CACS

The energy of the network is a very important factor, and all the crucial tasks are streamlined by this parameter. The network is also tested in both conditions for energy. It behaves differently in both cases. The results of these experiments are shown in Fig. 11, in which energy (nanojoules) versus number of rounds (round means of a specific number of data transmissions or communication events) are mapped. In this figure, the blue line represents the energy usage when idle listening is not utilized, and the red line represents the energy consumption when idle listening is applied. These results prove that idle listening contributes to higher energy consumption because the sleep SN_n needs extra energy for the wake-up process, and these nodes consume extra energy. To calculate the difference for each round, the overall difference is 20% in both cases. It means that CACS with idle listening consumes 20% more energy than without it.

6.4. States of DN_n

In a scenario that is comprised of 100-DN_n, each device there is a sensor in any state from all four states within the four-state model. Every DN_n node can adopt any of the defined states from the model, resulting in a total of 100 states across the network. However, there is a restriction imposed to maintain balance: no state should be represented by less than 10% of the total DN_n, nor should any exceed 50%. Under this constraint, the scenario undergoes experimentation over 20 sensing cycles, monitoring the state of each sensor throughout. The objective is to ensure that all four states are consistently present in the sensors at any given time, as depicted in Fig. 12. This experimental setup aims to validate the robustness and reliability of the network in maintaining a diverse range of states across its nodes, ensuring comprehensive coverage and functionality throughout the sensing cycles.

6.5. Traffic conditions in IoT for CACS

Another parameter for analysis is the number of packets received at the central device in IoT. We are claiming that CACS minimizes traffic and lessens the number of packets. We have performed another experiment and calculated the packet flow towards CN. It is proved that first CACS increases packet flow to CN, but after some time, the flow starts declining, as shown in Fig. 13. It is because sometimes the network remains consistent and there is a need for a few packets after checking the contents of packets. The relationship between the number of rounds and the number of packets is mapped in three modes: statistics, direct messages, and probability-based. It is observed that messages at CN increase with the passage of time and the increasing number of rounds in all three conditions. CACS-Statistics and Direct Messages schemes are looking to have a more gradual and consistent increase in the number of messages, while probability-based schemes have shown more fluctuation in packets towards CN.

6.6. Post quantum CACS security analysis

CACS is also analyzed for security analysis for three types of attacks that are related to the working structure of the proposed technique. These attacks are normally launched with a generic structure and they always target the working procedure of the scheme. CACS is analyzed throughout time and its behavior is mapped in Fig. 14. The x-axis is marked with timing and the y-axis is marked as security level from 0 to 10. Each attack is mapped with proper values and it is shown that security has increased over time. It also proves the thwarting of attackers in the structure of CACS over different security measures.

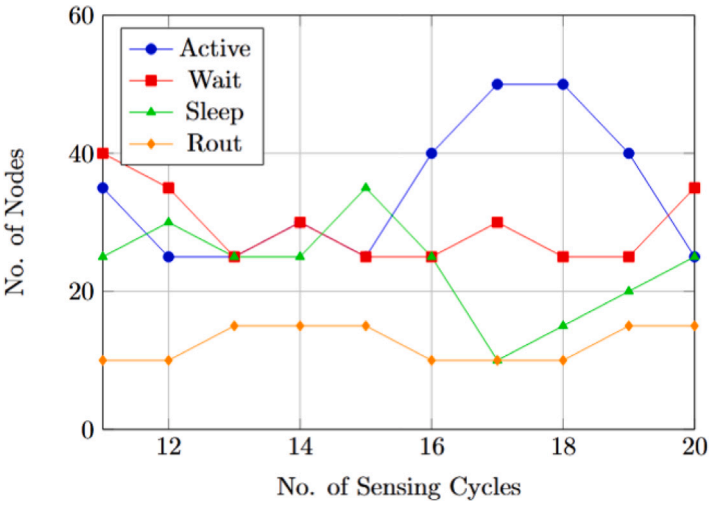


Fig. 12. Different States of Sensors in Four-State Model.

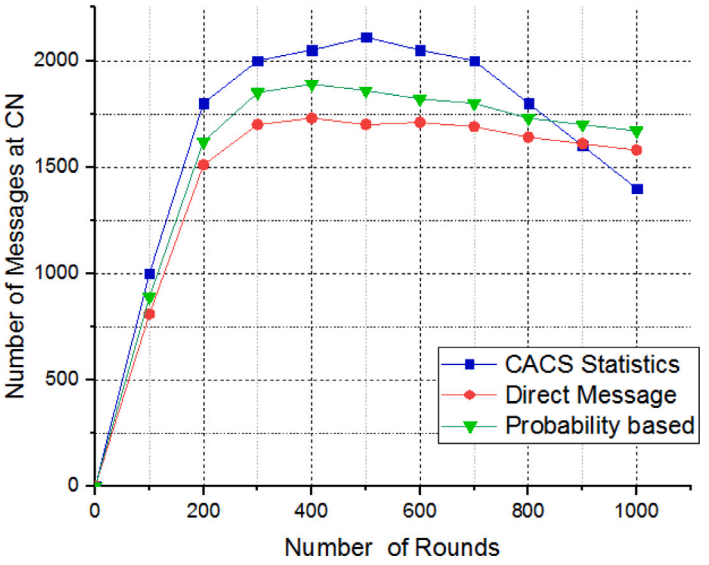


Fig. 13. Number of Messages CN in CACS, Direct, and Probability-based.

7. Comparative analysis of CACS with state-of-art schemes

In this section, CACS has been experimented with for different parameters and these values have been compared with these three recent and near-in functions protocols which are **TUA-IoT** [76], **HQCNN** [77] and **OSS-IoT** [78]. These are selected for comparing the values because the functionalities of these protocols are very near and similar to CACS. Some of the dominant parameters are comparison storage requirements for data sets, communication costs/overhead, and computational complexities.

7.1. Time cost/overhead

Let $M_x R$ be the matrix multiplication of modulo R, $V_x R$ be the vector multiplication of modulo R, $A_x R$ be the vector addition modulo R and $H_x R$ be the one-way function used in all these schemes. The total cost time is calculated for each scheme and marked in Table 4. Each scheme starts from initialization, state changing, and authentication. The total cost is calculated from these values. The Initialization phase is the same for all schemes while in state changing, authentication, and total costs vary. The effectiveness of each scheme is indicated by the time cost/overhead in milliseconds, where TUA-IoT has the greatest time cost and OSS-IoT has the lowest while the CACS is a little high cost than OSS-IoT.

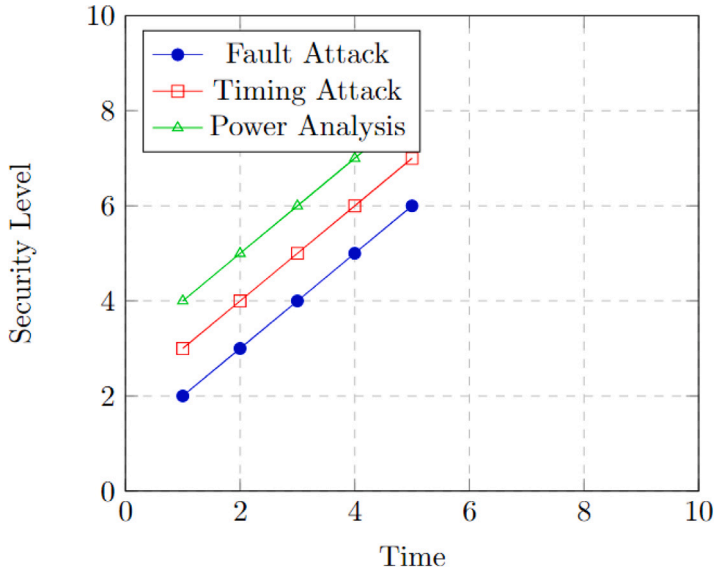


Fig. 14. Security Analysis of CACS in Timing Attack, Fault Attack, and Power Analysis Attack,.

Table 4
Comparison of computational overhead of CACS with TUA-IoT, HQCNN, and OSS-IoT.

Parameter	TUA-IoT	HQCNN	OSS-IoT	Proposed CACS
Initialization	M_xR	M_xR	M_xR	M_xR
State Changing	$2M_xR+2V_xR+2A_xR+3H_xR$	$2M_xR+2V_xR+5H_xR$	$2M_xR+3V_xR+A_xR+5H_xR$	$3M_xR+5H_xR$
Authentication	$3M_xR+V_xR+5A_xR+3H_xR$	$2M_xR+V_xR+5H_xR$	$3M_xR+3V_xR+A_xR+5H_xR$	$2M_xR+3H_xR$
Total Cost	$7M_xR+V_xR+9A_xR+5H_xR$	$4M_xR+V_xR+9H_xR$	$8M_xR+7V_xR+A_xR+7H_xR$	$5M_xR+9H_xR$
Time Cost/overhead (ms)	38.345	31.953	22.892	24.342

Table 5
Comparison of CACS with TUA-IoT, HQCNN, and OSS-IoT in complexity and computational cost.

Scheme	Complexity overhead	Computational cost
TUA-IoT	Low: O(n)	High: O(n log n)
HQCNN	Medium: O(log n)	Medium: O(log n)
OSS-IoT	High:O(n log n)	Low:O(n)
Proposed CACS	Low: O(n)	Medium $O(n^2)$

7.2. Computational complexities

The computational complexity of CACS is testified by the other three schemes, OSS-IoT, HQCNN, and TUA-IoT. The values and behavior of these schemes are noted in Table 5 and compared with CACS. The TUA-IoT has a high computational cost of O(n log n), however, it has a low complexity overhead of O(n). HQCNN sustained between medium complexity overhead with computational cost, which is marked as O(log n). OSS-IoT while maintains a low computational cost of O(n) while a significant complexity overhead of O(n log n). From these experiments, it is clear that the CACS scheme ensures a low complexity overhead of O(n) and a medium computational cost of $O(n^2)$.

7.3. Storage complexity of CACS with TUA-IoT, HQCNN, and OSS-IoT

Another parameter for comparing the behavior of CACS with others is storage complexity. There are many factors involved in storage complexity from initialization to authentication phase. Once the authentication completes, it is never initiated in the same session of data between any quantum IoT sensors. The packet size and size of the hash function H_xR is $2logn$. The communication cost with storage will be 2 logn (2 nlog(n) +11). The communication cost with storage complexity is calculated for each scheme and presented in Table 6.

Table 6
Storage complexity of CACS with TUA-IoT, HQCNN, and OSS-IoT.

Scheme	Data storage	Data length
TUA-IoT	512/1024 Bytes	2 log n(11 k log n)+1
HQCNN	512/1024 Bytes	n(2 k log n)+n
OSS-IoT	1024	5 log n(2 k log n)+1
Proposed CACS	1024	2 log n+1(k log n)+1

8. Conclusion and future work

IoT makes possible the integration of everyday objects with sensors, seamless connectivity, and computational capabilities, enabling them to collect, exchange, and act upon real-time data. This paradigm shift can change our lives, including how we interact with our homes and cities and how businesses improve activities and deliver services. It improves efficiency, convenience, and productivity across several areas, opening the way for a more intelligent, connected world. Sensors are the basic units for data collection from real-world scenarios and provide communication between these objects. However, these resource-restricted devices do not have enough resources to make it possible to use them to handle massive or repeated tasks. IoT sensors are always busy with continuous sensing and sending out data, which may result in listening in on redundant or useless data or remaining idle for a long in quantum IoT scenarios. To handle this problem of overhearing and idle listening for security and resilience against quantum IoT attacks, we have proposed "Self-Adaptive and Content-Based Scheduling (CACS) for Reducing Idle Listening and Overhearing in Securing the Quantum IoT Sensors". This technique dynamically configures network conditions based on the contents of detected packets which reduces overhearing and idle listening. It guarantees a significant 22.23% decrease in BER and minimizes energy consumption by about 20%, marking a better efficiency improvement across the quantum IoT network. CACS significantly increases energy efficiency by reducing overall network traffic and ensuring full coverage using a four-state transition model.

In the future, CACS can perform better by integrating quantum computing techniques and tools for testing each possible case of adversaries in quantum IoT sensors. Using the logic of QuBit, deep packet inspection of the traffic generated at IoT sensors can be checked and formulated for different tasks. The biosensor values should be checked and should be optimized for decision-making in health-critical applications. Although, there are many challenges in implementing CACS within post-quantum computing scenarios here we have implemented the initial sketch and it will improve with further testing and experimentation.

CRediT authorship contribution statement

Muhammad Nawaz Khan: Writing – original draft, Software, Methodology, Conceptualization. **Irshad Khalil:** Writing – review & editing, Resources, Investigation, Data curation. **Inam Ullah:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Formal analysis. **Sushil Kumar Singh:** Resources, Formal analysis, Data curation. **Sami Dhahbi:** Visualization, Validation, Resources. **Habib Khan:** Software, Resources, Investigation, Formal analysis. **Abdullah Alwabli:** Validation, Resources, Investigation, Data curation. **Mahmoud Ahmad Al-Khasawneh:** Visualization, Resources, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that the research was conducted without any commercial or financial relationships that could be construed as a potential conflict of interest.

Data availability

Data will be made available on request.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number RGP2/449/45. This research was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (RS-2023-00259004) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Ethical approval

We confirm that relevant guidelines and regulations are carried out in all methods.

References

[1] A. Imteaj, U. Thakker, S. Wang, J. Li, M.H. Amini, A survey on federated learning for resource-constrained IoT devices, IEEE Internet Things J. 9 (1) (2021) 1–24.
[2] S. Abbasian Dehkordi, K. Farajzadeh, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, M. Abbasian Dehkordi, A survey on data aggregation techniques in IoT sensor networks, Wirel. Netw. 26 (2020) 1243–1263.
[3] M.-H. Fu, Integrated technologies of blockchain and biometrics based on wireless sensor network for library management, Inf. Technol. Libr. 39 (3) (2020).
[4] A. Khanna, S. Kaur, Internet of Things (IoT), applications and challenges: A comprehensive review, Wirel. Pers. Commun. 114 (2020) 1687–1762.
[5] S.Y.Y. Tun, S. Madanian, F. Mirza, Internet of Things (IoT) applications for elderly care: A reflective review, Aging Clin. Exper. Res. 33 (2021) 855–867.
[6] M.N. Khan, H.U. Rahman, T. Hussain, B. Yang, S.M. Qaisar, Enabling trust in automotive IoT: Lightweight mutual authentication scheme for electronic connected devices in Internet of Things, IEEE Trans. Consum. Electron. (2024).
[7] S. Rani, S.H. Ahmed, R. Rastogi, Dynamic clustering approach based on wireless sensor networks genetic algorithm for IoT applications, Wirel. Netw. 26 (2020) 2307–2316.
[8] M.N. Khan, H.U. Rahman, M. Faisal, F. Khan, S. Ahmad, An IoT-enabled information system for smart navigation in museums, Sensors 22 (1) (2021) 312.
[9] M. Alaa, A.A. Zaidan, B.B. Zaidan, M. Talal, M.L.M. Kiah, A review of smart home applications based on Internet of Things, J. Netw. Comput. Appl. 97 (2017) 48–65.
[10] A. Deshpande, P. Pitale, S. Sanap, Industrial automation using Internet of Things (IOT), Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) 5 (2) (2016) 266–269.
[11] M.H. Kashani, M. Madanipour, M. Nikravan, P. Asghari, E. Mahdipour, A systematic review of IoT in healthcare: Applications, techniques, and trends, J. Netw. Comput. Appl. 192 (2021) 103164.
[12] M.R.M. Kassim, Iot applications in smart agriculture: Issues and challenges, in: 2020 IEEE Conference on Open Systems, ICOS, IEEE, 2020, pp. 19–24.
[13] F. Zantalis, G. Koulouras, S. Karabetsos, D. Kandris, A review of machine learning and IoT in smart transportation, Fut. Internet 11 (4) (2019) 94.
[14] S. Aheleroff, X. Xu, Y. Lu, M. Aristizabal, J.P. Velásquez, B. Joa, Y. Valencia, IoT-enabled smart appliances under industry 4.0: A case study, Adv. Eng. Inform. 43 (2020) 101043.
[15] P. Powroźnik, P. Szcześniak, K. Piotrowski, Elastic energy management algorithm using IoT technology for devices with smart appliance functionality for applications in smart-grid, Energies 15 (1) (2021) 109.
[16] B. Mishra, A. Kertesz, The use of MQTT in M2M and IoT systems: A survey, IEEE Access 8 (2020) 201071–201086.
[17] I. Ullah, D. Adhikari, X. Su, F. Palmieri, C. Wu, C. Choi, Integration of data science with the intelligent IoT (IIoT): Current challenges and future perspectives, Digit. Commun. Netw. (2024).
[18] O.S. Althobaiti, M. Dohler, Cybersecurity challenges associated with the Internet of Things in a post-quantum world, Ieee Access 8 (2020) 157356–157381.
[19] A. Lohachab, A. Lohachab, A. Jangra, A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks, Internet Things 9 (2020) 100174.
[20] M. Baldi, P. Santini, G. Cancellieri, Post-quantum cryptography based on codes: State of the art and open challenges, in: 2017 AEIT International Annual Conference, IEEE, 2017, pp. 1–6.
[21] S. Singhal, S. Betgeri, S.K. Singh, et al., Strategies for mitigating security concerns in IoT-enabled smart cities, in: Secure and Intelligent IoT-Enabled Smart Cities, IGI Global, 2024, pp. 239–273.
[22] J.P. Mattsson, B. Smeets, E. Thormarker, Quantum-resistant cryptography, 2021, arXiv preprint arXiv:2112.00399.
[23] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S.X. Ng, L. Hanzo, The evolution of quantum key distribution networks: On the road to the qinternet, IEEE Commun. Surv. Tutor. 24 (2) (2022) 839–894.
[24] C.-H. Huang, A. Mandal, D. Peña-Colaiocco, E.P. Da Silva, V.S. Sathe, Regenerative breaking: Optimal energy recycling for energy minimization in duty-cycled domains, IEEE J. Solid-State Circuits 58 (1) (2022) 68–77.
[25] S. Kumari, M. Singh, R. Singh, H. Tewari, A post-quantum lattice based lightweight authentication and code-based hybrid encryption scheme for IoT devices, Comput. Netw. 217 (2022) 109327.
[26] A. Kumar, Sustainable Security Practices Using Blockchain, Quantum and Post-Quantum Technologies for Real Time Applications, Springer Nature.
[27] K.F. Hasan, L. Simpson, M.A.R. Baee, C. Islam, Z. Rahman, W. Armstrong, P. Gauravaram, M. McKague, A framework for migrating to post-quantum cryptography: Security dependency analysis and case studies, IEEE Access (2024).
[28] I. Ullah, A. Noor, S. Nazir, F. Ali, Y.Y. Ghadi, N. Aslam, Protecting IoT devices from security attacks using effective decision-making strategy of appropriate features, J. Supercomput. 80 (5) (2024) 5870–5899.
[29] Improving source location privacy in social Internet of Things using a hybrid phantom routing technique, Comput. Secur. 123 (2022) 102917.
[30] P. Rupa, S. Singh, S. Arvind, P. Johri, A comprehensive survey on applications of wireless sensor networks and approaches to control congestion, in: ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, Springer, 2020, pp. 805–812.
[31] M.N. Khan, H.U. Rahman, M.Z. Khan, G. Mehmood, A. Sulaiman, A. Shaikh, A. Alqhatani, Energy-efficient dynamic and adaptive state-based scheduling (EDASS) scheme for wireless sensor networks, IEEE Sens. J. 22 (12) (2022) 12386–12403.
[32] S.K. Singh, Y.-S. Jeong, J.H. Park, A deep learning-based IoT-oriented infrastructure for secure smart city, Sustainable Cities Soc. 60 (2020) 102252.
[33] A. Raja Basha, A review on wireless sensor networks: Routing, Wirel. Pers. Commun. 125 (1) (2022) 897–937.
[34] O.A. Amodu, U.A. Bukar, R.A.R. Mahmood, C. Jarray, M. Othman, Age of information minimization in UAV-aided data collection for WSN and IoT applications: A systematic review, J. Netw. Comput. Appl. (2023) 103652.
[35] B. Rana, Y. Singh, Duty-cycling techniques in IoT: Energy-efficiency perspective, in: Recent Innovations in Computing: Proceedings of ICRIC 2021, Vol. 1, Springer, 2022, pp. 505–512.
[36] K. Debasis, L.D. Sharma, V. Bohat, R.S. Bhadoria, An energy-efficient clustering algorithm for maximizing lifetime of wireless sensor networks using machine learning, Mob. Netw. Appl. (2023) 1–15.
[37] Y.L. Cheng, M.H. Lim, K.H. Hui, Impact of Internet of Things paradigm towards energy consumption prediction: A systematic literature review, Sustainable Cities Soc. 78 (2022) 103624.
[38] C. Mangla, S. Rani, N.M.F. Qureshi, A. Singh, Mitigating 5G security challenges for next-gen industry using quantum computing, J. King Saud Univ.-Comput. Inform. Sci. 35 (6) (2023) 101334.
[39] N. Sharma, R. Ketti Ramachandran, The emerging trends of quantum computing towards data security and key management, Arch. Comput. Methods Eng. 28 (7) (2021) 5021–5034.
[40] V. Narayan, A. Daniel, CHHP: Coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network, Int. J. Syst. Assur. Eng. Manag. 13 (Suppl 1) (2022) 546–556.
[41] N. Mahesh, S. Vijayachitra, Hierarchical autoregressive bidirectional least-mean-square algorithm for data aggregation in WSN based IoT network, Adv. Eng. Softw. 173 (2022) 103275.
[42] D.M.C. Milbradt, G.V. Hollweg, P.J.D. de Oliveira Evald, W.B. da Silveira, H.A. Gründling, A robust adaptive one sample ahead preview controller for grid-injected currents of a grid-tied power converter with an LCL filter, Int. J. Electr. Power Energy Syst. 142 (2022) 108286.

[43] T.-C. Wang, M.-H. Shu, Optimum design of generalized adaptive sampling plan for solid supplier-buyer purchasing partnership with yield-driven validation, Expert Syst. Appl. 203 (2022) 117388.
[44] R.H. Yang, J.X. Jin, X.Y. Chen, T.L. Zhang, S. Jiang, M.S. Zhang, Y.Q. Xing, A battery-energy-storage-based DC dynamic voltage restorer for DC renewable power protection, IEEE Trans. Sustain. Energy 13 (3) (2022) 1707–1721.
[45] R. Barzegarkhoo, M. Farhangi, R.P. Aguilera, S.S. Lee, F. Blaabjerg, Y.P. Siwakoti, Common-ground grid-connected five-level transformerless inverter with integrated dynamic voltage boosting feature, IEEE J. Emerg. Sel. Top. Power Electron. 10 (6) (2022) 6661–6672.
[46] H.K. Yugank, R. Sharma, S.H. Gupta, An approach to analyse energy consumption of an IoT system, Int. J. Inf. Technol. 14 (5) (2022) 2549–2558.
[47] J. Shreyas, H. Deepa, P. Udayaprasad, D. Chouhon, N. Srinidhi, D.K. SM, Energy optimization to extend network lifetime for IoT based wireless sensor networks, in: 2022 4th International Conference on Smart Systems and Inventive Technology, ICSSIT, IEEE, 2022, pp. 90–93.
[48] D. Djenour, R. Laidi, Y. Djenouri, Deep learning for estimating sleeping sensor's values in sustainable IoT applications, in: 2022 International Balkan Conference on Communications and Networking, BalkanCom, IEEE, 2022, pp. 147–151.
[49] G. Doumenis, I. Masklavanos, K. Tsiapali, Lightweight operation scheduling for self-powered IoT devices, in: 2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM, IEEE, 2022, pp. 1–7.
[50] G. Singh, P. Joshi, A.S. Raghuvanshi, A novel duty cycle based cross layer model for energy efficient routing in IWSN based IoT application, KSII Trans. Internet Inform. Syst. 16 (6) (2022).
[51] X. Cao, J. Wang, Y. Cheng, J. Jin, Optimal sleep scheduling for energy-efficient AoI optimization in industrial Internet of Things, IEEE Internet Things J. (2023).
[52] J.C. Giacomin, T. Heimfarth, PROA: Pipelined receiver oriented anycast MAC for IoT, in: International Conference on Advanced Information Networking and Applications, Springer, 2022, pp. 68–80.
[53] A. Dhandapani, P. Venkateswari, T. Sivakumar, C. Ramesh, P. Vanitha, Cooperative self-scheduling routing protocol based IOT communication for improving life time duty cycled energy efficient protocol in SDN controlled embedded network, Meas.: Sens. 24 (2022) 100475.
[54] F. Afroz, R. Braun, Empirical analysis of extended QX-MAC for IOT-based WSNS, Electronics 11 (16) (2022) 2543.
[55] K. Gaiova, M. Prauzek, J. Konecny, M. Borova, A concept for a cloud-driven controller for wireless sensors in IoT devices, IFAC-PapersOnLine 55 (4) (2022) 254–259.
[56] R. Rashad, S. Sudhir, Load balancing technique based on network segmentation and adaptive sleep scheduling for 5G-IoT networks, 2022.
[57] A. Ciuffoletti, Deep-sleep for stateful IoT edge devices, Information 13 (3) (2022) 156.
[58] H. Wang, W. Liu, N.N. Xiong, S. Zhang, T. Wang, LIAA: A listen interval adaptive adjustment scheme for green communication in event-sparse IoT systems, Inform. Sci. 584 (2022) 235–268.
[59] A. Chawla, N. Kannan, S. Goyalia, V. Ramanna, J. Sheth, B. Dezfouli, SEMFI: A software-based and real-time energy monitoring platform for WiFi IoT devices, in: 2022 IEEE Global Humanitarian Technology Conference, GHTC, IEEE, 2022, pp. 212–218.
[60] C. Benrebbouh, L. Louail, S. Cherbal, Distributed TDMA for IoT using a dynamic slot assignment, in: International Conference on Advances in Computing Research, Springer, 2023, pp. 469–480.
[61] T. Bhowmik, R. Mojumder, D. Ghosh, I. Banerjee, Efficient scheduling algorithm based on duty-cycle for e-health monitoring system, in: International Conference on Computational Intelligence in Pattern Recognition, Springer, 2022, pp. 211–220.
[62] M. Mohamadi, B. Djamaa, M.R. Senouci, RAST: Rapid and energy-efficient network formation in TSCH-based industrial Internet of Things, Comput. Commun. 183 (2022) 1–18.
[63] M.S. Peelam, A.A. Rout, V. Chamola, Quantum computing applications for Internet of Things, IET Quantum Commun. 5 (2) (2024) 103–112.
[64] R.M. Abd El-Aziz, A.I. Taloba, F.A. Alghamdi, Quantum computing optimization technique for iot platform using modified deep residual approach, Alex. Eng. J. 61 (12) (2022) 12497–12509.
[65] A.A. Abd El-Latif, B. Abd-El-Atty, I. Mehmood, K. Muhammad, S.E. Venegas-Andraca, J. Peng, Quantum-inspired blockchain-based cybersecurity: Securing smart edge utilities in IoT-based smart cities, Inf. Process. Manage. 58 (4) (2021) 102549.
[66] M. Bey, P. Kuila, B.B. Naik, S. Ghosh, Quantum-inspired particle swarm optimization for efficient IoT service placement in edge computing systems, Expert Syst. Appl. 236 (2024) 121270.
[67] K. Shahbazi, S.-B. Ko, Area and power efficient post-quantum cryptosystem for IoT resource-constrained devices, Microprocess. Microsyst. 84 (2021) 104280.
[68] F. Shahid, A. Khan, G. Jeon, Post-quantum distributed ledger for Internet of Things, Comput. Electr. Eng. 83 (2020) 106581.
[69] A.A. Abd EL-Latif, B. Abd-El-Atty, E.M. Abou-Nassar, S.E. Venegas-Andraca, Controlled alternate quantum walks based privacy preserving healthcare images in Internet of Things, Opt. Laser Technol. 124 (2020) 105942.
[70] L. Chen, D.-G. Zhang, J. Zhang, T. Zhang, W.-J. Wang, Y.-H. Cao, A novel offloading approach of IoT user perception task based on quantum behavior particle swarm optimization, Future Gener. Comput. Syst. 141 (2023) 577–594.
[71] M.A. Akbar, A.A. Khan, S. Hyrynsalmi, Role of quantum computing in shaping the future of 6 G technology, Inf. Softw. Technol. 170 (2024) 107454.
[72] A.E. Azzaoui, P.K. Sharma, J.H. Park, Blockchain-based delegated quantum cloud architecture for medical big data security, J. Netw. Comput. Appl. 198 (2022) 103304.
[73] S. Choi, The design of CC2420-based laboratory apparatus monitoring system, in: International Conference on Circuits, Control, Communication, Electricity, Electronics, Energy, System, Signal and Simulation, Springer, 2011, pp. 289–295.
[74] G.F. Riley, T.R. Henderson, The ns-3 network simulator, in: Modeling and Tools for Network Simulation, Springer, 2010, pp. 15–34.
[75] T.R. Henderson, M. Lacage, G.F. Riley, C. Dowell, J. Kopena, Network simulations with the ns-3 simulator, SIGCOMM Demonstr. 14 (14) (2008) 527.
[76] A.A. Al-Saggaf, T. Sheltami, H. Alkhzaimi, G. Ahmed, Lightweight two-factor-based user authentication protocol for iot-enabled healthcare ecosystem in quantum computing, Arab. J. Sci. Eng. 48 (2) (2023) 2347–2357.
[77] Z. Qu, W. Shi, B. Liu, D. Gupta, P. Tiwari, IoMT-based smart healthcare detection system driven by quantum blockchain and quantum neural network, IEEE J. Biomed. Health Inform. (2023).
[78] G. Krishna, A.K. Saha, Optimal sensor spacing in IoT network based on quantum computing technology, Int. J. Parallel Emergent Distrib. Syst. 38 (1) (2023) 58–84.