**ENHANCED Q-LEARNING DQMAC PROTOCOL FOR IOT NETWORKS**

*A Thesis*

*Submitted in partial fulfillment of the requirements for the award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

BY

**ABHISHEK KUMAR (BTECH/15094/20)**

**SHANTI KUMARI (BTECH/15120/20)**

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

BIRLA INSTITUTE OF TECHNOLOGY, MESRA, PATNA CAMPUS, BIHAR-800014, INDIA

# APPROVAL OF THE GUIDE

Recommended that the thesis entitled **“Enhanced Q-Learning based DQMAC protocol for IOT networks”** presented by **Abhishek Kumar and Shanti Kumari** under my supervision and guidance be accepted as fulfilling this part of the requirements for the award of Degree of **Bachelor of Technology in Electronics & Communication Engineering.** To the best of my knowledge, the content of this thesis did not form a basis for the award of any previous degree to anyone else.

Date: ____________

_____________________


Dr.Atul Kumar Pandey

Assistant Professor

Dept.of Electronics & Communication Engineering

Birla Institute of Technology Mesra, Patna

# DECLARATION CERTIFICATE

I certify that

a) The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.

b) The work has not been submitted to any other Institute for any other degree or diploma.

c) I have followed the guidelines provided by the Institute in writing the thesis.

d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e) Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f) Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

ABHISHEK KUMAR (BTECH/15094/20)

SHANTI KUMARI (BTECH/15120/20)

# CERTIFICATE OF APPROVAL

This is to certify that the work embodied in this thesis entitled **“Enhanced Q-Learning based DQMAC Protocol for IOT networks”**, is carried out by **Abhishek Kumar (BTECH/15094/20) and Shanti Kumari (BTECH/15120/20)** has been approved for the degree of Bachelor of Technology in Electronics & Communication Engineering at Birla Institute of Technology, Mesra, Patna.

Date:

Place:

**Internal Examiner** **External Examiner**

**Head of Department**

# ABSTRACT

The Enhanced Q-Learning-based Distributed Queueing (EQL-DQ) algorithm represents a novel approach to Medium Access Control (MAC) protocol design tailored specifically for Internet-of-Things (IoT) networks. In this project, we present a comprehensive overview of the EQL-DQ algorithm, detailing its core principles, design considerations, and performance evaluation.

The project begins by introducing the motivation behind the development of EQL-DQ, highlighting the growing importance of efficient MAC protocols in IoT environments characterized by diverse traffic types, dynamic network conditions, and stringent energy constraints. We outline the key challenges faced by existing MAC protocols and articulate the need for a more adaptive, high throughput, low latency and energy-efficient approach.

Next, we delve into the core components of the EQL-DQ algorithm, starting with an explanation of its reinforcement learning-based adaptive adjustment mechanism for the number of contending nodes. We describe how EQL-DQ dynamically optimizes contention parameters to balance throughput, delay, and energy consumption, ensuring efficient resource utilization and improved network performance.

Furthermore, we elucidate the distributed queueing mechanism employed by EQL-DQ to handle collisions and parallel transmissions of data . By leveraging distributed queues, EQL-DQ optimizes the allocation of transmission opportunities, minimizing contention and enhancing overall system throughput and fairness.

The project also highlights the parallel execution of contention slots as a key feature of EQL-DQ, enabling efficient utilization of computational resources and enhancing simulation performance. We discuss the rationale behind parallel execution and its implications for scalability and simulation accuracy in evaluations.

To empirically evaluate the performance of EQL-DQ, extensive simulations were conducted, comparing its performance against existing Q-learning-based MAC protocols and traditional approaches. Throughput, average contentions, average delay, and energy consumption metrics were analyzed to demonstrate the superiority of EQL-DQ in various network scenarios.

# ACKNOWLEDGEMENT

An endeavour over a long period can be successful only with the advice and support of many well-wishers. The task would be incomplete without mentioning the people who have made it possible, because is the epitome of hard work. So, with gratitude, we acknowledge all those whose guidance and encouragement owned our efforts with success.

We confirm that this project report is the original and independent work of Abhishek Kumar and Shanti Kumari. It has not been submitted in part or whole for any other degree or academic purpose. All the information and data presented in this report are genuine and have been obtained from reliable, duly acknowledged sources.

We wish to express grateful acknowledgment to our guide Dr. Atul Kumar Pandey**,** B.Tech., M.E., Ph.D., Assistant Professor, Department of Electronics & Communications Engineering, BIT PATNA for his inspiring guidance and continuous encouragement throughout the project.

Finally, we would like to extend our deep sense of gratitude to all the friends, and last but not least we are greatly indebted to our parents who inspired us under at all circumstances.

Thank you.

DATE:

Abhishek Kumar (BTECH/15094/20)
Shanti Kumari (BTECH/15120/20)

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLE

# CHAPTER-1

# INTRODUCTION

## 1.1. Introduction to IOT Network Protocols:

The concept of the Internet of Things (IoT) revolves around the connectivity of sensor devices to the internet in real-time. These devices interact with each other through networks, necessitating the establishment of standards and regulations governing data exchange. These regulations are known as IoT Network Protocols. Presently, there exists a diverse array of IoT devices, each requiring distinct protocols.

The functionality of an IoT application determines its workflow or architecture, which typically comprises four layers: the Sensing layer, Network layer, Data processing layer, and Application layer.

The Sensing layer encompasses all hardware components such as sensors, actuators, and chips responsible for gathering data. This layer interfaces with the subsequent layer, the network layer, through specific protocols. In the Network layer, devices communicate via various network protocols such as cellular, Wi-Fi, Bluetooth, Zigbee, etc.

Data collected by IoT devices undergo processing in the Data processing layer, utilizing technologies like data analytics and machine learning algorithms. The processed data is then presented to users through web portals, applications, or interfaces provided by the Application layer, enabling direct interaction and visualization of IoT data.

Designing protocols for IoT poses challenges due to the limited components of IoT devices, primarily comprising small batteries and sensors. Additionally, all operations, including constructing topological structures and address assignments, must be performed wirelessly, adding to the complexity.

### 1.1.1 IoT protocols must fulfill several critical requirements:

1. Simultaneous Communication:- They should facilitate communication among multiple devices concurrently to support the interconnected nature of IoT systems.

2. Communication Security:-Given the deployment of IoT in sensitive domains like healthcare, industries, and home surveillance, protocols must ensure robust security measures to protect data and privacy.

3. Efficient Data Transport:- Efficient data transmission is essential for optimizing resource utilization and ensuring timely delivery of information across the IoT network.

4.Scalability:- Protocols should be scalable to accommodate the dynamic nature of IoT networks, allowing for the seamless addition or removal of devices without compromising performance or reliability.

When selecting a protocol, it's crucial to consider the specific environment and requirements it's designed for. Protocols vary in terms of range, data rates, power consumption, and other factors.

For short-range communication with low data rates and low power consumption, Bluetooth is a viable option. Operating in the 2.4GHz frequency range, Bluetooth offers a range of 10m to 100m and supports data rates of up to 1Mbps. It accommodates two network topologies: point-to-point and mesh.

Bluetooth is well-suited for transmitting small amounts of data to personal devices such as speakers, earphones, smartwatches, and smart shoes. Additionally, it can be employed in Smart Home applications, including alarms, HVAC systems, and lighting control.

Zigbee, based on the IEEE802.15.4 standard, operates within the 2.4GHz frequency range, similar to Bluetooth. It offers a range of up to 100 meters with a maximum data rate of 250KBPS. Zigbee is ideal for transmitting small amounts of data over short distances and is particularly suitable for applications requiring high authentication and robustness. It supports multiple network topologies, including star, mesh, and cluster tree.

Common applications of Zigbee include monitoring device health in industrial settings and facilitating smart home solutions.

6LoWPAN, short for IPv6 Low Power Personal Area Network, operates within the frequency range of 900 to 2400MHz, providing a data rate of 250KBPS. It supports star and mesh network topologies and is optimized for low-power consumption in personal area network environments.

For short-range communication with high data rates, Wi-Fi (Wireless LAN) is an excellent choice. Wi-Fi offers high bandwidth, supporting data rates ranging from 54Mbps up to 600Mbps. It typically covers a range of 50m in a local area, but with private antennas, it can extend up to 30km. Wi-Fi enables easy connectivity of IoT devices, allowing them to share large amounts of data. It finds applications in smart homes, smart cities, offices, and various other settings.

For long-range communication with high data rates and low power consumption, two notable options are LoRaWAN and LTE-M:

LoRaWAN (Long Range Wide Area Network) provides extensive coverage, with a range of approximately 2.5km to 15km. Although its data rate is relatively low, ranging from 0.3KBPS to a maximum of 50KBPS, it can support numerous connected devices. LoRaWAN is utilized in applications such as Smart City deployments and Supply Chain Management.

LTE-M (Long Term Evolution for Machines) is a type of LPWAN (Low Power Wide Area Network) that operates within a frequency range of 1.4MHz to 5MHz. It offers data rates of up to 4MBPS and is often used in conjunction with cellular networks to provide enhanced security features. LTE-M is suitable for various IoT applications where reliable, high-speed data transmission is required.

For long-range communication with low data rates and low power consumption, Sigfox is a viable option. Sigfox is designed for wide area coverage with minimal power consumption, aiming to connect billions of IoT devices. Operating at a frequency range of 900MHz, Sigfox offers a range spanning from 3km to 50km, with a maximum data rate of 1KBPS.

Alternatively, for long-range communication with low data rates but high power consumption, cellular networks, including 2G, 3G, 4G, and 5G, are utilized. Cellular networks operate across various frequency ranges such as 900MHz and 1.8/1.9/2.1 GHz, providing a range of approximately 35km to 200km. The average data rates range from 35KBPS to 170KBPS. However, cellular networks consume relatively high power, which may not be suitable for all IoT devices. They are commonly employed in IoT applications like connected cars, where the benefits of widespread coverage and reliable communication outweigh the power consumption concerns.

Now, let's delve into the intricacies of MAC protocols and their significance in the realm of IoT connectivity and communication.

## 1.2. Introduction of MAC Protocol:

The MAC sub-layer constitutes an integral component within the data link layer protocol. It furnishes a mechanism for enabling multiple devices to access the channel when sharing the medium. In a wireless environment where multiple devices share the medium and communication is broadcasted, any transmission from one device is received by all other devices within its transmission range. This scenario may result in interference and collisions of frames when simultaneous transmissions from two or more devices occur at a single point. In a scattered, dense, and rugged sensor field, sensor nodes typically communicate through multi-hop paths over the wireless medium. A MAC protocol governs the flow of communication over a common medium, establishing a foundational network structure for sensor nodes to interact. Hence, it bestows nodes with self-organizing capabilities and endeavors to maintain network unity by facilitating collision-free and errorless communication between senders and receivers. Furthermore, the standard prerequisites for extending the lifespan of a WSN without necessitating power replacement or human intervention have stimulated the creation of innovative protocols across all layers of the communication stack. Yet, significant benefits can be attained within the data link layer, where the MAC protocol directly manages radio operations, the most power-intensive aspect of resource-limited sensor nodes. Effective MAC protocols employ the radio resource wisely to preserve its energy. Therefore, the MAC protocol contributes significantly to meeting key design goals of WSNs by delineating how nodes utilize radio, manage channel sharing, prevent collisions in correlated and broadcasting environments, promptly respond to inquiries, and prolong their operational lifespan.

The IEEE 802 model provides details about the MAC (Media Access Control) layer protocols, which are specifically designed for local area networks (LANs). The MAC layer is responsible for controlling how devices access the shared physical medium (e.g., Ethernet cable, wireless channel) to transmit data. In the IEEE 802 model, the MAC layer is positioned above the Physical layer and below the LLC (Logical Link Control) layer.

Three prominent MAC layer protocols as per the IEEE 802 model are:

1. CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

- This is the MAC protocol used in traditional Ethernet networks.
- CSMA/CD allows multiple devices to share the same physical medium by listening for traffic before transmitting (Carrier Sense).
- If two devices transmit simultaneously, a collision occurs, which is detected (Collision Detection), and the devices retransmit after a random backoff period.

2. Token Bus:

- This is a MAC protocol used in token-bus networks, such as IEEE 802.4.
- In a token-bus network, devices are connected to a shared bus, and a small "token" frame is passed around to grant transmission rights.
- A device can transmit data only when it holds the token, preventing collisions.

3. Token Ring:

- This is a MAC protocol used in token-ring networks, such as IEEE 802.5.
- Devices are connected in a ring topology, and a small "token" frame circulates around the ring.
- A device can transmit data only when it holds the token, ensuring only one device transmits at a time.

## 1.3. Classification of WSN MAC Protocols:

Several MAC protocols have been successfully proposed to meet the stringent design requirements of WSNs. Actually; these protocols depend on how protocol allows nodes to access the channel. We have classified WSN based MAC protocol as depicted in Figurure( "The figures used in this section were referenced from [34]." )into four categories as; contention based, scheduling based, channel polling based, and hybrid protocols.

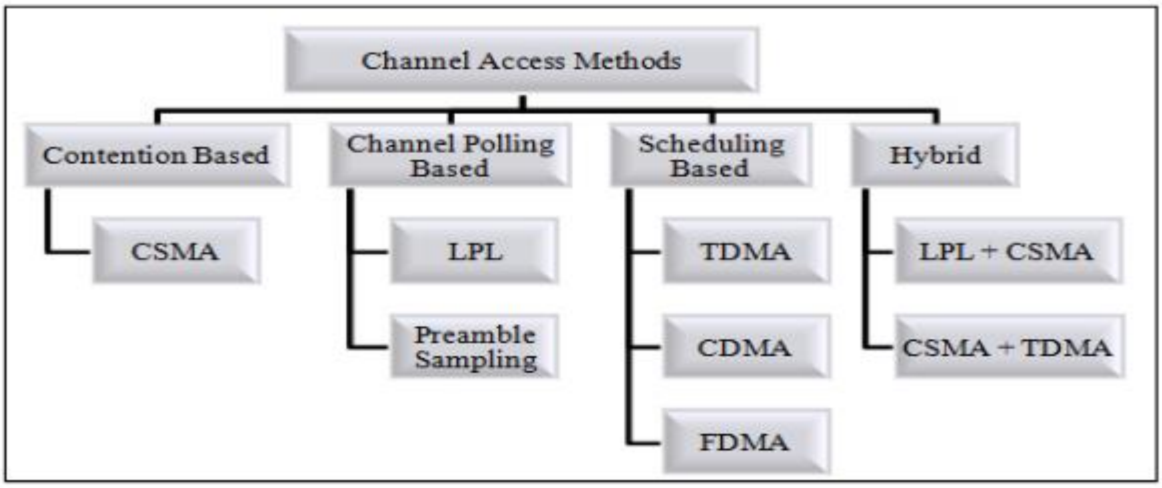


Fig.1.1. Channel Accessing taxonomy in wsn

## 1. Contention Based MAC Protocols:

Contention-based MAC protocols indeed play a crucial role in wireless sensor networks (WSNs) due to their simplicity, flexibility, and suitability for event-driven applications. Here's a breakdown of the key points mentioned in the excerpt:

i. **Acquiring Channel Access**:-Nodes in contention-based MAC protocols compete with their neighbors to access the channel for transmission. Before starting data transmission, a node senses the carrier to check if the channel is idle. If the channel is idle, the node starts transmission; otherwise, it defers transmission for a random period using a backoff algorithm.

ii. **Reduced Resource Consumption**:-Contention-based MAC protocols are preferred for event-driven WSN applications because they reduce processing resource consumption. These protocols do not require clustering or topology information, allowing each node to independently contend for channel access without coordinating frame exchanges.

iii. **Flexibility and Scalability**:- Contention-based MAC protocols are flexible and dynamic, making them suitable for networks of varying scales. They do not rely on centralized coordination and can adapt to changes in network conditions.

iv. **Challenges and Issues**:-Despite their benefits, contention-based MAC protocols face challenges such as hidden and exposed terminals, which can lead to collisions, overhearing, idle listening, and reduced throughput. These issues arise when nodes contend for channel access independently, leading to inefficient utilization of the channel and decreased network performance.

v. **Synchronized Contention Times**:- Some MAC protocols synchronize the contention times of nodes according to a schedule. At each periodic interval, neighboring nodes wake up simultaneously to exchange packets. This synchronization helps in reducing collisions and improving overall network efficiency.

## 2. Channel Polling Based MAC Protocols:

The channel polling scheme, also known as preamble sampling or Low Power Listening (LPL), involves sending preamble data packets with extra bytes by nodes. The preamble serves to ensure that the destination node detects radio activity and wakes up before receiving the actual payload from the sender. Here's a breakdown of how the channel polling scheme works:

i. **Preamble Transmission**:- Nodes send a preamble over the channel to alert potential receivers of radio activity. The preamble is transmitted before the actual data payload, allowing receivers to wake up and listen for incoming packets.

ii. **Receiver Wake-Up**:- Upon detecting radio activity from the preamble, the receiver node wakes up its radio to receive data packets. If no radio activity is detected, the receiver node goes back to sleep mode until the next polling interval.

iii. **Check Interval Duration**:- The receiver periodically checks for radio activity during the check interval duration, which lasts until the preamble is sent. This ensures that the receiver remains awake long enough to detect incoming packets.

iv. **Synchronization and Clustering**:- Unlike other MAC protocols that rely on synchronized active/sleep schedules, channel polling schemes do not require synchronization, scheduling, or clustering among nodes. Each node independently wakes up to listen for incoming transmissions based on the presence of preambles.

v. **Combination with ALOHA**:- The combination of ALOHA with preamble sampling is a common example of the extended preamble-based channel polling scheme. This approach helps in reducing energy consumption by allowing nodes to sleep when no radio activity is detected.

vi. **Berkeley MAC (BMAC) Protocol**:- The Berkeley MAC protocol (BMAC) is an example of a channel polling scheme where the channel polling method is referred to as Low Power Listening (LPL). BMAC utilizes extended preambles to enable low-power listening and efficient communication in WSNs.

## 3. Scheduling Based MAC Protocols:

Scheduling-based MAC schemes allocate collision-free links between neighboring nodes during the initialization phase. These schemes divide the system time into slots and assign them to nodes, controlling participant authorization on resources with regular timing. Here's a breakdown of how scheduling-based MAC schemes work and their advantages and challenges:

i. **Time Division Multiplexing (TDMA)**:- TDMA schemes divide system time into slots and allocate them to neighboring nodes. The schedule, regulated by a central authority, controls resource access. Nodes do not contend with neighbors and can only access their allocated time slots. TDMA offers advantages such as minimum collisions, reduced overhearing, avoidance of idle listening, and predictable end-to-end delay.

ii. **Advantages**:-
- Minimum collisions and reduced overhearing.
- Predictable end-to-end delay.
- Bounded and predictable queuing delay.

iii. **Challenges**:-
- High average queuing delay due to waiting for allocated time slots.
- Overhead and extra traffic.
- Lack of adaptability.
- Reduced scalability.

- Difficulty in allocating conflict-free TDMA schedules.
- Inability for peer-to-peer communication without involving the central authority.

iv. **Representative Protocols**:-

- **DRAND**:- A distributed slot assignment scheme that overcomes the difficulty of obtaining global topology information.
- **PACT**:- Another distributed slot assignment scheme aimed at large networks.
- **TRAMA**:- A protocol that obtains local topology and interference information at each node.
- **Flow-Aware Medium Access (FLAMA)**:- Derived from TRAMA, optimized for periodic monitoring applications to avoid overhead.
- **Low-Complexity Slot Selection Mechanisms**:- Simplified schemes like LMAC aim to achieve good energy efficiency by reducing radio state transitions and protocol overhead.
- **Adaptive Information-centric LMAC (AILMAC**):- AILMAC tailors slot assignment to actual traffic needs, addressing the fixed frame length issue of LMAC.
- **Joint Interconnect Protocol**:- Addresses maximum throughput and fair rate allocation in WSNs with consideration for slot reuse-based TDMA.

## 4. Hybrid MAC Protocol:

Hybrid MAC protocols offer a promising approach to leverage the advantages of different MAC schemes while mitigating their individual weaknesses. By combining synchronized schemes, such as TDMA (Time Division Multiple Access), with asynchronous schemes, like CSMA (Carrier Sense Multiple Access), hybrid protocols aim to achieve a balance between efficiency, scalability, and complexity.

One notable example of a hybrid MAC protocol is the Zebra MAC (Z-MAC) protocol. Z-MAC combines the strengths of TDMA and CSMA, leveraging the deterministic access provided by TDMA for predictable communication scheduling while also benefiting from the flexibility and adaptability of CSMA. By dynamically switching between TDMA and CSMA modes based on network conditions, Z-MAC optimizes both energy efficiency and throughput.

Another example is the Scheduled Channel Polling MAC (SCP-MAC), which integrates scheduled channel polling with contention-based access methods. SCP-MAC allows nodes to operate in a synchronized manner, similar to TDMA, while also supporting contention-based access during periods of low traffic. This hybrid approach enables SCP-MAC to achieve low-latency communication while maintaining scalability and adaptability.

Similarly, the Funneling-MAC protocol combines TDMA-based scheduling with funneling techniques to improve channel utilization and reduce contention overhead. By organizing nodes into hierarchical structures and employing time-slotted channel access, Funneling-MAC optimizes resource allocation and minimizes collisions in large-scale networks.

While hybrid MAC protocols offer the potential for significant performance improvements, they may also introduce challenges related to scalability and complexity. Managing multiple working modes and coordinating the interaction between synchronized and asynchronous schemes can increase overhead and complicate protocol design. However, with careful design and optimization, hybrid MAC protocols can effectively address the diverse requirements of IoT applications and improve overall network performance.

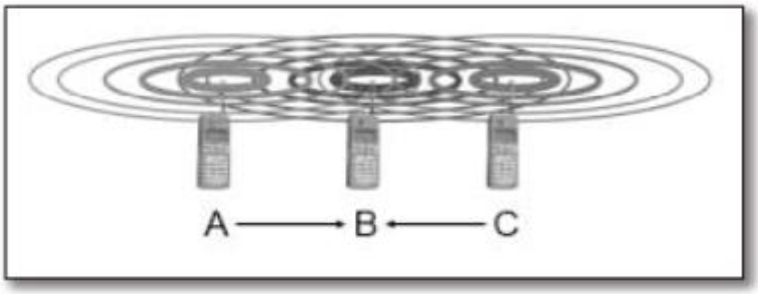


Fig.1.2. Hidden terminal

The hidden terminal problem occurs when two nodes, such as nodes A and C in the example, are within range of a common receiver, node B, but are out of range of each other. In this scenario, node C may mistakenly sense the medium as idle when node A is transmitting to node B, leading to a collision at node B because node A cannot detect the transmission from node C. This problem arises due to the inability of nodes to detect transmissions from hidden nodes, resulting in collisions at the receiver.

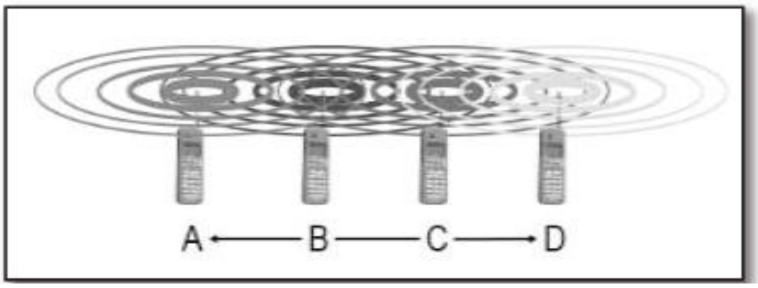

Fig.1.3. Exposed terminal

On the other hand, the exposed terminal problem occurs when a node, such as node C in the example, refrains from transmitting even though the medium is idle because it mistakenly assumes that the transmission from node B to node A will interfere with its transmission to node D. In reality, there would be no interference, and node C unnecessarily delays its transmission, leading to decreased network efficiency. This problem arises due to nodes being overly cautious about potential interference from transmissions outside their range.

Both the hidden terminal problem and the exposed terminal problem can significantly impact the performance and reliability of a wireless network, leading to increased collisions, reduced throughput, and inefficient use of the available bandwidth. Effective MAC protocols should aim to mitigate these problems through techniques such as carrier sensing, clear channel assessment, and efficient channel access mechanisms.

## 1.4. Characteristics of MAC Protocols:

The prevalent attributes of MAC protocols can be encapsulated as follows:

**Throughput**:-Throughput refers to the speed at which messages are processed. It can be quantified in messages or symbols per second, although the predominant unit of measurement is bits per second. The objective is to optimize this rate.

**Transmission delay**:- Transmission delay refers to the duration during which a solitary message remains within the MAC protocol.

**Fairness**:- Fairness in a MAC protocol is defined by its ability to distribute the channel among competing nodes based on predetermined fairness criteria.

**Scalability**:-Scalability refers to the capability of a communication system to maintain its performance standards regardless of the network's size and the number of nodes vying for resources.

**Robustness**:-Robustness encompasses reliability, availability, and dependability in its composition.

**Stability**:-Stability refers to the protocol's capability to effectively manage variations in traffic load over an extended duration.

## 1.5. Goals of MAC Protocols:

- **Avoid Overhearing**: Refrain from needlessly receiving packets meant for other nodes.
- **Reduce Idle Listening**: Avoid remaining active to receive data when there is no sender.
- **Decrease Active Time**: Minimize the duration of active listening.
- **Prevent Packet Collisions**: Take measures to eliminate collisions between packets.
- **Reduce Control Packet Overhead**: Minimize the amount of overhead caused by control packets.
- **Avoid Buffer Overflow**: Prevent overflow in the buffer to maintain efficient operation.

## 1.6. IEEE 802.15.4 MAC

The IEEE introduced the 802.15.4 MAC standard for Wireless Personal Area Networks (WPANs), incorporating a duty cycle mechanism that allows for adjustable sizes of active and inactive periods during PAN formation.

**Network Architecture and Types/Roles of Nodes:**

The IEEE 802.15.4 MAC combines both the schedule-based and contention-based protocols and supports two network topologies, star and peer-to-peer as shown in Figure

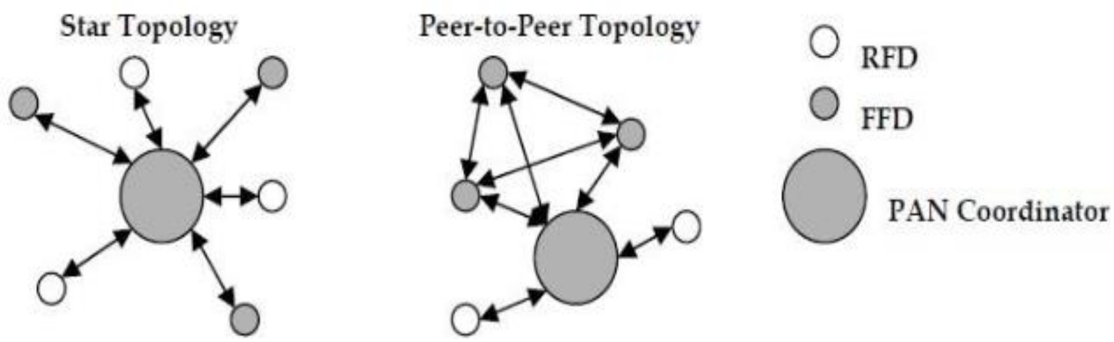


Fig.1.4. Topology configurations supported by IEEE 802.15.4

### 1.6.1. Applications of IEEE 802.15.4

1. Wireless Sensor Networks
2. Home Automation Systems
3. Home Networking Solutions
4. Interconnecting Devices with PCs
5. Home Security Systems
6. Industrial Automation

7. Smart Grids for Energy Management
8. Healthcare Monitoring Systems
9. Environmental Monitoring and Control
10. Asset Tracking and Management

There are two distinct peer-to-peer topology variants. The first, termed a cluster-tree network, has found widespread application in ZigBee technology. The second variant, referred to as a mesh network, is extensively utilized within the IEEE 802.15 WPAN Task Group 5 (TG5). The standard delineates two node types: the Full Function Device (FFD) and the Reduced Function Device (RFD). FFD nodes possess the flexibility to assume three distinct roles: PAN coordinator, coordinator, and device. Conversely, RFDs are limited to operating solely as devices. Under all network conditions, devices are required to be associated with a coordinator. Multiple coordinators can function within either a peer-to-peer or star topology, with one coordinator assuming the role of the PAN coordinator. The star topology is better suited for delay-critical applications and small network coverage due to its efficient communication structure. In contrast, the peer-to-peer topology is more applicable for large networks with multi-hop requirements, albeit at the expense of higher network latency. Moreover, the standard delineates two modes for data exchange: beacon mode and non-beacon mode. Beacon mode offers synchronization measures to networks, whereas non-beacon mode provides asynchronous features.

### 1.6.2. Superframe Structure:

The beacon mode of IEEE 802.15.4 MAC introduces a superframe structure to organize channel access and data exchanges. The structure figure 1.6, comprises two primary periods: the active period and the inactive period. The active period is subdivided into 16 time slots, typically beginning with the transmission of the beacon frame in the first slot. Following this, two additional segments, the Contention Access Period (CAP) and the Contention-Free Period (CFP), utilize the remaining slots. The CFP, also referred to as Guaranteed Time Slots (GTS), can employ up to 7 time slots.

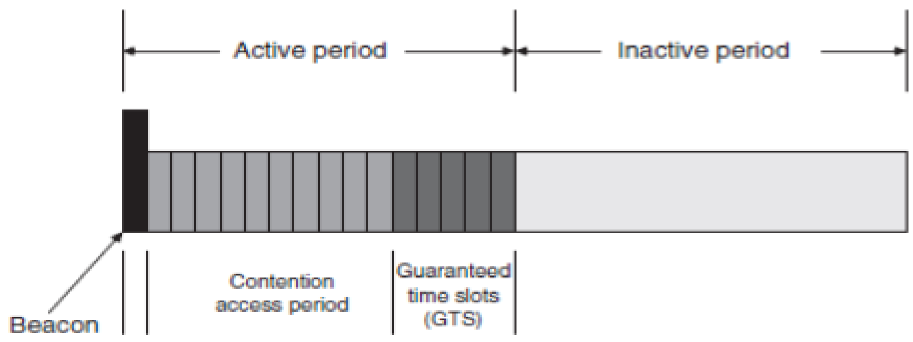

Fig.1.5. Superframe structure of IEEE 802.15.4

The duration of both the active and inactive periods, as well as the length of an individual time slot, are configurable and vary based on traffic conditions. Data transmissions can take place either in the Contention Access Period (CAP) or in Guaranteed Time Slots (GTS). In CAP, data communication employs slotted CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance), while in GTS, nodes are allocated fixed time slots dedicated to data communication. To achieve energy-efficient operations in IEEE 802.15.4 MAC, the strategy involves putting nodes to sleep during the inactive period, as well as when there is neither data to transmit nor any data to fetch from the coordinator. Nevertheless, the energy burden falls primarily on the coordinator, which must remain active throughout the entire active period.

### 1.6.3. GTS Management:

The coordinator assigns Guaranteed Time Slots (GTS) to devices only upon receiving appropriate request packets during the Contention Access Period (CAP). A flag within the request indicates whether the requested time slot is for transmission or reception.

In a transmit slot, the device sends packets to the coordinator, while in a receive slot, data flows in the opposite direction. Another field in the request specifies the desired number of contiguous time slots in the GTS phase.

The coordinator responds to the request packet in two steps: Firstly, an immediate acknowledgment packet confirms the receipt of the request packet by the coordinator, but contains no information regarding the success or failure of the request.

Upon receiving the acknowledgment packet, the device must monitor the coordinator's beacons for a specified duration known as a GTSDescPersistenceTime. When the coordinator has adequate resources to allocate a GTS to the node, it inserts an appropriate GTS descriptor into one of the subsequent beacon frames. This GTS descriptor includes the short address of the requesting node and details about the number and position of the time slots within the GTS phase of the superframe.

A device can utilize its allocated slots each time they are announced by the coordinator in the GTS descriptor. If the coordinator lacks sufficient resources, it generates a GTS descriptor for an (invalid) time slot zero, indicating the available resources in the descriptor's length field. Upon receiving such a descriptor, the device may consider renegotiation. If the device does not receive a GTS descriptor within the specified GTSDescPersistenceTime after sending the request, it assumes that the allocation request has failed. A GTS is allocated to a device regularly until it is explicitly deallocated. The device can request deallocation by sending a special control frame. Upon sending this frame, the device should refrain from using the allocated slots further.

Alternatively, the coordinator can trigger deallocation based on certain criteria. Specifically, the coordinator monitors the usage of the time slot. If the slot remains unused for a certain number of superframes, it is deallocated. The coordinator signals deallocation to the device by generating a GTS descriptor with start slot zero.

### 1.6.4. Data Transfer:

When a device intends to transmit a data packet to the coordinator and possesses an allocated transmit GTS, it awakens just before the time slot initiation and promptly sends its packet without conducting carrier-sense or other collision-avoidance procedures. However, this action is feasible only if the entire transaction, including the data packet, an immediate acknowledgment from the coordinator, and appropriate InterFrame Spaces (IFSs), can fit within the allocated time slots.

When a device needs to transmit a data packet to the coordinator and has an allocated transmit GTS, it sends the packet without carrier-sensing operations if the entire transaction, including the packet, coordinator's immediate acknowledgment, and InterFrame Spaces (IFSs), fits within the allocated time slots. If not or if the device lacks allocated slots, it transmits during the Contention Access Period (CAP) using slotted CSMA.

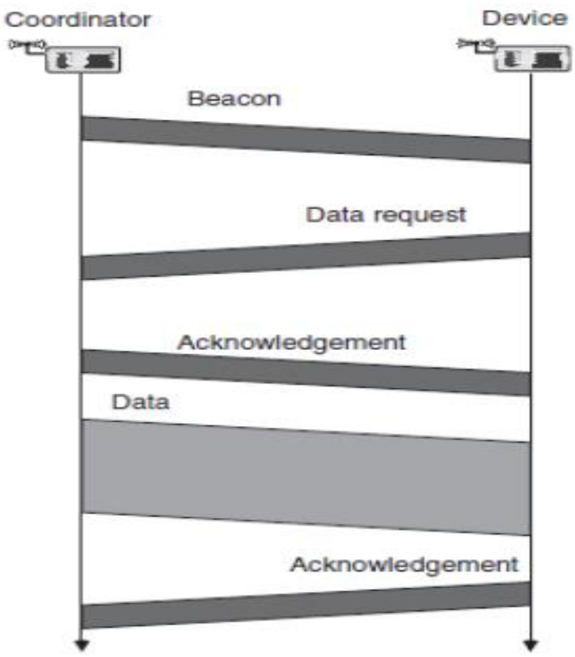

Fig.1.6. Handshake between coordinator and device when the device retrieves a packet

The coordinator acknowledges the packet. Conversely, if the coordinator wants to send a packet to the device and the device has an allocated receive GTS that accommodates the packet cycle,

the coordinator transmits it in the allocated slot without further coordination. The device acknowledges the packet, following a handshake protocol. The coordinator signals a buffered packet by including the device's address in the pending address field of the beacon frame, prompting the device to send a data request packet during CAP. Upon receiving this, the coordinator acknowledges and sends the data packet, with the device preparing for its reception. If unsuccessful, the device retries during subsequent superframes, optionally powering off its transceiver until the next beacon.

### 1.6.5. Slotted CSMA-CA Protocol:

When nodes need to transmit data or management/control packets during the Contention Access Period (CAP), they utilize a slotted CSMA protocol. However, this protocol does not incorporate provisions to address hidden-terminal situations. For instance, there is no RTS/CTS handshake implemented. In order to diminish the likelihood of collisions, the protocol integrates random delays, thereby classifying it as a CSMA-CA protocol (Carrier Sense Multiple Access with Collision Avoidance). The time slots within the Contention Access Period (CAP) are further divided into smaller segments known as backoff periods. Each backoff period spans a length equivalent to 20 channel symbol times, and only these backoff periods are taken into consideration by the slotted CSMA-CA protocol.

## 1.7. DQ Mechanism:-

The DQ protocol, pioneered by [17], introduces a novel approach where active devices contend in contention slots before proceeding with data transmission. Central to this protocol is the role of a coordinator, tasked with disseminating crucial feedback information concerning the state of each contention slot. This feedback assists in organizing devices into one of two logical queues: the collision resolution queue (CRQ) and the data transmission queue (DTQ). Upon receiving feedback, devices are strategically placed based on the observed state of contention. If collisions are detected during contention, devices are segmented and queued into the CRQ for subsequent resolution. Conversely, if no collisions occur, devices are queued into the DTQ, awaiting a collision-free transmission opportunity. This dynamic queue management system optimizes network efficiency by mitigating collisions and ensuring smoother data transmission processes.

In the context of [17], the gateway assumes the pivotal role of a coordinator within the DQ protocol framework. All devices operating within this environment are required to operate in class B mode to facilitate the reception of feedback packets at predefined periodic intervals.

This periodic feedback mechanism enables devices to stay synchronized with the coordinator's instructions and adapt their operations accordingly. Furthermore, each superframe is meticulously divided into three distinct segments: the contention window, data slot, and feedback slot. These segments serve specific purposes within the protocol's operation, with the contention window facilitating contention resolution, the data slot enabling data transmission, and the feedback slot facilitating the broadcast of crucial feedback information. This structured division of the superframe ensures efficient utilization of network resources and enhances the overall performance and reliability of the DQ protocol implementation.

The research paper [33] presents PDQRAP (Prioritized Distributed Queueing Random Access Protocol), a prioritized variant of the DQRAP protocol designed to facilitate efficient transport of multimedia traffic over shared mediums such as LANs, MANs, and WANs. PDQRAP introduces two distinct formats: extra-bit and extra-slot, enabling nodes to differentiate between high-priority and normal-priority traffic. This prioritization scheme aims to enhance throughput and reduce delay for high-priority traffic segments, with observed improvements ranging from 20% to 30% under 90% load conditions, experiencing one-third to one-half the delay compared to normal-priority traffic. However, while PDQRAP offers significant benefits, it also poses several potential limitations. These include increased overhead resulting from prioritization mechanisms, concerns regarding fairness and potential starvation of lower-priority traffic, heightened protocol complexity, and scalability challenges as network size or priority levels increase. Despite these considerations, PDQRAP presents a promising approach to effectively accommodate multimedia services with varying priorities over shared network infrastructures.

In research paper [10] addresses the critical issue of sleep/wake-up scheduling, vital for conserving the limited energy resources of sensor nodes. By maximizing network lifetime while ensuring efficient packet delivery, sleep/wake-up scheduling plays a pivotal role in WSNs. The paper explores the use of duty-cycling techniques, where nodes alternate between active and sleep modes based on a predefined schedule. However, this approach presents a trade-off between energy savings and packet delivery delay, as nodes risk missing transmissions during sleep. To mitigate these trade-offs, researchers have proposed self-adaptive sleep/wake-up scheduling algorithms, enabling nodes to autonomously determine their operational mode in each time slot using reinforcement learning techniques. While promising, designing such mechanisms poses challenges, as nodes must balance energy conservation with network connectivity and responsiveness, necessitating sophisticated algorithm design and decision-making processes.

The paper [1] explores the concept of optimizing system throughput in high-traffic network scenarios by employing dormant states for certain nodes, which intermittently awaken to transmit data, potentially reducing collisions and enhancing overall throughput. Introducing the Q-learning-based distributed queuing medium access control (QL-DQMAC) protocol for Internet of Things (IoT) networks, this work aims to determine the optimal number of contending IoT nodes in each contention period. Each node independently calculates its active rate using a Q-learning algorithm and decides whether to be active or enter a dormant state in the next contention period based on this rate. By optimizing the number of active nodes, the QL-DQMAC protocol effectively reduces collision probabilities, leading to lower energy consumption and decreased delays associated with the medium access control (MAC) contention process. Comparative analysis against other DQMAC approaches underscores the superior performance of the QL-DQMAC protocol in IoT network environments.

In the realm of Software-Defined Networking (SDN), the paper [23] delves into the remarkable programmability and extensive capabilities offered by SDN for network management operations. The centralized intelligence of the controller empowers it to adapt routing policies dynamically based on application requirements. To further augment these capabilities, the integration of Artificial Intelligence (AI) tools endows the controller with the ability to autonomously reconfigure the network in response to changing conditions. The paper specifically focuses on deploying a Q-learning algorithm for optimizing routing, with a primary emphasis on minimizing latency. The authors investigate the impact of exploration-exploitation strategies on the algorithm's performance and propose enhancements to optimize its efficacy within the target environment. These enhancements include incorporating a congestion-avoidance mechanism during the exploration phase and introducing a novel strategy based on the Max-Boltzmann Exploration method (MBE), which amalgamates traditional ε-greedy and softmax strategies. The empirical results underscore the superiority of the MBE strategy, coupled with the congestion-avoidance mechanism, demonstrating superior performance in terms of average latency, convergence time, and computational efficiency when compared to ε-greedy, ε-decay, and softmax strategies.

## 1.8. Machine learning:-

Machine learning, with its ability to extract patterns and insights from large datasets, has indeed become a valuable tool in addressing complex problems in communication networks. Here are some key ways in which machine learning is applied in this domain:

1. **Network Optimization**:- Machine learning algorithms can optimize various aspects of communication networks, such as routing, resource allocation, and spectrum management. By analyzing network traffic patterns and performance metrics, machine learning models can dynamically adjust network parameters to maximize throughput, minimize latency, and optimize resource utilization.

2. **Anomaly Detection and Security**:- Machine learning techniques are used to detect anomalies and security threats in communication networks. By learning patterns of normal network behavior, machine learning models can identify deviations indicative of malicious activities, such as network intrusions, denial-of-service attacks, or anomalous traffic patterns.

3. **Quality of Service (QoS) Improvement**:- Machine learning algorithms can enhance the quality of service in communication networks by predicting network congestion, optimizing traffic routing, and prioritizing critical data packets. By analyzing historical network performance data, machine learning models can dynamically adjust network configurations to meet QoS requirements and ensure reliable and efficient data transmission.
4. **Resource Management and Allocation**:- Machine learning is applied to optimize the allocation of network resources, such as bandwidth, power, and computing resources. By learning from past usage patterns and user behavior, machine learning models can allocate resources more efficiently, ensuring fair distribution and minimizing wastage.
5. **Network Planning and Deployment**:- Machine learning techniques are used in network planning and deployment to optimize the placement of network nodes, antennas, and infrastructure components. By analyzing geographical data, terrain information, and user demographics, machine learning models can determine optimal locations for network deployment, maximizing coverage and minimizing interference.

### 1.8.1. The three main types of machine learning algorithms and their applications in communication networks:

1. **Supervised Learning**:- This involves training a model using labeled data to predict output for new, unseen data. In communication networks, supervised learning is used for tasks such as network traffic classification, link quality prediction, and resource allocation, where the goal is to make accurate predictions based on known input-output

relationships.

2. **Unsupervised Learning**:- This type of learning aims to discover hidden patterns and structures in data without labeled information. Algorithms like K-means clustering and DBSCAN are commonly used in communication networks for tasks such as anomaly detection and network topology identification, where the goal is to identify clusters or groups within the data based on similarities.

3. **Reinforcement Learning (RL):-** RL involves an agent learning to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. In communication networks, RL has been applied to problems such as dynamic spectrum access, power control, and routing optimization, where the agent learns to take actions that maximize cumulative rewards over time the essence of reinforcement learning (RL) well! RL indeed revolves around an agent learning to navigate an environment by taking actions and receiving feedback, typically in the form of rewards or penalties, to maximize long-term cumulative rewards.

In the context of communication networks, RL has shown promise in addressing various challenges, including:

1. **Dynamic Spectrum Access**:- RL algorithms can be used to adaptively allocate available spectrum resources to users or devices in real-time based on changing network conditions and demand. By learning optimal spectrum utilization strategies, RL agents can improve spectrum efficiency and minimize interference.

2. **Power Control**:- RL techniques can optimize power control strategies in wireless networks to enhance energy efficiency, extend battery life, and improve overall network performance. Agents learn to adjust transmission power levels dynamically based on channel conditions and traffic patterns to achieve desired objectives.

3. **Routing Optimization**:- RL algorithms can optimize routing decisions in communication networks by learning to select the most efficient paths for data transmission. Agents learn to balance factors such as latency, throughput, and energy consumption to optimize overall network performance.

By leveraging RL techniques in these areas, communication networks can become more adaptive, resilient, and efficient, ultimately enhancing user experience and enabling the successful deployment of emerging technologies like IoT and 5G.

## 1.9. Introduction to Qlearning:-

Q-learning is a fundamental reinforcement learning technique that has been widely applied in various optimization problems, including those encountered in communication networks. Here's a breakdown of its key characteristics and applications:

1. **Action-Value Function**:- Q-learning learns an action-value function (Q-function) that estimates the expected cumulative reward for taking a particular action in a specific state. This function guides the agent's decision-making process by associating actions with their potential long-term benefits.

2. **Simplicity**:- Q-learning is known for its simplicity and elegance. The algorithm iteratively updates Q-values based on observed rewards, requiring minimal prior knowledge about the environment. This simplicity makes it easy to implement and understand, even in complex network scenarios.

3. **Convergence Guarantees**:- Q-learning is guaranteed to converge to the optimal action-value function under certain conditions, ensuring that the agent eventually learns the best policy for maximizing rewards. This convergence property provides confidence in the effectiveness of Q-learning algorithms.

4. **Handling Unknown Environments**:- Q-learning excels in environments with unknown dynamics or complex state-action spaces. The agent explores the environment by trying different actions and learning from the observed rewards, gradually refining its policy over time to adapt to changing conditions.

In communication networks, Q-learning has been successfully applied to various optimization problems:

- **Resource Allocation**:- Q-learning can optimize resource allocation strategies by dynamically allocating network resources such as bandwidth, spectrum, or power to users or applications. The agent learns to balance competing objectives such as throughput, delay, and fairness to improve overall network performance.

- **Medium Access Control (MAC):-** Q-learning algorithms can optimize MAC protocols by learning to make efficient decisions regarding channel access, contention resolution, and transmission scheduling. This helps mitigate issues like collisions, latency, and energy consumption in wireless communication systems.

- **Routing**:- Q-learning techniques can optimize routing decisions by learning to select the most appropriate paths for data transmission based on network conditions and performance objectives. Agents learn to adaptively route traffic to minimize congestion, packet loss, and end-to-end delay.

By leveraging Q-learning in these areas, communication networks can become more adaptive, efficient, and resilient, ultimately enhancing the quality of service for users and applications.

Absolutely, machine learning, particularly reinforcement learning (RL), can play a crucial role in enabling intelligent actions and adaptive behaviors in IoT networks. Here's how RL can be applied to IoT nodes for accessing the communication channel under dynamic traffic load.

1. **Dynamic Channel Access**:- In IoT networks, the communication channel often experiences varying levels of congestion and interference due to the dynamic nature of IoT traffic. RL algorithms can enable IoT nodes to dynamically adjust their channel access strategies based on real-time network conditions. By learning from past experiences and feedback, IoT nodes can optimize their channel access decisions to maximize throughput, minimize latency, and ensure fair resource allocation.

2. **Adaptive Behavior**:- RL allows IoT nodes to adapt their channel access policies in response to changes in network topology, traffic patterns, and environmental conditions. For example, when congestion is detected, IoT nodes can learn to back off and retransmit packets at a later time to avoid collisions and improve overall network efficiency. Similarly, RL algorithms can help IoT nodes prioritize critical traffic and adjust transmission parameters dynamically to meet quality-of-service (QoS) requirements.
3. **Optimization of Resource Utilization**:- RL-based channel access schemes can optimize the utilization of available resources, such as bandwidth, spectrum, and energy, by intelligently allocating them to different IoT devices and applications. By learning to balance conflicting objectives, such as maximizing throughput while

   minimizing energy consumption, RL-equipped IoT nodes can achieve more efficient resource utilization and prolong network lifetime.
4. **Self-Learning and Adaptation**:- RL enables IoT nodes to learn from interactions with the environment and autonomously improve their channel access strategies over time. As network conditions evolve and new challenges arise, RL-equipped IoT nodes can continuously adapt and refine their behavior without human intervention. This self-

learning capability is particularly valuable in dynamic and unpredictable IoT environments where traditional rule-based approaches may fall short.

The proposed EQL-DQ algorithm in this study falls under the category of reinforcement learning, specifically leveraging the Q-learning technique. By combining Q-learning with distributed queueing along with joint contention and multiparallel channel model, EQL-DQ aims to adaptively manage the dynamic conditions of IoT networks and provide differentiated services to support the diverse quality-of-service needs of various IoT applications.

## 1.10. Introduce EQL-DQ :

The EQL-DQ, an algorithm aimed at enhancing the efficiency of IoT networks. By integrating Q-learning, which leverages reinforcement learning with an enhanced reward function, with distributed queuing techniques, EQL-DQ facilitates parallel execution of contention resolution queues (CRQ) and data transfer queues (DTQ). This joint approach works synergistically to improve overall network performance, contention management, and data transfer capabilities. EQL-DQ dynamically adjusts node contention, proficiently managing collisions and transmissions. The algorithm prioritizes balanced optimization of throughput, delay, and energy efficiency, thereby addressing challenges present in existing MAC protocols for IoT applications.

The EQL-DQ algorithm offers several key features that contribute to its efficacy in enhancing IoT network efficiency:

i. Reinforcement Learning-based Optimization:- EQL-DQ adopts a reinforcement learning-based approach to adaptively optimize the number of contending nodes. Leveraging Q-learning with an enhanced reward function, the algorithm dynamically adjusts the number of contending nodes in each contention period. This adaptive optimization aims to maximize throughput while minimizing energy consumption, thus improving overall network efficiency.

ii. Distributed Queueing Mechanism:- EQL-DQ incorporates a distributed queueing mechanism that enables parallel execution of contention resolution queues (CRQ) and data transfer queues (DTQ). This mechanism facilitates the joint contention resolution in the CRQ and parallel multichannel data transfer in the DTQ, leading to improved performance in contention management and data transfer capabilities. By concurrently handling contention resolution and data transmission tasks, EQL-DQ enhances network efficiency and resource utilization.

# CHAPTER-2

## Literature Review

In [30], the paper presents RC-SFAMA, a MAC protocol specifically designed for underwater acoustic networks, with the goal of overcoming the limitations associated with the prevalent slotted-FAMA protocol. RC-SFAMA integrates fundamental concepts from slotted-FAMA while introducing a novel time slot mechanism to mitigate data collisions, particularly in dense network environments. It also addresses the challenge of multiple RTS (Request-to-Send) attempts in dense networks through an RTS competition mechanism, thereby enhancing throughput efficiency. The protocol offers several advantages, including improved throughput efficiency, especially in dense network scenarios, reduced energy consumption by minimizing unsuccessful transmission attempts, and enhanced applicability in dense network deployments. However, it is noteworthy that RC-SFAMA exhibits certain limitations, such as increased complexity due to the RTS competition mechanism, the requirement for device synchronization, and sensitivity to fluctuating network conditions. Despite these limitations, RC-SFAMA represents a significant advancement in MAC protocols for underwater acoustic networks, offering promising solutions to address the unique challenges posed by such environments.

The paper [9] provides a comprehensive exploration of alternative Medium Access Control (MAC) protocols and techniques, aiming to address the limitations of the widely used ALOHA protocol in wireless communication systems. It delves into contention-based protocols like Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), emphasizing the utilization of carrier sensing and backoff procedures to enhance throughput by mitigating collisions. Moreover, reservation-based protocols such as Time-Division Multiple Access (TDMA) and Frequency-Division Multiple Access (FDMA) are examined, offering deterministic access and improved Quality of Service (QoS) support. The paper also investigates hybrid protocols, such as CSMA/CA with reservation, to leverage the advantages of both contention-based and reservation-based approaches. Furthermore, reinforcement learning-based MAC protocols are discussed for their potential to dynamically adapt to varying network conditions and traffic demands. These alternative MAC solutions offer various benefits, including enhanced throughput, spectral efficiency, QoS support, and adaptability to diverse communication requirements, such as those encountered in IoT and 5G contexts. However, the paper acknowledges that implementing and deploying these alternative MAC solutions may introduce additional complexity and overhead compared to the simpler ALOHA protocol, potentially hindering their widespread adoption.

In paper [22], a comprehensive distributed queue-based random access framework is proposed to address the significant challenges associated with supporting massive Machine-Type Communications (mMTC) within LTE/LTE-A networks featuring mixed-type traffic. The central idea revolves around employing a distributed queueing (DQ) mechanism, complemented by MAC layer load estimation, to effectively manage contention among massive Machine-Type Devices (MTDs) and enhance delay performance while minimizing disruptions to the LTE access procedure and air interface. A notable advantage of this framework is its ability to dynamically prioritize access by leveraging congestion information from the DQ process without incurring additional signaling overhead. Simulation results demonstrate that the proposed framework surpasses the 3GPP baseline Access Class Barring (ACB) scheme in terms of access delay and energy consumption, particularly in scenarios where all MTDs hold equal importance. Furthermore, in scenarios where MTDs possess different priorities, the framework can be fine-tuned to achieve lower delays across all classes and reduced overall energy usage compared to both the baseline and dynamic ACB solutions, particularly in bursty massive access scenarios. However, a potential limitation lies in the necessity for optimized parameter settings to fully harness the capabilities of the proposed DQ and prioritization techniques across a spectrum of diverse mMTC traffic patterns.

In [32], the paper addresses the shortcomings of the slotted ALOHA random access procedure utilized for Machine-to-Machine (M2M) or Machine-Type Communications (MTC), particularly concerning high user loads and limited available preambles. It introduces an enhancement to the 3GPP Extended Access Barring (EAB) mechanism by proposing a static and dynamic expansion of the contention space, aiming to mitigate collisions and increase the success probability to meet stringent delay and energy constraints. The primary advantage of this contention space expansion approach is its ability to outperform the existing 3GPP EAB mechanism in terms of access delay, collision rate, and energy efficiency, particularly in MTC scenarios with a massive number of devices. However, a potential limitation lies in the possibility of static expansion leading to inefficient resource utilization during low device loads, while dynamic expansion could introduce additional signaling overhead. The paper employs both an analytical model and simulations to assess the performance impact of the proposed contention space expansion on the 3GPP EAB mechanism across varying device loads, providing insights into its efficacy in real-world deployment scenarios.

In their paper [11], the authors present an enhanced SARSA(λ) reinforcement learning algorithm specifically tailored for wireless communication systems. SARSA(λ) is an extension of the traditional SARSA algorithm, leveraging eligibility traces to enhance learning efficiency by considering long-term action consequences. When applied to wireless communication systems, the primary objective is to optimize key system performance metrics such as throughput, energy efficiency, or quality of service. The reinforcement learning approach enables adaptive decision-making based on interactions with the environment and feedback in the form of rewards or penalties. The benefits of the enhanced SARSA(λ) algorithm include demonstrated superiority over traditional SARSA in terms of convergence speed and overall system performance, adaptability to changing network dynamics, and potential for more efficient resource allocation. However, several challenges are noted, including increased computational complexity, the exploration-exploitation tradeoff inherent in reinforcement learning, sensitivity to initial conditions, scalability concerns for larger networks, and the absence of theoretical guarantees on convergence or optimality. Despite these challenges, the enhanced SARSA(λ) algorithm represents a promising approach for optimizing wireless communication systems by leveraging the power of reinforcement learning techniques.

In the [5] realm of Machine-Type Communications (MTC) and Internet of Things (IoT) environments, characterized by numerous devices contending for channel access, traditional single-channel ALOHA faces inherent limitations due to congestion, collisions, and reduced throughput. To address these challenges and accommodate the growing device population, multichannel ALOHA emerges as a promising solution. Multichannel ALOHA extends the ALOHA concept by allowing devices to transmit data over multiple channels randomly, thereby mitigating congestion and enhancing system throughput. Variations, such as segmenting channels into data and control channels, have been explored to analyze their throughput and stability properties. The adoption of multichannel ALOHA offers several benefits, including increased throughput, improved channel load balancing, and reduced potential delays and collisions. However, traditional multichannel ALOHA still faces challenges such as high collision rates under heavy channel loads, absence of channel sensing or exploration mechanisms, and the lack of means to estimate or adapt to the dynamic number of active devices. Therefore, further optimization is necessary to ensure efficient operation in MTC and IoT contexts.

In [3], the paper by Wu et al. introduces a distributed queueing-based random access protocol tailored for LoRa (Long-Range) networks, a popular low-power wide-area network (LPWAN) technology extensively utilized in various Internet of Things (IoT) applications. The protocol aims to tackle the challenges associated with efficient channel access and collision avoidance in dense LoRa networks, where numerous devices contend for limited radio resources. The key concepts discussed include: Distributed Queueing: The protocol employs a distributed queueing mechanism to coordinate channel access among LoRa devices. Each device maintains a local queue to manage its transmission attempts. This approach, combined with a random access scheme, reduces the likelihood of collisions and enhances overall network performance. The protocol accounts for specific characteristics of LoRa networks, such as spread-spectrum modulation, limited orthogonal channels, and bursty IoT traffic. The distributed queueing-based random access protocol for LoRa networks offers several benefits: Improved Channel Access Efficiency: The distributed queueing mechanism enhances channel access coordination, reducing collisions and improving overall network throughput. Scalability: The protocol accommodates a large number of LoRa devices in dense networks, making it suitable for IoT applications with high device density. Reduced Latency: By prioritizing channel access, the protocol minimizes transmission latency experienced by LoRa devices. Energy Efficiency: The random access scheme and distributed queueing approach improve energy efficiency by reducing failed transmission attempts and unnecessary channel access. However, the protocol faces some potential limitations: Complexity: Implementing distributed queueing and associated decision-making processes may introduce additional complexity, impacting computational and memory requirements. Synchronization: The protocol assumes device synchronization, which may be challenging in decentralized network environments. Sensitivity to Network Dynamics: Performance may be sensitive to changes in network conditions, requiring adaptive mechanisms for optimal operation. Overhead: Additional signaling, control information exchange, and processing requirements may impact energy consumption and network efficiency. Fairness: The protocol may not provide strict fairness in channel access opportunities for all devices, especially in congested networks. Limited Analytical Insights: The paper primarily focuses on protocol design and simulation-based performance evaluation, lacking in-depth analytical insights into theoretical properties and performance bounds. Despite these limitations, the distributed queueing-based random access protocol presents a promising approach to address challenges in LoRa networks, offering enhanced efficiency and scalability for IoT applications.

In [29], Abedi and Pourhasani present a paper titled "Prioritized Multi-Channel MAC Protocol in Ad Hoc Networks Using a TDMA/CSMA Approach," aiming to address the demand for efficient medium access control (MAC) protocols in ad hoc networks. Emphasizing the importance of channel prioritization and hybrid TDMA/CSMA techniques, the paper proposes a prioritized multi-channel MAC protocol. This protocol integrates Time Division Multiple

Access (TDMA) and Carrier Sense Multiple Access (CSMA) methods to enhance network performance by improving channel utilization and reducing contention, thereby boosting overall throughput and reliability. The proposed MAC protocol offers benefits such as increased network capacity, reduced packet collisions, and enhanced Quality of Service (QoS). By prioritizing channels and employing a hybrid TDMA/CSMA approach, the protocol aims to optimize resource utilization, particularly in scenarios with dynamic traffic patterns. However, challenges such as increased implementation complexity and variability in effectiveness based on network topology and traffic characteristics may arise. Further research and experimentation are necessary to comprehensively evaluate the performance of the prioritized multi-channel MAC protocol.

In paper [2], Distributed Queuing (DQ) emerges as a promising solution, offering collision-free contention-based random access to enhance the scalability of LoRa networks compared to the default Aloha protocol. The fundamental idea behind DQ involves organizing nodes into a virtual queue using a collision resolution algorithm and servicing them sequentially to access the channel, thus mitigating collisions typical of Aloha. DQ presents several advantages, including near-optimal throughput performance irrespective of traffic load, bounded access delay, and decentralized operation without necessitating synchronization. However, existing DQ schemes, such as the Collision Tree Algorithm, were initially devised for cable TV networks featuring full-duplex links, rendering them inefficient for the half-duplex LoRa environment due to high signaling overhead stemming from the contention resolution process. Extending DQ to multichannel LoRa networks poses another challenge, namely achieving balanced load distribution across channels. Despite DQ's superiority over Aloha in dense LoRa deployments, meticulous protocol design is imperative to realize these advantages, particularly considering LoRa's half-duplex nature and utilization of multiple channels.

The paper [31] introduces a novel Medium Access Control (MAC) protocol named Hybrid ALOHA (H-ALOHA), which amalgamates the Pure ALOHA (P-ALOHA) and Slotted ALOHA (S-ALOHA) protocols to cater to specific needs in wireless networks, including energy conservation, delay reduction, and throughput enhancement. The fundamental idea is to capitalize on the respective strengths of both protocols: P-ALOHA exhibits superior energy efficiency and packet delivery for individual transmissions, while S-ALOHA generally outperforms P-ALOHA. To evaluate the steady-state performance of H-ALOHA comprehensively, the paper develops a finite-state Markovian model, encompassing metrics such as normalized throughput, backlogged throughput, access delay, backlogged delay, and energy consumption. Through simulations, the proposed hybrid protocol demonstrates significant advantages, surpassing S-ALOHA by 60% in normalized throughput, 15% in access delay, and 23% in total energy consumption during transmission on average. However, it's important to note a potential limitation: the analysis is confined to specific scenarios and assumptions, and the protocol's performance might vary under different wireless network conditions and requirements.

In [1], the Internet of Things (IoT) has emerged as a pivotal technology across various domains, ranging from smart cities to industrial automation. Efficient Medium Access Control (MAC) protocols play a critical role in ensuring reliable and energy-efficient data transmission within IoT networks. One promising approach to tackle the challenges inherent in IoT environments is the utilization of Distributed Queuing (DQ) MAC protocols, which aim to mitigate collisions and enhance system throughput. The paper discusses the Q-learning-based Distributed Queuing Medium Access Control (QL-based DQMAC) protocol tailored for IoT networks, which endeavors to optimize the number of contending nodes in each contention period to reduce collision probability. The protocol leverages the Q-learning algorithm, empowering each node to compute its active rate autonomously and decide whether to remain active or enter a sleep mode in the subsequent contention period. This adaptive behavior contributes to improvements in throughput, delay, and energy consumption. However, the protocol's limitations include its lack of consideration for traffic prioritization, potentially hindering its ability to effectively manage the diverse Quality of Service (QoS) requirements characteristic of IoT applications.

In [23], H. Hassen, S. Meherzi, and Z. Ben Jemaa delve into the realm of Software-Defined Networking (SDN) to tackle the pressing issue of optimizing multipath routing. Recognizing the paramount importance of efficient resource allocation and heightened network performance, the paper introduces refined strategies for multipath routing based on Q-learning. These strategies aim to enhance exploration mechanisms to achieve more effective resource utilization and bolster overall network efficiency within SDN environments. The proposed approach promises several benefits, including heightened network efficiency, congestion reduction, and improved quality of service. By employing innovative exploration strategies, the study seeks to alleviate network bottlenecks and elevate the overall user experience within SDN networks. However, the paper acknowledges certain limitations, notably potential scalability issues and overhead associated with implementing Q-learning in large-scale SDN networks, underscoring the need for further exploration and optimization in this domain.

| Approach | Advantages | Disadvantages |
|---|---|---|
| EQL-DQMAC (proposed) | Adaptive optimization, energy-efficient, Reduced contention and delay | Complexity, overhead, tuning challenges, scalability concerns |
| QL-based DQMAC [1] | Optimized contending nodes, throughput, delay, energy | No prioritization, QoS limitations |
| Multichannel Aloha [5] | Increased throughput, Improved load balancing, Reduced delays | High collision rates, Lack of sensing mechanisms, Inability to adapt |
| SARSA(λ) [11] | Convergence, adaptability, resource allocation | Complexity, exploration-exploitation trade-off, scalability, lacks guarantees |
| RC-SFAMA [30] | Throughput in dense networks, energy-efficient | Complexity, synchronization, network sensitivity |
| Energy-efficient MAC protocols [6] | Energy efficiency, lifetime | Dynamic IoT limitations, QoS limitations |
| Distributed Queueing for LoRa Networks [3] | Channel access, scalability, latency, energy | Complexity, synchronization, dynamics sensitivity, overhead, fairness |
| Alternative MAC Protocols [9] | Throughput, spectral efficiency, QoS, adaptability | Complexity, overhead |
| RL-MAC [16] | Adaptability, QoS differentiation, resource utilization | Computational overhead, reward design, single-hop |
| Distributed Queue-based Random Access Framework [22] | Comprehensive framework, Effective contention resolution, Dynamic access prioritization | Parameter optimization, LTE access impact, Traffic pattern adaptation |
| Improved Exploration Strategy for Q-Learning [23] | Efficiency, congestion reduction, QoS | Scalability issues, Q-learning overhead |
| Prioritized Multi-Channel MAC Protocol [29] | Capacity, collision reduction, QoS | Implementation complexity, topology/traffic sensitivity |

Table.2.1. Advantage and Disadvantages of various protocol

# Chapter-3

# Q-Learning

## 3.1 Q-learning in Markovian environments

In the context of estimating the number of contention nodes within a fixed frame length, the objective is to reduce collision probability. However, in scenarios where contention node numbers are high, collisions and packet propagation delays tend to increase, subsequently lowering IoT network channel utilization. Addressing this challenge necessitates the limitation of contention nodes. Yet, the dynamic nature of IoT network traffic load and contention node numbers complicates this task. Herein lies the significance of the Q-learning (QL) algorithm, which operates effectively even without prior knowledge of the environment. By continuously adapting to ongoing interactions between IoT nodes and their environment, QL optimizes system performance by accurately estimating contention node numbers and adjusting accordingly.

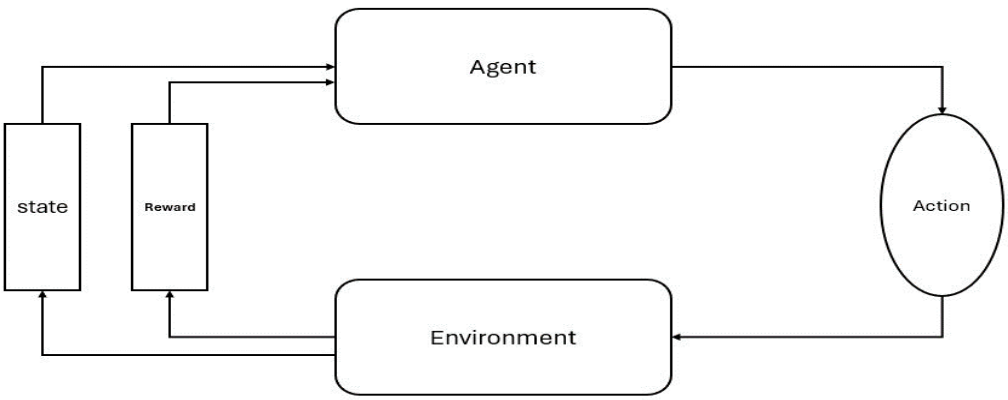


Fig.3.1: The Agent-Environment Interaction in a Markov Decision Process

The Q-learning (QL) algorithm stands out for its simplicity, requiring minimal computational resources from MAC controllers while keeping communication overhead low for IoT network nodes. In wireless networks, the reliability of unicast communication is often ensured through reception acknowledgments. Although widely used in contention-based MAC protocols, this acknowledgment scheme significantly adds to their overall overhead.

In Fig. 3.1, the MDP presents a straightforward problem framework, employing interaction learning to achieve its objectives. The MDP's elements—states (S), actions (A), and rewards (R)—are finite. Both Rt and St represent random discrete probability distribution variables, influenced by preceding states and actions. In essence, all prior states and actions hold probability within specified values for all s′ ,s ∈ S, r ∈ R, and a ∈ A(s):

$$p(s', r \mid s, a) = \Pr(St = s', Rt = r \mid St-1 = s, At-1 = a), \tag{3.1}$$

With

$$\sum\sum p(s, r|s, a) = 1, \forall s \in S, a \in A(s) s' \in S\ r \in R \tag{3.2}$$

In the MDP, the function ( p ) is dynamic and defined as ( p : S ×R ×S ×A →[0, 1]), representing an ordinary deterministic function with four arguments. The policy represents a potential action chosen by a state within the IoT environment. If the agent adheres to policy ( π) at time step ( t ), then ( π(a ∣ s)) denotes the probability of ( A_t= a ) given ( S_t= s ). The MDP is tackled using an RL mechanism,without necessitating complete information. Within the QL system, the agent, environment, policy, reward, and Q-value function constitute the fundamental elements.

## 3.2. Q-learning Function Overview:

In the realm of Reinforcement Learning (RL), the Q-learning function plays a pivotal role in guiding the decision-making process of agents operating within IoT environments.

## Policy Definition:

A policy dictates the actions taken in specific states within an IoT environment. Typically, it's represented as a function or lookup table and serves as the heart of an RL agent, determining its behavior. This policy can be deterministic or probabilistic, guiding actions with specified probabilities.

**Reward Definition:**

The reward defines the objective of the RL mechanism, being a single numerical value relayed to the agent within the RL IoT environment. The sole aim of the agent is to maximize this reward. If the reward obtained from a policy's selected action is deemed low, the agent adjusts its course by selecting alternative actions and potentially modifying the policy for similar future scenarios.

**Learning and Adaptation:**

Through interactions with the environment, the agent learns from experience, updating its Q-values based on observed rewards and transitions between states. This iterative process of learning allows the agent to refine its decision-making strategy over time, ultimately maximizing cumulative rewards and achieving optimal performance within the IoT environment.

**Role of Reward in Q-learning:**

In Q-learning, the reward obtained from an action under a policy serves as a crucial feedback mechanism for the agent within IoT environments. When the reward obtained from the selected action under a policy is high, it indicates that the Q-value effectively signals success for that scenario. In essence, the Q-value represents the cumulative reward associated with a state.

**Influence on Agent Behavior:**

Future actions are then guided by this Q-value, shaping the agent's behavior accordingly. If a particular action consistently leads to high rewards, the agent is more likely to select that action in similar future scenarios. Conversely, if the reward obtained from a selected action is low, the agent adjusts its behavior to explore alternative actions and improve its decision-making strategy.

## 3.3. Application in IoT Networks:

In high collision probability scenarios within IoT networks, the frequency of successful transmissions during time slots diminishes, while the incidence of failed transmissions rises. This dynamic environment highlights the importance of effective reward signals in guiding the agent's behavior towards actions that maximize successful transmissions, thereby improving network efficiency and performance.

## 3.4. Reward Calculation in IoT Environments

In IoT environments, the likelihood of channel collisions significantly impacts the success of transmission attempts. This correlation is crucial in determining the reward structure for the reinforcement learning agent. The calculation of rewards involves considering several factors:

1. **Number of Contention Periods per Beacon Interval**: Higher collision probabilities lead to an increased number of contention periods within each beacon interval. This results in lower, potentially negative rewards due to the heightened competition for channel access.

2. **Number of Contentions per Node until Successful Slot Access:** Each node contends for access to the channel, with the number of contentions required varying based on

collision probabilities. Higher collision probabilities result in more failed transmission attempts per node, leading to lower rewards.

3. **Influence of Collision Probabilities on Rewards:**Increased collision probabilities lead to a decrease in successful transmission time slots and an increase in failed transmission occurrences, directly impacting reward calculations. Conversely, lower collision probabilities result in higher rewards, reflecting the improved efficiency of channel access.

4. **Integration of Q-Value Estimations:**In addition to collision probabilities, rewards are derived from Q-value estimations. The Q-values guide the agent in selecting optimal actions by indicating the potential long-term benefits associated with each action. When Q-values are absent, indicating a reward of zero, the agent relies solely on the Q-value function to make decisions.

5. **Optimal Action Selection and Long-Term Reward Optimization:**The agent selects optimal actions based on maximizing Q-values, ultimately aiming for long-term reward optimization through a series of actions. While maximizing individual rewards is essential, it's important to note that achieving success in IoT environments may not always require maximizing immediate rewards. Instead, the focus lies on selecting actions that lead to the most favorable long-term outcomes, considering factors such as network stability, energy efficiency, and overall system performance.

## 3.5. Q-learning Algorithm:

In Q-learning (QL), the agent maintains a Q-table denoted by ( $Q(S_t, A_t)). At$ each time step ( t ) in an IoT network, the agent observes the state ( $S_t$) of the Markov Decision Process (MDP). It then selects an action ( $A_t$) from the set of available actions ( $A$ ). After taking action ( $A_t$), the agent receives a reward ( $R(t)$) and observes the next state ( $S_{\{t+1\})}$.This sequence of events—observation of state ( $S_t$), action selection ( $A_t$), reward reception ( $R(t)$),and observation of next state ( $S_{\{t+1\})}$—constitutes the learning experience of the agent. The agent updates the sequence of events associated with the state-action pai$r$ ( ($S_t, A_t$)) in the Q-table according to the QL function.

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha[R(t)+1 + \gamma \max_{a} Q(S_t+1, a) - Q(S_t, A_t)] \tag{3.3}$$

The agent selects an action based on the current state ( $S_t$). Then, the maximum Q-value for the next state ( $S_t + 1$) is determined according to the action ( $A_t$), and this maximum Q-value updates the current Q-value.The discount factor ( $\gamma$) typically falls within the range ( $0 < \gamma < 1$ ). It models the importance of future rewards. When ($\gamma$) is set to 0, the agent prioritizes immediate rewards, ignoring future ones. Conversely, when ( $\gamma$) is set to 1, the agent aims to maximize long-term rewards. The learning rate ( $\alpha$ $(0 < (\alpha) < 1)$ dictates how much the new information overrides the old in the Q-value updates.

### 3.5.1. Influence of Learning Rate on Estimation Adjustments:

The learning rate (α) plays a crucial role in adjusting estimations within the EQL-based DQMAC framework. It determines the speed at which the agent learns from new information and updates its estimations.

- Higher Learning Rate:- A higher learning rate accelerates the adjustment process, allowing the agent to quickly incorporate new information into its estimations. This rapid adjustment can be beneficial in dynamic environments where changes occur frequently. However, it may also lead to instability if the learning rate is too high, causing the agent to overreact to individual observations.

- Lower Learning Rate:- Conversely, a lower learning rate slows down the adjustment process, providing the agent with more stability and robustness against sudden changes in the environment. While slower to adapt, a lower learning rate allows the agent to integrate information more gradually, resulting in smoother estimation updates. However, setting the learning rate too low may cause the agent to converge slowly or get stuck in local optima.

### 3.5.2. Impact on Learning Behavior:

- $(\alpha = 0){:}-$ When the learning rate (α) is set to 0, no new learning occurs, and the agent's estimations remain unchanged. This scenario effectively freezes the learning process, preventing the agent from updating its estimations based on new observations.

- (α = 1):- Conversely, if the learning rate (α) is set to 1, only the most recent information is considered in the estimation updates. In this case, the agent completely disregards past observations and focuses solely on the latest data, potentially leading to volatile estimation adjustments.

# CHAPER-4

## Proposed Methodology  EQL-based DQMAC

In EQL-based DQMAC, the learning rate (alpha) influences the calculation of the number of IoT nodes contending in each contention period (i) according to the Q-learning mechanism. By adjusting the learning rate appropriately, the protocol can strike a balance between responsiveness to changing environmental conditions and stability in estimation updates, ultimately optimizing network performance and efficiency.

$$N_{\text{con}}^{i} = \begin{cases} N_{\text{avg}} - \sum_{j=1}^{i-1} N_{\text{suc}}^{j} & 2 \le i \le n; \\ N_{\text{avg}} & i = 1. \end{cases} \tag{4.1}$$

The active rate of the number of contention nodes in contention period *i* is defned as follows:

$$\text{Rate}_{\text{active}} = 1 - \frac{N_{\text{sleep}}}{N_{\text{con}}^{i}} = 1 - \frac{N_{\text{con}}^{i} - X}{N_{\text{con}}^{i}} \tag{4.2}$$

Each contention node decides whether to remain active or enter a sleeping state in the subsequent contention period based on the active rate. ( $N_{\{\{sleep\}\})}$ represents the count of sleeping nodes during the contention period, while ( X ) denotes the targeted optimal number of nodes in the contention period.

In a distributed queuing MAC protocol, IoT nodes succeed and transmit data sequentially as per the resolution mechanism. Hence, ( $N_{\{\{avg\}\})} and$ ( $N_{\{\{sleep\}\})}$ vary over time based on this mechanism.

In the proposed EQL-based DQMAC protocol, the determination of the number of contention IoT nodes for the upcoming contention period is crucial. If this count is significant, indicating a high level of contention among nodes, DQ mechanism with the Joint Contention for CRQ and parallel dtata transfer from DTQ using Multiparallel channel . This mechanism efficiently coordinates channel access among the nodes by managing a queue system, thus mitigating collisions and improving overall network performance. On the other hand, if the count of contention nodes is minimal, implying lower contention levels, all contention nodes will participate in the same contention period without the need for the distributed queuing mechanism, simplifying the access process while still ensuring effective channel utilization.

This adaptive approach allows the protocol to dynamically adjust its operation based on the prevailing network conditions, optimizing performance accordingly.

Typically, the optimal contention nodes cannot exceed the total number of time slots in the contention period, denoted as ( $X \leq m$ ). All sleeping nodes intending to transmit data will awaken in the subsequent contention period, with new sleeping nodes established during that period. In the next contention period, all contention nodes include the failed contention nodes and the sleeping nodes from the previous period. The active rate is determined by each IoT node itself using Eq. (4.2) after each contention period, enabling the node to decide whether it will be active or in sleep mode in the next contention period.The decision to employ the distributed queuing mechanism depends on a specific condition, defined as follows:

$$\mathbf{DQcondition=X} \tag{4.3}$$

If the number of contention nodes exceeds a threshold, denoted as $\{DQ\_condition\}$, then the distributed queuing mechanism with joint contention for contention resolution queue and multiparallel channel for data transfer is executed. Otherwise, all contention nodes proceed to the next contention period without employing the Enhanced Q learning distributed queuing mechanism.

## 4.1. Proposed EQL DQMAC:

The proposed scheme, termed the EQL-DQMAC protocol, introduces several enhancements to the Q-learning algorithm to optimize the number of contention nodes effectively. It leverages Q-learning, a reinforcement learning technique, to dynamically determine the optimal number of contention nodes for each contention period, thereby minimizing collision probabilities and improving network throughput.

Additionally, the EQL-DQMAC protocol incorporates parallel concepts for managing the Data Transfer Queue (DTQ) and Contention Resolution Queue (CRQ). By employing joint contention for contention resolution and a multiparallel channel model for data transfer, the protocol enhances overall network performance and efficiency. This parallel execution strategy allows for improved utilization of available resources and reduced contention delays.

Furthermore, the EQL-DQMAC protocol enhances the reward function based on delay. By considering delay as a critical factor in determining the reward for each action taken by the nodes, the protocol incentivizes actions that lead to reduced transmission delays. This incentivization encourages nodes to make decisions that contribute to minimizing overall.
network latency, thereby enhancing the quality of service (QoS) experienced by users.

Overall, the EQL-DQMAC protocol represents a comprehensive approach to medium access control in IoT networks, combining Q-learning-based optimization with parallel queue management and an enhanced reward function. This scheme aims to address the challenges of contention and delay in wireless communication environments, ultimately improving network performance and efficiency.

## 4.2. Joint contention tree structure:

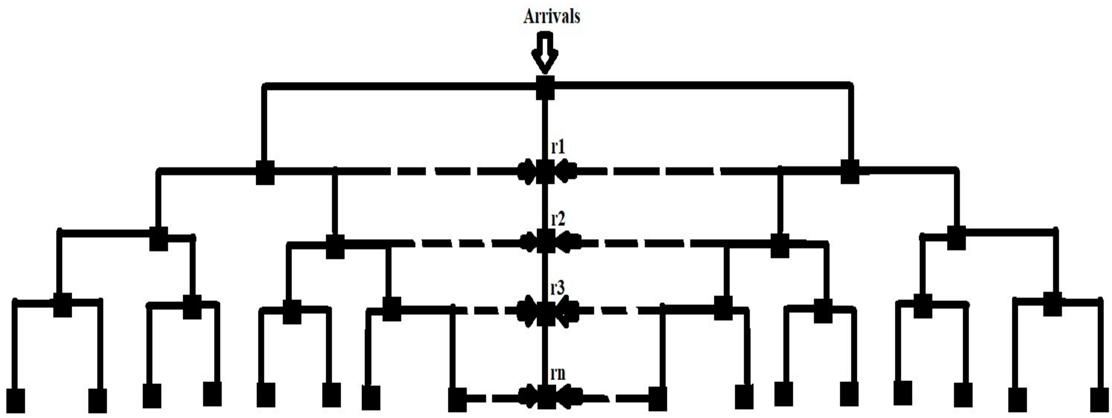


Fig.4.1. Joint Contention

In the proposed research paper, a novel contention resolution mechanism inspired by the joint contention tree structure is introduced, aiming to achieve efficient bulk collision resolution in communication networks. Drawing inspiration from the Augmented Contention Mechanism (ACM) algorithm, commonly utilized in wireless and multiple access networks, the proposed approach focuses on mitigating collision propagation across multiple leaf nodes within each contention round.

At the heart of this approach is the concept of the Joint Contention Tree (JCT), which employs a unique technique of merging all newly created leaf nodes at every tree level. This merging process ensures that all stations involved in a collision participate in every contention round until the contention is successfully resolved. By preventing the fragmentation of colliding stations across different branches of the tree, the JCT structure enhances collision resolution efficiency.

The primary objective of implementing the JCT structure is to improve the performance of the Contention Resolution Queue (CRQ) component within the network architecture. By efficiently handling collisions, the CRQ can minimize the overhead associated with contention resolution, ultimately leading to enhanced network throughput.

Overall, the proposed contention resolution mechanism offers a promising approach to address the challenges of collision management in communication networks. By leveraging the principles of bulk collision resolution and efficient queue management, the JCT structure has the potential to significantly improve the reliability and performance of network communication systems.

## 4.3. Multiparallel channel Model:

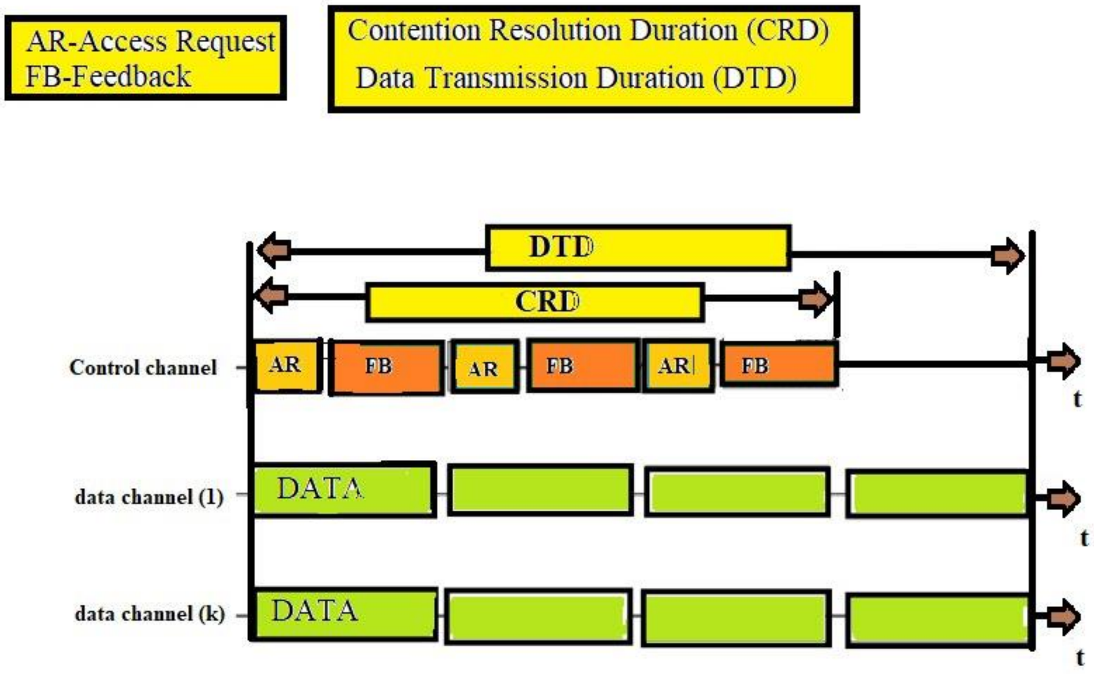


Fig-4.2. Structure of Multiparallel channel

The figure illustrates the structure of the control channel model proposed in this paper for the EQL-based distributed queuing MAC protocol designed for IoT networks.

Here are the key elements outlined in the figure:

**1. Control Channel:**

- The control channel serves as the medium for the contention resolution process.
- It comprises two main slots: Access Request (AR) and Feedback (FB) slots.

**2. Data Transmission Duration (DTD):**

- This segment represents the duration allocated for the transmission of actual data after the contention resolution process is completed.

**3. Contention Resolution Duration (CRD):**

- The CRD denotes the duration dedicated to the contention resolution process, during which stations compete for access to the shared medium.

**4. Control Channel:**

- The control channel is a singular channel utilized for the contention resolution phase. All contending stations initially select this channel to participate in the contention process.

**5. Data Channels (1 to k-1):**

- These channels are designated for the transmission of actual data.
- Following the contention resolution on the control channel, the successfully contending stations proceed to transmit their data over these data channels.

Overall, the proposed control channel model delineates the allocation of time slots and channels for contention resolution and subsequent data transmission, thereby facilitating efficient communication within IoT networks.

The provided figure illustrates the proposed EQL DQMAC protocol, focusing on the contention resolution mechanism within three time slots and involving six nodes (d1, d2, d3, d4, d5, d6). Here's a breakdown of the figure:

**(a) Initial State:**

- Three nodes (d1, d2, d3) are assigned to the first time slot.
- One node (d5) is assigned to the second time slot.
- Two nodes (d4, d6) are assigned to the third time slot.

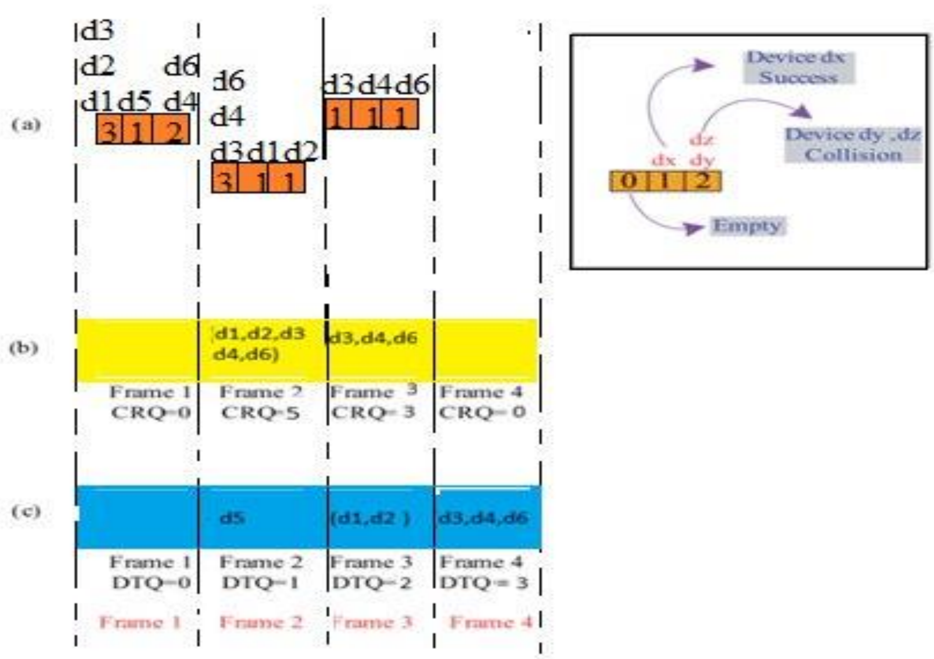


Fig-4.3. Proposed EQL-DQ MAC Working

**(b) Contention Resolution Queue (CRQ):**

- The CRQ manages contention resolution.
- Frames CRQ-0, CRQ-5, CRQ-3, and CRQ-0 depict the progression of contention resolution.
- Nodes involved in collisions are queued for resolution.

**(c) Data Transfer Queue (DTQ):**

- The DTQ manages successful data transfers using a multiparallel channel model.
- Frames DTQ-0, DTQ-1, DTQ-2, and DTQ-3 illustrate the progression of data transfer.
- Nodes successfully transmitting data are added to the corresponding frames.

## 4.4. Operation of the Proposed Mechanism:

1. Initially, all nodes attempt transmission.
2. Node d5 successfully transmits in the second time slot and is added to DTQ-1.
3. Collisions occur in the first and third time slots.
4. Nodes involved in collisions are combined and queued in the CRQ for joint contention resolution (CRQ-5).
5. Contention resolution continues in subsequent frames (CRQ-3, CRQ-0) until all nodes successfully transmit or a maximum iteration limit is reached.
6. Nodes successfully transmitting data are added to the corresponding frames in the DTQ.
7. The joint contention resolution in the CRQ and parallel multichannel data transfer in the DTQ work together to enhance network performance, contention management, and data transfer capabilities.

## 4.5. Detail principles of Enhanced Q-learning:

### 4.5.1Action selection:

The optimal number of contention nodes is modified before each beacon interval by undertaking one of the following actions:

$$Xt + 1 \leftarrow - Xt, a \in (Xt - d, Xt, Xt + d), \tag{4.4}$$

In the context provided, ( $X_t$) represents the optimal number of contention nodes at time step ( $t$ ), and ( $d$ ) denotes the variation in the number of contention nodes.

Reinforcement Learning (RL) stands out from supervised and unsupervised learning methods due to its unique approach to learning from interactions with an environment. Unlike supervised learning, where the model learns from labeled examples provided by a supervisor, and unsupervised learning, where the model learns patterns from unlabeled data, RL learns from feedback received through its actions in an environment.

One of the primary challenges in RL is the exploration-exploitation dilemma. The agent must balance between exploiting actions that are known to yield rewards based on past experiences and exploring new actions that may lead to even higher rewards in the future. If the agent exploits too much, it may miss out on potentially better actions. Conversely, if it explores too much, it may waste time and resources on actions that do not yield immediate rewards.

To address this challenge, RL algorithms employ various strategies to balance exploration and exploitation effectively. These strategies include epsilon-greedy policies, where the agent selects the best-known action most of the time but occasionally explores random actions with a certain probability (epsilon). Other methods include softmax action selection, where the probabilities of selecting each action are determined by their estimated values, and Upper Confidence Bound (UCB) algorithms, which prioritize actions with higher uncertainty.

To navigate this challenge, the agent avoids repeating actions that previously failed to yield rewards and instead capitalizes on experiences that were rewarding. However, it must also explore new actions to identify potentially better strategies for future situations. Consequently, neither exploration nor exploitation can be pursued exclusively, presenting a dilemma.

One of the simplest action selection rules involves choosing the action with the highest estimated value. In cases where multiple actions possess the maximum estimated value, one of them is randomly selected. This method of selecting greedy actions is commonly described as "greedy action selection".

$$\pi(s). = argmax_a Q(s, a), \quad (4.5)$$

where $argmax_a$ denotes the action a for which the expression is maximized.

**4.5.2 Convergence requirements:**

The convergence of the QL algorithm is based on all actions that are repeatedly sampled in all states. Furthermore, the action values are indicated discretely, with an optimal action-value probability of 1 when the QL algorithm converges [17].

In maximizing immediate reward, the current state of knowledge often leads to greedy-action selection. However, this approach doesn't allow for additional sampling time to explore potentially better options. To address this limitation, a small probability £ is introduced, enabling exploration alongside exploitation. This means that, in most cases, greedy-action selection is augmented with a small probability ε.The ε-greedy method is defined such that the greedy action is not randomly selected from all actions with the same probability. Instead, it is independently selected according to the action-value estimates. As the number of steps increases, the number of samplings for each action approaches infinity. Qt(a) is guaranteed to converge to q∗(a), which implies that the converged probability of selecting an optimal action will be greater than 1 − ε.The convergence speed of QL algorithms depends on the application and its associated environmental complexities [18]. When QL is applied in a new environment, the agent must explore and exploit the reward to gradually discover the optimal action At that maximizes the Q-value. Note that ε is defined as follows:

$$\varepsilon = e^{-\frac{Trun}{Tsimu}} \tag{4.6}$$

where Trun is the running time and Tsimu is the system simulation time. Convergence to an optimal policy is guaranteed by the decay function in our proposed QL-based MAC for an IoT-enabled MANET.

### 4.5.3. A priori approximate controller:

In the iterative Q-learning algorithm, such as the one defined in Equation 4.6, an initial condition is necessary. After the first update, this initial condition changes. For Q-learning, the initial condition is always set to zero. The initial untrained Q-table is structured as shown in Table 3: the rows represent possible states, while the columns represent the action space.In this setup, the rows denote the optimal number of contention nodes, where ( $Xt$ ) represents the current number of contention nodes at time step ( t ). ( $Xt - d$ ) indicates a decrease in the number of contention nodes by ( d ), while ( $Xt + d$ ) denotes an increase in the number of contention nodes by ( d ).The initial values for the untrained Q-table with size [m, 3] are all set to zero, except for ( Q[0, 0] ) and ( Q[m, 2] ), which are initialized to extreme negative values (-100). This ensures that these states are never visited by the IoT node.

| Xt - d | Xt | Xt + d |
|---|---|---|
| -100 | 0 | 0 |
| 0 | 0 | 0 |
| : | : | : |
| : | : | : |
| 0 | 0 | 0 |
| 0 | 0 | -100 |

Table.4.1. Initial Q-value table for our proposed EQL-based MAC scheme

Each node uses a controller that depends on the traffic rate and the number of contention nodes for each contention period. In this study, all the nodes are in a one-hop environment, and the traffic destination is either the cluster head or other IoT nodes in the IoT network. The controller is trained a priori for $\gamma = 0.9$

### 4.5.4. Formulation of the Reward Function:

The QL agent receives positive or negative rewards based on their ability to learn and act correctly in an IoT network. The primary function of IoT networks is to accomplish Low contention collision ensures excellent system throughput, little propagation latency, and low energy MAC contention. This can be accomplished with the binary reward function. The agent receives a reward based on the outcome of the conflict.

The reward function is defined as:

$$reward = \mu * \left(\frac{avg_{contentions}}{(len(contention_{period}) + \varepsilon)}\right) + \nu * \left(\frac{avg_{contentions}}{(sum(contention_{period}) + \varepsilon)}\right) - rho \ * \ delay \tag{4.7}$$

\`μ\` is the weighting factor that determines the importance of minimizing the average contentions normalized by the number of contention slots, \`ν\` is the weighting factor that determines the importance of minimizing the average contentions normalized by the total number of contending nodes, and \`rho\` is the weighting factor that determines the importance of minimizing the average delay experienced by packets.

Let's break down the different components of this reward function and understand how they contribute to the "Enhanced" aspect of the algorithm:

**1. Minimizing Average Contentions:**

- The first two terms in the reward function aim to minimize the average contentions.
- \`μ* (avg_contentions / (len(contention_period) + ε))\`:- This term represents the average contentions normalized by the number of contention slots. By minimizing this term, the algorithm is incentivized to reduce the average number of contentions in each contention slot.
- \` ν* (avg_contentions / (sum(contention_period) + ε))\`:- This term represents the average contentions normalized by the total number of contending nodes. By minimizing this term, the algorithm is incentivized to reduce the overall average contentions across all contending nodes.
- Minimizing the average contentions is crucial for improving the throughput of the network, as fewer collisions lead to fewer retransmissions and more successful transmissions.

**2. Minimizing Delay:**

- The third term in the reward function, \`-rho * delay\`, represents the negative of the average delay experienced by packets.
- By minimizing this term, the algorithm is incentivized to reduce the overall delay in the network, which is particularly important for time-sensitive IoT applications.

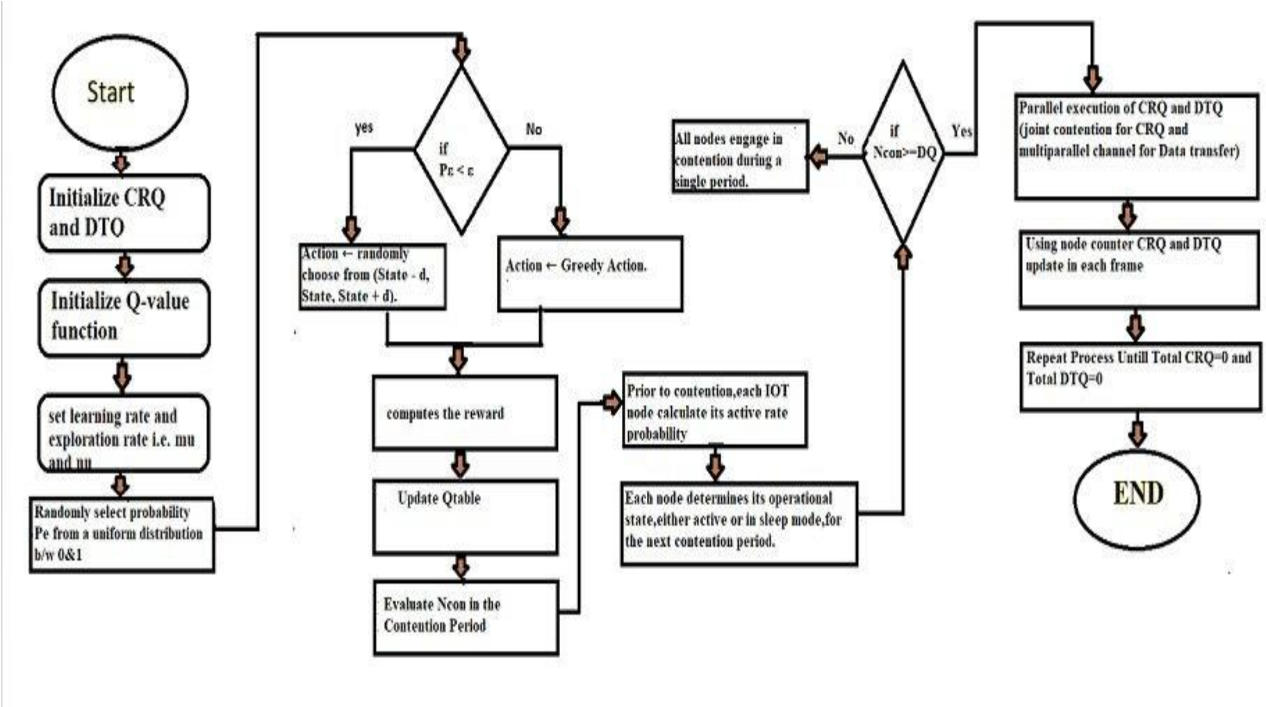


Fig.4.4. Flowchart of proposed algorithm

# 4.6. Algorithms:

### 4.6.1. Algorithm 1: EQL-DQMAC – Joint contention and parallel data transfer:

1. **Initialize CRQ and DTQ**:- Initialize the Contention Resolution Queue (CRQ) and Data Transfer Queue (DTQ) to manage contention and data transfer processes, respectively.

2. **Compute Optimal Contention Number**:- Each node computes the optimal intended contention number using algorithms(2).

3. **Evaluate Contention Number (Ncon):-** Evaluate Ncon in the contention period from Eq. (4.1).

4. **Calculate Active Rate Probability**:- Prior to contention, each IoT node calculates its active rate probability according to Eq. (4.2).

5. **Determine Operational State**:- Each node determines its operational state, either active or in sleep mode, for the next contention period based on the results of Eq. (4.2).

6. **Parallel Execution of CRQ and DTQ**:-

- ➢ If ( Ncon>=DQ condition), then execute the CRQ and DTQ processes in parallel (Joint contention of CRQ and multiparallel channel for DTQ).

- Update the node counter CRQ and DTQ in each frame.

**7. Single Contention Period**: - If Ncon < DQ condition), then all nodes engage in contention during a single period.

8. End If

### 4.6.2. Algorithm 2: Prediction of Optimal contention nodes:

1. **Initialize Q-value Function**:- Initialize the Q-value Function ( Q0(State, Action) ) at time ( t = 0 ).

2. **Assign Parameters**:- Assign values to the learning rate parameter ( mu ) and the exploration rate ( nu ).

3. **Randomly Select Probability**:- Randomly select Probability Pε from a uniform distribution between 0 and 1.

4. **Exploration or Exploitation:-**
   - if Pε < ε then:
   - Action ⟵ randomly choose from (State - d, State, State + d).
   - else if Pε ≥ ε then:
   - Action ⟵ Greedy Action.
5. **Compute Reward**:- Each node computes the reward using the specified equations (4.7).

**6. Update Q-value Function**:- Update Q-value Function ( Q(Next State, Next Action) ) from Algorithm 3.

### 4.6.3. Algorithm 3: Update Q-table:

1. **Initialize Q-Value**:- Initialize the initial Q-Value ( Q_0 ) to 1 at time ( t = 0 ).
2. **Set Learning Rate**:- Set the Learning Rate parameter (α ), where ( 0 < α < 1 ).
3. **Update Q-Value**:-
   - Compute the Q-Value for the Next $State$ ( $Q\_\{t+1\} = (1 - \alpha) * Q_t + \alpha * \{Reward\}$).
   - Update the Q-Value ( $Q_{\{t+1\}}$).

#### 4.6.4. Algorithm 4: Multiparallel channel

1. Find DTQ Length:- Determine the length of the Data Transfer Queue (DTQ), denoted by ( X ).

2. Find Available Channels:- Determine the number of available channels, denoted by ( Y ).

3. Channel Allocation:-

- If ( X > Y ):
- Allocate ( Y ) channels to data transfer.
- Else:
- Allocate ( X ) channels to data transfer.
- End if.

# CHAPTER-5

## Results and Discussion

To evaluate the performance of the proposed EQL-DQ algorithm, we conducted extensive simulations and compared it with the Q-learning-based distributed queuing MAC protocol proposed by Chien-Min Wu et al. [1] and Traditional DQMAC. The simulation parameters and scenarios were designed to emulate the typical characteristics of IoT networks.
The simulation results demonstrate the superior performance of the EQL-DQ algorithm compared to the Q-learning-based distributed queuing MAC protocol [1].

### 5.1. Simulation Parameters:

This section presents the simulation parameters for the EQL-based DQMAC protocol for an IoT network. The programming language Python and its library was used to complete the simulation.We ran our simulation programs on a windows 10 operating system (Intel Core i5-8$^{th}$ Gen) and GPU (intel r uhd graphics 620), jupyter Notebook.

The time taken for data transmission and propagation delay in wireless communication, along with energy consumption for each node, were experimentally determined. The energy consumption for transmission, reception, and sleep modes were calculated based on specifications from the radio-transceiver datasheet. According to the datasheet, the energy consumption for transmission, reception, and sleep modes are 13.5 mW, 24.75 mW, and 15 µW, respectively, with the note that reception and listening consume the same amount of energy . This information was gathered through multiple iterations to ensure accuracy.

The key differences between EQL-DQ, QL-based DQMAC, and traditional DQMAC are: EQL-DQ uses reinforcement learning (Q-learning) to dynamically optimize the number of contending nodes, incorporates a distributed queueing mechanism with joint contention mechanism for CRQ ,multiparallel channel for DTQ and employs parallel execution of crq and dtq and enhanced reward function; QL-based DQMAC also uses Q-learning to calculate the optimal number of contentions and includes a sleeping mechanism, whereas traditional DQMAC has random slot selection, no sleeping, and higher collision probability leading to increased delays and lower transmission efficiency.The channel's transmission rate was 2 Mbps, and the frame length for each contention period was 3, ensuring that the system throughput was optimized using the DQ mechanism [3].

The simulation parameters for proposed QL-based DQMAC protocol can be found in Table 2.

| Parameter | Value |
|---|---|
| SIMULATION TIME | 10000 seconds |
| TOTAL NUMBER OF IOT NODES | 100 |
| REGION SIZE | (200, 200) meters |
| CHANNEL DATA RATE | 2 Mbps |
| MINIMUM_CONTENTION_NODES | 3 |
| MAXIMUM CONTENTION NODES | 50 |
| DELTA CONTENTION NODES | 1 |
| TX POWER | 24.75 mW |
| RX POWER | 13.5 mW |
| LISTEN POWER | 13.5 mW |
| SLEEP POWER | 0.015 mW |
| DISCOUNT FACTOR | 0.9 |
| LEARNING RATE | 0.1 |
| INITIAL CONTENTION WINDOW | 10 |
| MAXIMUM DATA CHANNEL | 5 |
| MAXIMUM_ARRIVAL_RATE | 100 |

Table-5.1. Simulation parameters

## 5.2. Throughput:

In [19], the author derived a performance analysis of IEEE 802.11. The throughput of the licensed channels was analyzed in [20]. The throughput per contention period size for IoT networks owing to the simulation ending, $\zeta$ , is defined as follows:

$$\zeta = \frac{(R_{CH} * T_{suc})}{T_{simu}} \tag{5.1}$$

where Rch is the data transmission rate for the licensed channel, Tsuc is the total successful data transmission time, and the Tsimu is the system simulation time.

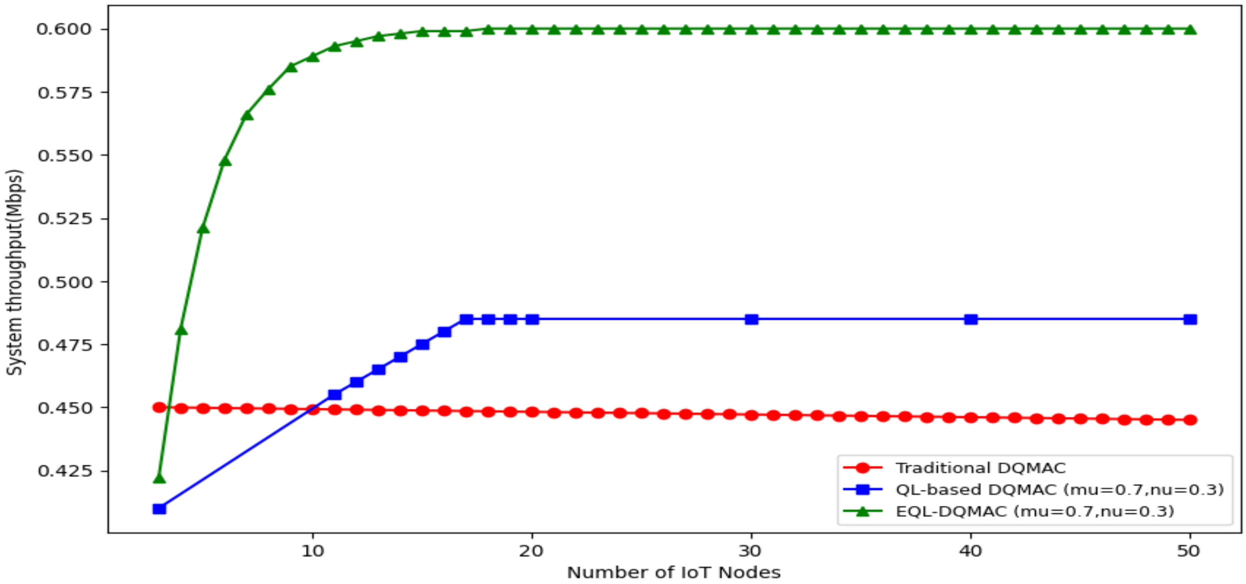


Fig.5.1. Comparison of System throughput of IOT node in traditional DQMAC,QL-based DQMAC, and EQL-DQMAC in a IOT networks

Figure 5.1. shows the Comparison of System throughput of IOT node in traditional DQMAC,QL-based DQMAC, and EQL-DQMAC in a IOT networks. The system throughput in EQL-based DQMAC ranges from 0.422 to 0.600 Mbps for μ = 0.7 and ν = 0.3. The system throughput in QL-based DQMAC ranges from 0.455 to 0.485 Mbps for μ = 0.7 and ν = 0.3. The number of contention for MAC contention in traditional DQMAC ranges from 0.445 to 0.450 Mbps. The largest improvement (improvement calculated by using equation 5.2 )for system throughput in EQL-based DQMAC compared with Q-Learning based is 23.71% and with traditional DQMAC is 33.3%.

**Reason**:-The EQL-DQMAC protocol achieves the highest throughput among the three, as its use of Q learning with enhanced reward function joint contention for CRQ in distributed queueing mechanism with multiparallel channel for DTQ effectively utilize the channel and minimize collisions, leading to superior performance.

$$Improvement\ percentage = \left(\frac{Final\ Value - Starting\ Value}{|Starting\ Value|}\right) \times 100 \qquad (5.2)$$

## 5.3. Average Contentions:

Each IoT node sends SAR in the contention period for the EQL-based DQMAC protocol until it succeeds. Each node contends only once for each contention period until the successful transmission.The average number of contentions for one IoT node to achieve a successful transmission is defined as follows:

$$n\_cp \approx (\log\_m(N\_con - 1)) + (\frac{1}{2}) + (\frac{\gamma}{\log m}) + (\frac{1}{2N_{con}*\log m}) \qquad (5.3)$$

Here, γ ≈ 0.5772 is the Euler’s constant. The value of nCP is finite, whereas m is very low. Ncon is the number of contention nodes; m is the number of contention slots for the contention period. In addition, when Ncon is low or high, the value of nCP is similar regardless of the number of contention slots, m. In the proposed EQL-based DQMAC protocol, the IoT node has two states: sleeping and active. The number of MAC contentions is the sum of the number of contention failure and one successful contention. Therefore, the number of MAC contention in the proposed EQL-based DQMAC for a node is nCP:

$$N_{cp} = N_{crq} \qquad (5.4)$$

where NCRQ is the average number of collision resolution in MAC contention period.

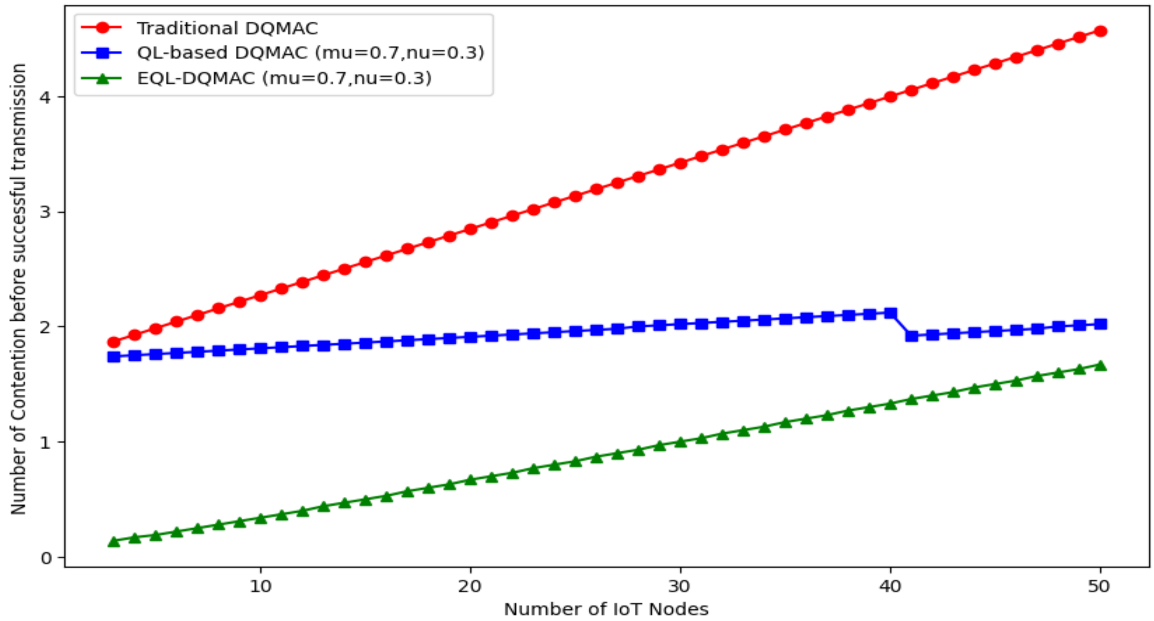

Fig.5.2. Comparison of Average contention before successful transmission of an IOT node in traditional DQMAC, QL-based DQMAC, and EQL-DQMAC

Figure 5.2 shows The level of MAC contention in traditional DQMAC, QL-based DQMAC and Enhanced Q-Learning in IoT networks. The level of MAC contention before successful transmission in EQL-based DQMAC ranges from 0.14 to 1.67 for $\mu = 0.7$ and $\nu = 0.3$. The level of MAC contention before successful transmission in QL-based DQMAC ranges from 1.74 to 2.43 for $\mu = 0.7$ and $\nu = 0.3$. The level for MAC contention before successful transmission in traditional DQMAC ranges from 1.87 to 4.57. The largest improvement in the level for MAC contention before successful transmission in EQL-based DQMAC compared with Q-Learning based is 31.27% and with traditional DQMAC is 63.45%.

**Reason**:-EQL-DQMAC maintains the lowest average contention levels by dynamically adjusting the number of contending nodes using of Q learning with parallel execution of crq and dtq , joint contention mechanism used for CRQ and multiparallel channel used for Data Transfer.

## 5.4. Average Delay:

The average delay in a network characterizes the time taken for a packet to traverse from its source to its destination. This metric is pivotal in assessing the performance of network protocols, especially in Internet-of-Things networks where timely data transmission is crucial. Average Delay (D): The average delay, denoted as (D), is defined as the mean time taken by all packets to reach their respective destinations. It is given by the equation:

$$D = \left(\frac{1}{N}\right) * \Sigma(i = 1\ to\ N)Di \tag{5.5}$$

Where:N is the number of packets.Di is the delay experienced by the i^th packet.Contention Probability (Pcoll): The contention probability, denoted as (Pcoll), is the probability of a collision occurring during the contention period. It is calculated as:

$$Pcoll = 1 - \left(1 - \left(\frac{1}{Niot}\right)\right)^{NumSarSlots - 1} \tag{5.6}$$

Where:

Niot is the number of IoT nodes.

NumSarSlots is the number of slots in the contention period.

Request-to-Send Collision Probability (Pcrq): The request-to-send collision probability, denoted as (Pcrq), is the probability of a collision occurring during the request-to-send (RTS) phase. It is calculated as:

$$Pcrq = \frac{1}{\left(1 - Pcoll * \left(1 + \mu * \frac{((Niot - 1))}{(vNiot)}\right)\right)} \tag{5.7}$$

Where:

μ is the average number of packets per IoT node.

v is the average packet arrival rate.

Time Slot Duration (τ): The time slot duration, denoted as (τ), is the duration of a single time slot in the contention period.

Time for Collision Resolution Queue (tcrq): The time taken for a collision resolution queue request, denoted as (tcrq), is the time required to handle a collision in the contention period.

Using these definitions, the average delay (D) can be derived as:

$$D = \frac{tcrq}{\tau} \tag{5.8}$$

Figure 5.3 shows the delay of MAC contention before successful transmission in DQMAC, QL-based DQMAC and Enhanced Q-Learning in IoT networks. The delay of MAC contention before successful transmission in EQL-based DQMAC ranges from 0.31 to 4.25 slots for $\mu = 0.7$ and $\nu = 0.3$. The delay of MAC contention before successful transmission in QL-based DQMAC ranges from 3.16 to 6.30 slots for $\mu = 0.7$ and $\nu = 0.3$.The delay of MAC contention in traditional DQMAC ranges from 4.62 to 12.72 slots. The largest improvement in the delay of MAC contention before successful transmission in EQL-based DQMAC compared with Q-Learning based is 32.54 % and with traditional DQMAC is 66.58%.

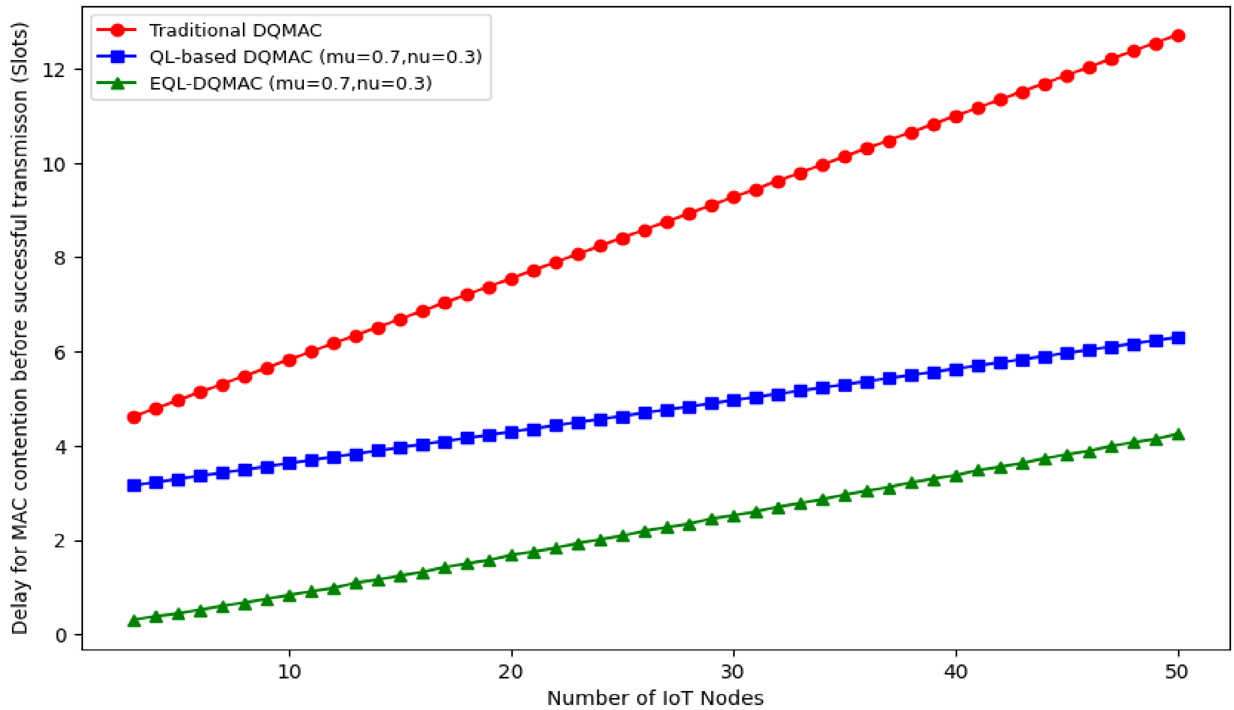


Fig.5.3. Comparison of MAC contention delay before success transmission of IOT node in Traditional DQMAC, QL-based DQMAC, and EQL-DQMAC

**Reason**:-EQL-DQMAC exhibits the lowest delay, due to dynamically adjusting the number of contending nodes using of Q learning with enhanced reward function , joint contention in distributed queueing mechanism and multiparall channel used for data transfer .

## 5.5. Energy Consumption:

The average energy consumption for an IoT node for MAC contention in a EQL-based DQMAC can be expressed as follows:

$$\overline{E} = \overline{E}_{\text{CRQ}} + \epsilon_{\text{sleep}} \overline{T}_{\text{sleep}} \tag{5.9}$$

here ECRQ is the average consumption during the collision resolution in the contention period, £sleep is the power consumption in the sleeping mode, and Tsleep is the average time in the sleeping mode. ECRQ can be expressed as follows:

$$E_{CRQ} = n_{cp} * E_{SAR} + n_{listen} * \varepsilon_{listen} * T_{listen} \tag{5.10}$$

The average energy consumption $E\_SAR$ occurs when an IoT node sends one SAR control frame and contends with other IoT nodes until it successfully transmits. ( $T_{\{\{listen\}\}})$represents the average time spent in listening mode.During the contention period, each node sends the SAR control frame in a chosen mini-slot, while it remains in listening mode during other mini-slots. Following the SAR period, each node receives one SAC control frame broadcasted by the cluster head. The expression for $E_{\{SAR\}}$ is as follows:

$$E_{\{SAR\}} = \left(e_{\{tx\}} + (m - 1)e_{\{listen\}}\right)T_{\{SAR\}} + e_{\{rx\}T_{\{SAC\}}} \quad (5.11)$$

where etx , etxlisten , and erx are the power consumption in the transmission, listening and reception modes, respectively; TSAR is the average time required to send one SAR con trol frame to a contention slot; and TSAC is the average time required to send one SAC control frame by a cluster head in a contention slot.

$$E = E_{CRQ} + \varepsilon_{sleep}T_{sleep} = n_{cp}E_{SAR} + n_{listen}\varepsilon_{listen}T_{slot} + \varepsilon_{sleep}T_{sleep} = n_{cp}\big((\varepsilon_{tx} + (m - 1)\varepsilon_{listen})T_{SAR} + \varepsilon_{rx}T_{SAC}\big)n_{listen}\varepsilon_{listen}T_{slot} + \varepsilon_{sleep}T_{sleep} \quad (5.12)$$

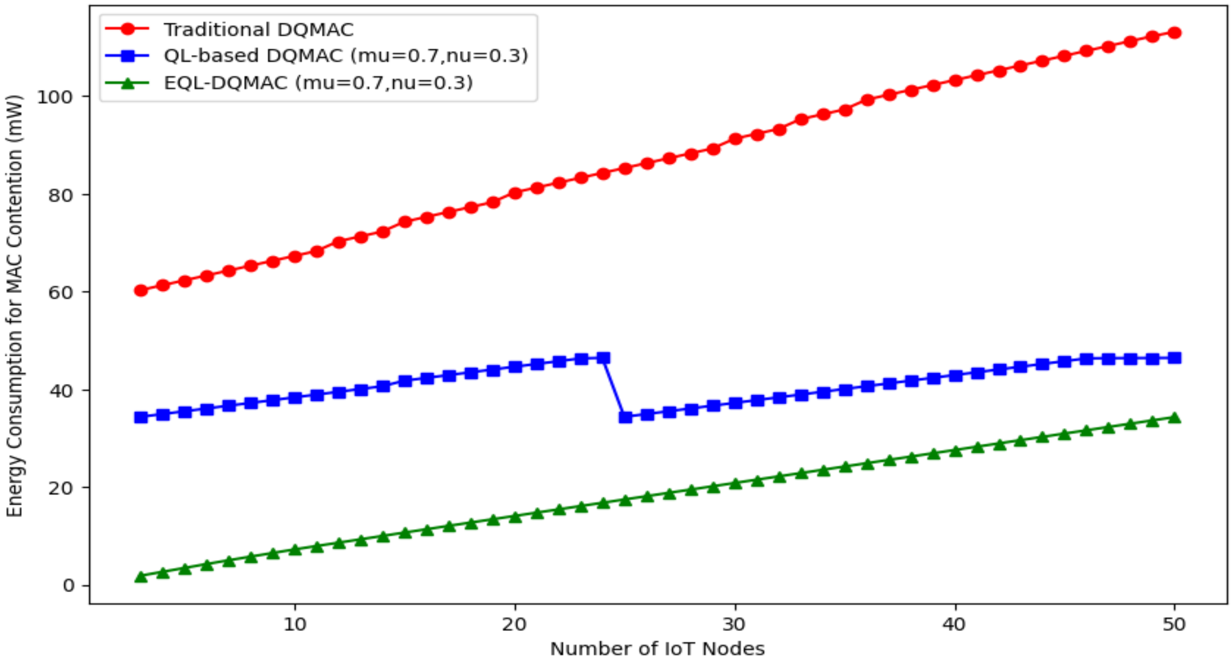


Fig. 5.4. compares the energy consumption for MAC contention in traditional DQMAC, QL-based DQMAC, and EQL-DQMAC in an IOT network.

Figure 5.4 shows the energy consumption for MAC contention in traditional DQMAC, QL-based DQMAC and Enhanced Q-Learning based in IoT networks. The energy consumption for MAC contention in QL-based DQMAC ranges from 1.83 to 34.31mW for $\mu = 0.7$ and $\nu = 0.3$. The energy consumption for MAC contention in QL-based DQMAC ranges from 34.35 to 46.44 mW for $\mu = 0.7$ and $\nu = 0.3$. The energy consumption for MAC contention in traditional DQMAC ranges from 60.29 to 113.19 mW. The largest improvement in energy consumption for MAC contention in EQL-based DQMAC compared with QL-based DQMAC is 26.12 % and with traditional DQMAC is 69.68%

**Reason**:- The energy-efficient design of EQL-DQMAC, incorporating reinforcement learning, join contention distributed queueing,enhanced reward function and parallel processing, results in the lowest energy consumption during the MAC contention process compared to the other protocols.An IoT node will enter the sleeping mode after successful transmission for EQL based DQMAC. The IoT node, after successful transmission, will also enter the sleeping mode for traditional DQMAC.

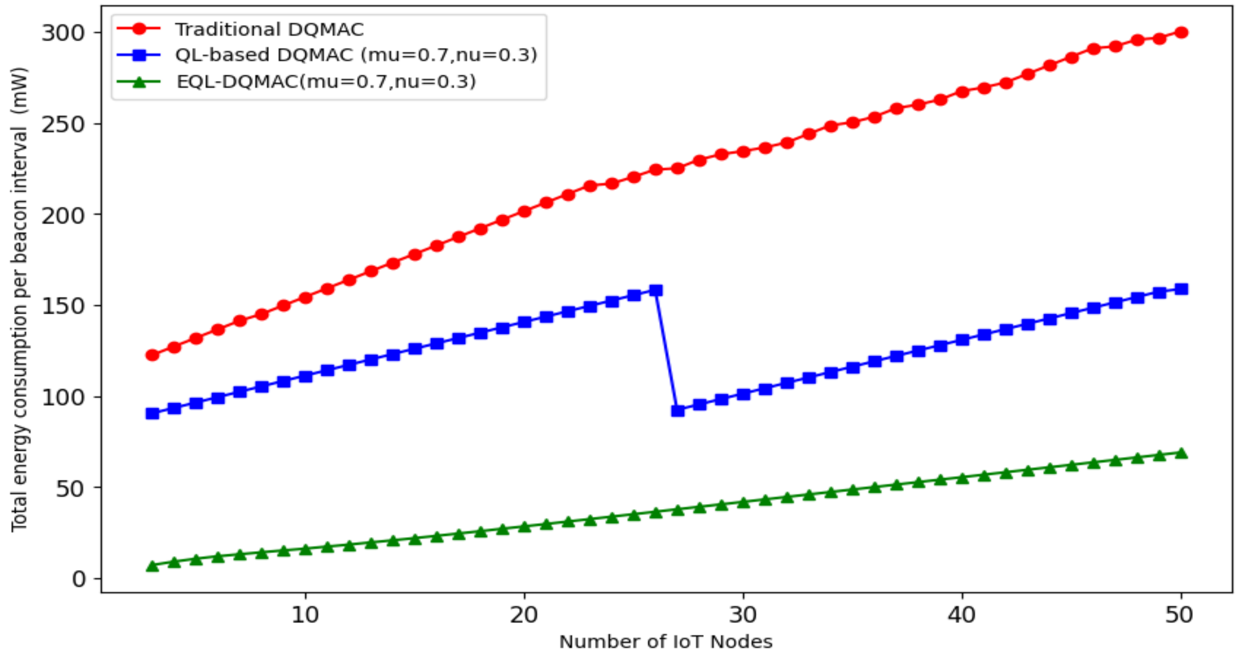


Fig.5.5 Compares the total energy consumption per beacon interval in traditional DQMAC QL-based DQMAC and EQL- DQMAC in an IoT network

Figure 5.5 shows the total energy consumption per beacon interval in traditional DQMAC ,QL-based DQMAC and EQL-DQMAC in an IoT network. The total energy consumption per beacon interval in EQL-based DQMAC ranges from 7.07 to 68.99 mW for $\mu = 0.7$ and $\nu = 0.3$. The total energy consumption per beacon interval in QL-based DQMAC ranges from 90.49 to 158.76 mW for $\mu = 0.7$ and $\nu = 0.3$. The total energy consumption per beacon interval in traditional DQMAC ranges from 122.46 to 300.22 mW. The largest improvement in the total energy consumption per beacon interval in EQL-based DQMAC compared with QL-based DQMAC is 56.54% and with traditional DQMAC is 77%.

**Reason**:-The superior energy efficiency of EQL-DQMAC, achieved through its comprehensive optimization techniques, translates to the lowest total energy consumption of IoT nodes, making it a more suitable choice for battery-powered IoT devices with strict energy requirements.

# Chapter-6

# Conclusion and Future Work

## 6.1 Conclusion:

The EQL-DQ algorithm proposed in this paper leverages the power of Q-learning joint contention for CRQ and multiparallel channel for data transfer in distributed queueing, to enhance the performance of IoT networks. By combining these techniques, the algorithm aims to optimize system throughput, minimize contention delays, and improve overall efficiency in IoT communication. Through simulations and analysis, we have demonstrated the effectiveness of the EQL-DQ algorithm in addressing the unique challenges faced by IoT networks, paving the way for more efficient and reliable IoT communication systems.

The EQL-DQ algorithm introduces several key contributions to enhance the performance of IoT networks:

i. **Adaptive Adjustment of Contending Nodes**:- By employing a Q-learning-based approach, the algorithm dynamically adjusts the number of contending nodes in each contention period. This adaptive mechanism optimizes contention resolution, leading to improved overall system performance.

ii. **Joint Contention for CRQ and Multiparallel Channel for Data Transfer**:- The algorithm incorporates joint contention for the contention resolution queue (CRQ), allowing for efficient bulk collision resolution. Additionally, it utilizes a multiparallel channel model for data transfer, enabling concurrent transmission over multiple channels to enhance data transfer capabilities.

iii. **Parallel Execution of CRQ and DTQ**:- To expedite the simulation process and improve efficiency, the algorithm facilitates parallel execution of the contention resolution queue (CRQ) and the data transfer queue (DTQ). This parallel processing approach optimizes resource utilization and enhances simulation performance.

iv. **Enhanced Reward Function**:- The algorithm introduces an enhanced reward function (Equation 4.7) that considers various factors such as average contentions normalized by contention slots, average contentions normalized by the total number of contending nodes, and average delay experienced by packets. This reward function provides a comprehensive measure of system performance, guiding the learning process effectively.

The extensive simulations conducted in our study conclusively demonstrated the superiority of the EQL-DQ algorithm over existing Q-learning-based distributed queuing MAC protocols across various performance metrics. Here's a summary of the key findings:

i. **Throughput**:- The EQL-DQ algorithm exhibited significantly higher throughput compared to existing protocols. This improvement in throughput indicates enhanced data transfer rates and overall network efficiency, allowing IoT devices to transmit data more effectively.

ii. **Average Contentions**:- Our simulations revealed that the EQL-DQ algorithm effectively reduced the average number of contentions during contention periods. By dynamically adjusting the number of contending nodes and optimizing contention resolution, the algorithm minimized collisions and contention-related delays.

iii. **Average Delay**:- The EQL-DQ algorithm demonstrated reduced average delay experienced by packets within the network. This reduction in delay indicates improved responsiveness and faster data transmission, contributing to enhanced quality of service (QoS) for IoT applications.

iv. **Energy Consumption**:- Our results indicated that the EQL-DQ algorithm achieved lower energy consumption compared to existing protocols. By optimizing contention resolution and data transfer processes, the algorithm minimized unnecessary energy expenditure, prolonging the battery life of IoT devices and improving overall energy efficiency.

The EQL-DQ algorithm's ability to adapt to the dynamic network conditions makes it a promising approach for addressing the challenges of IoT networks.

## 6.2 Future Work:

These future research directions provide valuable insights into further enhancing the EQL-DQ algorithm and its applicability in real-world IoT deployments:

i. **Incorporating Deep Reinforcement Learning (DRL**):- Deep reinforcement learning techniques can offer enhanced adaptability and decision-making capabilities. Exploring the integration of DRL with the EQL-DQ algorithm could lead to even more sophisticated and autonomous IoT systems.

ii. **Integration with Other IoT Protocols and Technologies**:- Combining the EQL-DQ algorithm with emerging IoT protocols like network slicing and edge computing could lead to synergistic effects, enabling more comprehensive optimization of IoT system performance across various dimensions such as latency, reliability, and resource utilization.

iii. **Evaluation on Real-World IoT Testbeds**:- Testing the EQL-DQ algorithm in real-world IoT environments is crucial for validating its performance, scalability, and practicality. Real-world evaluations can provide valuable insights into the algorithm's behavior under diverse conditions and help identify areas for improvement.

iv. **Enhanced Security Measures**:- As IoT networks continue to grow, ensuring robust security measures becomes increasingly important. Investigating additional security mechanisms to complement the EQL-DQ algorithm can help mitigate potential threats and vulnerabilities, enhancing the overall resilience of IoT deployments.

# CHAPTER-7

## REFERENCES

[1] C.-M. Wu et al., "A Q-learning-based distributed queuing MAC protocol for Internet-of-Things networks," EURASIP Journal on Wireless Communications and Networking, vol. 2023, no. 77, 2023.

[2] J. H. Huh, D. Tanjung, D.-H. Kim, S. Byeon, and J.-D. Kim, "Improvement of Multichannel LoRa Networks Based on Distributed Joint Queueing," in IEEE Internet of Things Journal, vol. 9, no. 6, pp. 4343, March 15, 2022

[3] W. Wu, Y. Li, Y. Zhang, B. Wang, and W. Wang, "Distributed queueing-based random access protocol for LoRa networks," IEEE Internet Things J., vol. 7, no. 1, pp. 763–772, 2020.

[4] T.-H. Yu and C.-H. Liao, "Asynchronous Duty-Cycling in Wireless Sensor Networks: An Overview and Analysis," in Proc. IEEE Int. Conf. Commun., Shanghai, China, May 2019, pp. 1-6.

[5] J. Choi, "Multichannel ALOHA with Exploration Phase," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, May 25-28, 2020, Seoul, Korea,DOI:10.1109/WCNC45663.2020.9120787.

[6] S. A. Gopalan and J. Park, "Energy-efficient MAC protocols for wireless body area networks: Survey," in 2010 Int. Congr. Ultra Mod. Telecommun. Control Syst. Workshops (ICUMT), 2010, pp. 739–744, doi: 10.1109/ICUMT.2010.5676554.

[7] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in Proc. Twenty-First Annu. Joint Conf. IEEE Comput. Commun. Societies, 2002, vol. 3, pp. 1567–1576.

[8] Y. Ye, R. Priya, and A. Rachedi, "The Internet of Everything: A survey," in 2021 13th Int. Conf. Comput. Intell. Commun. Netw. (CICN), 2021, pp. 28–33.

[9] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, ALOHA!," IEEE Access, vol. 4, pp. 2029–2044, 2016.

[10] D. Ye and M. Zhang, "A Self-Adaptive Sleep/Wake-up Scheduling Approach for Wireless Sensor Networks," in IEEE Transactions on Cybernetics, vol. 48, no. 3, pp. 979-992, March 2018, doi: 10.1109/TCYB.2017.2706184.

[11] H. Jiang, R. Gui, L. Chen, Z. Wu, J. Dang, and J. Zhou, "An improved SARSA(λ) reinforcement learning algorithm for wireless communication systems," IEEE Access, vol. 7, pp. 115418–115427, 2019.

[12] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," IEEE J. Sel. Areas Commun., vol. 37, no. 6, pp. 1277–1290, 2019.

[13] L. Jianjun, L. Lu, and W. Ying, "QoS-oriented media access control using reinforcement learning for next-generation WLANs," Comput. Netw., vol. 24, pp. 1–12, 2022.

[14] M. Grześ and D. Kudenko, "Online learning of shaping rewards in reinforcement learning," Neural Netw., vol. 23, pp. 699–705, 2010, doi: 10.1016/j.neunet.2010.01.001.

[15] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998.

[16] Z. Liu and I. Elhanany, "RL-MAC: a QoS-aware reinforcement learning based MAC protocol for wireless sensor networks," in Proc. IEEE Int. Conf. Netw., Sens. Contr., Ft. Lauderdale, FL, USA, Apr. 2006, pp. 70-77.

[17] W. Xu and G. Campbell, "A distributed queuing random access protocol for a broadcast channel," in Proc. ACM SIGCOMM Conf., San Francisco, CA, USA, Sep. 1993, pp. 270–278

[18] A. Pressas, Z. Sheng, D. Ali, F. Tian, and M. Nekovee, "Contention-based learning MAC protocol for broadcast vehicle-to-vehicle communication," in Proc. IEEE Veh. Netw. Conf. (VNC), Torino, Italy, Nov. 2017, pp. 109-117.

[19] G. Giuseppe, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. Sel. Areas Commun., vol. 18, no. 3, pp. 535–547, 2000.

[20] Z. Sadreddini, O. Makul, T. Çavdar, and F. B. Günay, "Performance analysis of licensed shared access based secondary users activity on cognitive radio networks," in Proc. Electr. Electron., Comput. Sci., Biomed. Eng.'s Meet. (EBBT), Istanbul, Turkey, Apr. 2018, pp. 1-5.

[21] K. H. Lee and D. Kim, "Cfx: contention-free channel access for IEEE 802.11ax," Sens. Netw., vol. 22, no. 33, pp. 1–20, 2022.

[22] A.-T. H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "A Comprehensive Distributed Queue-based Random Access Framework for mMTC in LTE/LTE-A Networks with Mixed-Type Traffic," IEEE Trans. Veh. Technol., vol. 67, no. 9, pp. 8305-8322, Sep. 2018.

[23] H. Hassen, S. Meherzi, and Z. Ben Jemaa, "Improved Exploration Strategy for Q-Learning Based Multipath Routing in SDN Network," J. Computer Networks Commun., vol. 2024, article no. 25, pp. 1-18, May 2024.

[24] R. Ali, Y.A. Qadri, Y.B. Zikria, T. Umer, B.S. Kim, S.W. Kim, "Q-learning-enabled

channel access in next-generation dense wireless networks for iot-based ehealth systems," EURASIP J. Wirel. Commun. Netw., vol. 2019, no. 178, pp. 1–12, Jul. 2019.

[25] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in Proc. Twenty-First Annu. Joint Conf. IEEE Comput. Commun. Societies, New York, NY, USA, Jun. 2002, vol. 3, pp. 1567-1576.

[26] J. Zhang, Z. Hu, Y. Xiong, and G. Ning, "A Collision-Free Hybrid MAC Protocol Based on Pipeline Parallel Transmission for Distributed Multi-Channel Underwater Acoustic Networks," Electronics, vol. 9, no. 4, p. 679, Apr. 2020.

[27] F. H. Panahi, S. Heshmati, and T. Ohtsuki, "Optimal Sleep & Wakeup Mechanism for Green Internet of Things," in Proc. Iranian Conf. Electr. Eng. (ICEE), Tehran, Iran, Apr. 2019, pp. 1-6, doi: 10.1109/IranianCEE.2019.8786620.

[28] R. Jain, "Comparative Analysis of Contention-Based and TDMA-Based MAC Protocols for Wireless Sensor Networks," in Proc. Int. Conf. on Wireless Commun., Ahmedabad, India, Mar. 2020, pp. 123-130.

[29] O. Abedi and A. Pourhasani, "Prioritized Multi-Channel MAC Protocol in Ad Hoc Networks Using a TDMA/CSMA Approach," Wireless Netw., vol. 27, pp. 2629-2640, Aug. 2021.

[30] L.-f. Qian, S.-l. Zhang, and M.-q. Liu, "A Slotted Floor Acquisition Multiple Access Based MAC Protocol for Underwater Acoustic Networks with RTS Competition," Front. Inf. Technol. Electron. Eng., vol. 16, no. 3, pp. 217-226, Mar. 2015.

[31] A. Bellouch, A. Zaaloul, and A. Haqiq, "Random Access Mechanism Enhancement Based on a Hybrid ALOHA Protocol Using an Analytical Model," IEEE Trans. Mob. Comput., vol. 21, no. 10, pp. 3643-3656, Oct. 2022

[32] S. Ben Alaya, M.T. Missaoui, and M. Frikha, "Enhanced Extended Access Barring Mechanism for Massive Machine-Type Communications in 5G Context," in Proc. Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM), Split, Croatia, Sep. 2018, pp. 1-6.

[33]H.-J. Lin and G. Campbell, "PDQRAP-prioritized distributed queueing random access protocol," in Proc. Conf. Local Comput. Netw., Tampa, FL, USA, Oct. 2002, pp. 163-172, doi: 10.1109/LCN.1994.386612

[34]Althobaiti, A. S., & Abdullah, M. (2015). Medium Access Control Protocols for Wireless Sensor Networks Classifications and Cross-Layering. College of Computing and Information Technology, Taif University, KSA. Retrieved from Faculty of Computing and Information Technology FCIT, King Abdulaziz University KAU, Jeddah, KSA