Exploring the Challenges and Opportunities of Edge Computing in Enhancing Data Processing Speed and Responsiveness in IoT Networks.

Dr. Sateesh Kumar Peddoju, Patel Jainil Subhashkumar, Ghandat Manas Sanjay, Tanmay Bakshi, Manan Garg, Sumit Kumar, Shaurya Priyadarshi, Gyanendra Kumar Banjare, Ahsen Kamal, Gaurav R Kochar

***Abstract*—The project provides a detailed exploration of the challenges and opportunities of Edge Computing in enhancing data processing speed and responsiveness in IoT networks. The comparison of performance and latency in IoT networks with and without Edge Computing will provide valuable insights into the advantages of Edge Computing in IoT applications. The simulation of IoT devices and a cloud IoT network will serve as a practical demonstration of these concepts. Despite the challenges, the benefits of Edge Computing make it a promising solution for enhancing the performance of IoT networks**

I. Introduction

Advances in Internet of Things (IoT) have a transformative impact on society and the environment through a multitude of application areas, including smart homes, smart agriculture, manufacturing, and healthcare. To achieve this, an ever-increasing number of heterogeneous IoT devices are continuously being networked to support real-time monitoring and actuation across different domains. Cisco predicts that 50 billion IoT devices are going to be connected by 2020.

Traditionally, the enormous amount of data, generally known as the big data, are sent to the cloud by IoT devices for further processing and analysis. However, the centralized processing in cloud is not suitable for numerous IoT applications due to the following reasons: (i) some applications require close coupling between request and response (ii) Delay incurred by the centralized cloud-based deployment is unacceptable for many latency-sensitive applications (iii) there is a higher chance of network failure and data loss, and (iv) sending all the data to cloud may drain the battery of the IoT device at a faster rate. To address the above issues, numerous concepts have been proposed, which offer cloud-like resources near to the edge of the network.

Edge computing is a promising technology that brings data processing and storage closer to the source of data, which is especially beneficial for Internet of Things (IoT) networks. This report explores the challenges and opportunities of edge computing in enhancing data processing speed and responsiveness in IoT networks.[7]

II. Edge Computing in IoT Networks

Edge computing addresses several issues related to data processing in IoT networks. Traditional cloud computing models often struggle with latency, bandwidth, and privacy issues due to the centralization of data processing and storage. Edge computing mitigates these issues by processing data at the edge of the network, closer to where it's generated. This results in improved data processing speed and responsiveness, which are critical for real-time applications in IoT networks.

At the edge, the things can not only request service and content from the cloud but also perform the computing tasks from the cloud. Edge can perform computing offloading, data storage, caching and processing, as well as distribute request and delivery service from cloud to user. With those jobs in the network, the edge itself needs to be well designed to meet the requirement efficiently in service such as reliability, security, and privacy protection.

Edge computing, in itself, is a very flexible concept. Any device in the network can act as an Edge Device. There are certain tradeoffs associated with designing edge-enabled networks. First tradeoff - Large edge-computing data center to cater to a distributed set of IoT devices (like in a city) vs Groups of devices with reduced computing power distributed across small physical spaces with limited power and cooling supply. Our implementation is flexible as it can be adapted to large-sized IoT networks in a hierarchical manner with minimal overhead and complexity.

III. Challenges in Edge Computing

Despite the advantages, implementing edge computing in IoT networks is not without its challenges:

- **Scalability:** As the number of IoT devices increases, managing and processing the large volume of data generated can be challenging.
- **Security:** Edge devices can be more vulnerable to attacks due to their exposure in the network. Ensuring robust security measures are in place is crucial.
- **Interoperability:** With various types of devices and platforms in an IoT network, ensuring they can effectively communicate and work together can be difficult.
- **Resource Constraints:** Edge devices often have limited processing power and storage capacity, which can limit the complexity of the tasks they can handle.

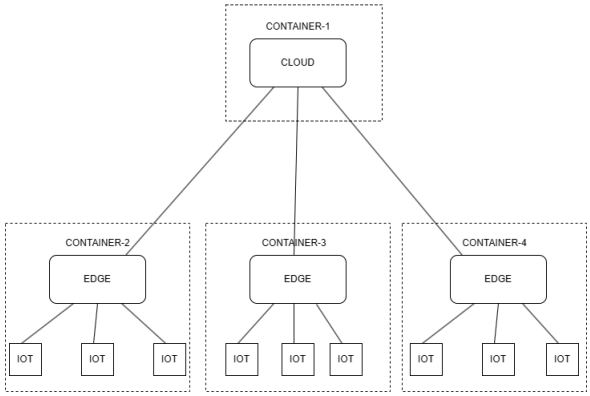

Fig. 1: Architecture for simulating IoT devices and Cloud

IV. Opportunities in Edge Computing

Despite these challenges, edge computing presents several opportunities:

- **Improved Performance:** By processing data closer to its source, edge computing reduces latency, resulting in faster response times and improved performance for real-time applications.
- **Efficient Bandwidth Utilization:** With data processing and filtering done at the edge, only useful data is sent to the cloud, reducing the amount of data transmitted and thus saving bandwidth.
- **Enhanced Privacy and Security:** Edge computing can improve privacy by keeping sensitive data at the edge of the network instead of sending it to the cloud. Additionally, edge computing can provide localized security measures tailored to the specific needs of each edge device.
- **Enabling New Applications:** Edge computing opens up possibilities for new applications that require real-time processing and decision-making capabilities, such as autonomous vehicles, smart cities, and telemedicine.

V. Project Architecture

The project architecture consists of a cloud (Container-1) that is connected to three branches of Edge and each Edge has three other branches with IoT. Each Edge with three IoTs are different Containers like Container-2, 3, and 4.

a. Container-1 simulates the cloud
b. Containers 2, 3, 4 simulate Edge and IoT devices. These are Docker containers.
c. IoT devices are represented by 2-3 different types of containers that simulate different IoT devices like a camera, Heat sensor, etc. They generate sensory data.
d. Edges are simulated by another container or a process on the same laptop. They filter out data, perform local computation, and then send data to the server.

VI. IoT Protocols Implemented

A. *Constrained Application Protocol (CoAP)*

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol designed for resource-constrained IoT (Internet of Things) devices [8]. It's specifically developed to enable simple communication in low-power, low-bandwidth environments where devices may have limited processing power and memory.

1) Key Features:

- **Lightweight Design:** CoAP's architecture is tailored to accommodate the constraints of IoT devices, offering a lightweight protocol that minimizes overhead, making it suitable for low-power and low-bandwidth scenarios.
- **RESTful Interactions:** Embracing the principles of REST, CoAP simplifies interactions by employing familiar HTTP-like methods such as GET, POST, PUT, and DELETE, enabling straightforward communication with resources.
- **Efficiency and Optimization:** Utilizing UDP (User Datagram Protocol) for communication, CoAP optimizes data transfer while supporting multicast for effective dissemination of information across multiple recipients simultaneously.

2) Implementation Overview:

- **Message Parsing:** Explain the process involved in parsing incoming CoAP messages. Describe how the server extracts header information, decodes message elements like version, type, method, message ID, URI, and payload.
- **Response Handling:** Detail the server's response generation mechanism based on the received CoAP message. Discuss the conditions and logic implemented to handle different CoAP methods (GET, POST, PUT, DELETE) and their respective responses (success, failure, or error codes).
- **Interactions with Cloud:** Elaborate on how the CoAP server interacts with the cloud. Describe the structure of the cloudPayload, the transmission process, and the reception of responses from the cloud. Highlight the time taken for communication with the cloud and its significance in the server's operations.

CoAP stands out as an optimal protocol for IoT due to its tailored design for resource-constrained devices. Its lightweight nature minimizes overhead, making it suitable for low-power scenarios. By embracing REST principles, CoAP simplifies interactions with familiar HTTP-like methods, ensuring ease of communication. Efficiency-driven features like UDP utilization and multicast support further enhance its capabilities, enabling synchronized and effective data exchange. CoAP's seamless interaction with the cloud adds to its prowess, facilitating timely and reliable communication crucial for IoT applications.

B. *HaLow Protocol*

[3] HaLow, also known as 802.11ah, is a specialized wireless communication protocol designed explicitly for the Internet of Things (IoT) and low-power, wide-area networks (LPWANs). It operates in the unlicensed sub-1 GHz spectrum, providing extended coverage range and better penetration through walls and barriers compared to traditional Wi-Fi protocols. This protocol aims to address the connectivity requirements of IoT devices by offering increased range, improved power efficiency, and reliable communication, making it suitable for various IoT applications.

1) Key Features:

- **Low-Power Operation:** HaLow ensures energy-efficient communication, making it ideal for battery-operated IoT devices that require long-term operation.
- **Extended Coverage:** Operating in the sub-1 GHz spectrum allows HaLow to offer expanded coverage, enabling communication across larger areas.
- **Improved Penetration:** HaLow's ability to penetrate obstacles and barriers enhances connectivity in environments where signal obstruction is common.
- **Reliable Connectivity:** The protocol ensures reliable connectivity for a diverse range of IoT devices, contributing to seamless communication within IoT networks.

2) Implementation Overview:

- **Client Implementation:** Simulates multiple clients sending and receiving messages using HaLow protocol.
- **State Management:** Manages the state of client connections, including authentication, message routing, and interaction with the cloud.
- **Cloud Interaction:** Facilitates communication between clients and the cloud by sending and receiving data, enhancing the protocol's capabilities.

HaLow protocol emerges as a significant advancement in IoT connectivity, offering extended range, improved penetration, and energy efficiency crucial for IoT device communication. The implementation showcased in the Python simulation demonstrates the protocol's features, including client authentication, message routing, and cloud interaction, emphasizing its suitability for diverse IoT applications, particularly in scenarios demanding extended coverage, low power consumption, and reliable connectivity.

C. *LoRaWAN Protocol*

[2] LoRaWAN (Long Range Wide Area Network) is a wireless communication protocol designed for long-range, low-power communication between IoT (Internet of Things) devices. It enables efficient communication in large-scale networks by leveraging the unlicensed radio frequency spectrum.

1) Key Features:

- **Long-Range Connectivity:** Enables communication over several kilometers, making it suitable for applications requiring wide-area coverage.
- **Low-Power Operation:** Devices can operate for extended periods on a single battery charge, reducing maintenance requirements.
- **Scalability and Flexibility:** LoRaWAN networks can handle a high density of devices, allowing scalability without compromising performance.
- **Secure Communication:** Provides encryption and authentication mechanisms to ensure secure data transmission between devices and gateways.

2) Implementation Overview:

- **Sensor Emulation:** Simulates multiple sensors sending encrypted data packets to a central gateway at regular intervals.
- **Packet Encryption:** Utilizes LoRaWAN Packet class to encrypt and decrypt sensor data using a specified encryption key.
- **Gateway Communication:** The Gateway receives encrypted sensor data, decrypts it, and forwards it to a cloud server for further processing.
- **Cloud Interaction:** The Gateway interacts with a cloud server by sending decrypted data and receiving responses, illustrating the data flow from edge devices to the cloud.

LoRaWAN Protocol serves as an efficient and reliable communication solution for IoT applications requiring long-range connectivity, low-power operation, scalability, and secure data transmission. The protocol's ability to enable long-range communication while maintaining low power consumption makes it suitable for various IoT use cases, from smart cities to industrial monitoring and agriculture, offering cost-effective and scalable connectivity solutions.

D. *Zigbee Protocol*

[1] [5] Zigbee is a wireless communication protocol. It is built on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz . The protocol is designed to be simpler and cheaper than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi, making it an ideal solution for small scale projects which require wireless connection

1) Key Features:

- **Low Cost:** Zigbee chips and modules are relatively inexpensive, making it a cost-effective solution for IoT applications.
- **Mesh Networking:** It uses a mesh network topology, which allows for devices to communicate with each other without the need for a central hub or router. This makes it ideal for use in smart home applications where devices need to communicate with each other and with a central control hub.

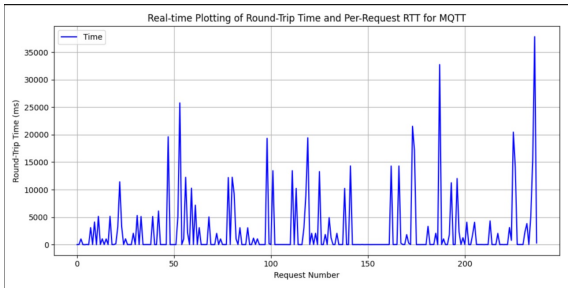

Fig. 2: Analysis for MQTT without edge

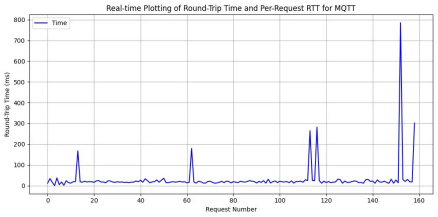

Fig. 3: Analysis for MQTT with edge

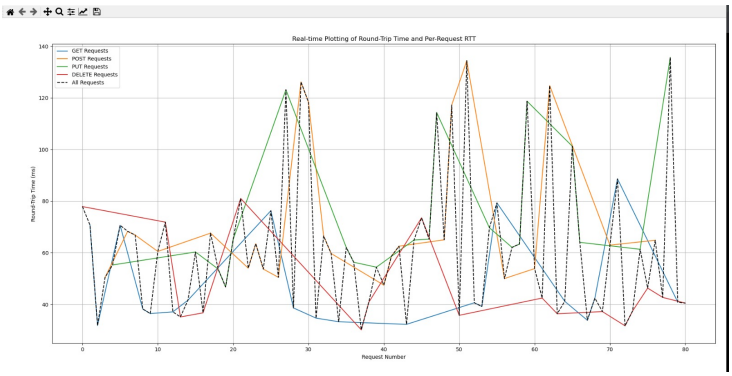

Fig. 4: Analysis for CoAP

- **Reliability:** It is designed to be highly reliable, with robust mechanisms in place to ensure that data is delivered reliably even in adverse conditions.
- **Limited Range:** Zigbee has a relatively short range compared to other wireless communications protocols, which can make it less suitable for certain types of applications or for use in large buildings.
- **Limited Data Rate:** It is designed for low-data-rate applications, which can make it less suitable for applications that require high-speed data transfer.

2) Implementation Overview:

- **Network Formation and Routing:** Zigbee network starts with a coordinator device. There are three device types: Zigbee Coordinator (ZC), Zigbee Router (ZR), and Zigbee End Device (ZED). The network uses a mesh topology that allows self-healing and automatic reconfiguration when devices are added or removed.
- **Routing in Zigbee:** The routing is managed by the network layer and is responsible for establishing, allocating addresses, and adding/removing devices.
- **Device Interactions and Communication:** Communication occurs post-association. Direct addressing uses radio address and endpoint identifier, while indirect addressing uses all relevant fields (address, endpoint, cluster, attribute). Support for group addressing enables efficient broadcast and multicast communications.
- **Security Aspects:** Zigbee ensures secure communications, key protection, and data encryption. It relies on the IEEE 802.15.4 security framework. Security is managed by a trust center device, which distributes security keys. Devices accept communications only from keys supplied by the trust center.

Zigbee supports a mesh network topology, which allows devices to communicate with each other without the need for a central hub or router. This makes it ideal for use in smart home applications where devices need to communicate with each other and with a central control hub.

E. Advanced Message Queuing Protocol

[6] [4] In the AMQP protocol, messaging involves the interaction between various components such as producers, brokers, queues, and consumers. Producers generate messages and utilize distinct routing keys for unicast routing, specifying the desired destination. Brokers, acting as intermediaries, receive these messages and determine their routes based on the provided keys. Queues, acting as storage, store the messages until they are consumed by relevant consumers. Each consumer is associated with a specific queue and uses a unique key for retrieval. This key-based unicast routing mechanism enables precise control over message flow, ensuring that each produced message reaches its intended consumer via a dedicated route, contributing to the efficiency and flexibility of communication within the AMQP-enabled messaging system.

1) Key Features:

- **Interoperability:** AMQP ensures interoperability in IoT networks by providing a standardized messaging framework. This allows diverse devices to exchange messages seamlessly, promoting a cohesive and efficient communication ecosystem across the IoT.
- **Reliability via TCP:** AMQP leverages the Transmission Control Protocol (TCP) as its transport layer, ensuring reliable and secure communication in IoT networks. By utilizing TCP, AMQP enhances data integrity through features like acknowledgments, providing a robust foundation for secure and dependable messaging.
- **Scalability:** AMQP's scalable design accommodates the dynamic nature of IoT networks. It efficiently handles varying workloads and device quantities, providing a

robust solution for growing IoT ecosystems without compromising on performance or message integrity.

2) Implementation Overview:

- **Message broker:** Listens on a TCP socket for incoming connections. Establishes an AMQP connection after receiving a key from each node post TCP connection. Routes messages using keys sent by nodes.
- **IoT Devices:** Simulated as threads. Each thread establishes a connection to the message broker. Sends messages intended for the cloud to a special node in server.py acting as a stub for both edge and non-edge computation.
- **Cloud stub:** Fixed key "key0" for connection to the exchange. Receives messages intended for the cloud from the broker. Forwards cloud messages to respective IoT nodes via the broker.

The utilization of the Advanced Message Queuing Protocol (AMQP) in IoT networks offers a robust and efficient framework for precise message routing and communication among various components. By leveraging distinct routing keys, TCP reliability, and standardized interoperability, AMQP ensures a scalable, reliable, and interoperable messaging system. The implementation details underscore how AMQP facilitates communication between IoT devices, brokers, and cloud stubs, highlighting its ability to support dynamic IoT ecosystems while maintaining efficient and reliable message flow. Ultimately, AMQP stands as a crucial protocol, providing a foundation for dependable and scalable communication within IoT networks.

REFERENCES

[1] R Prabakaran C. Muthu Ramya M Shanmugaraj. "Study on ZigBee technology". In: *2011 3rd International Conference on Electronics Computer Technology*. 2011, pp. 1–4. DOI: 10.1109/ICECTECH.2011.5942102. URL: https://www.researchgate.net/publication/261497749_Study_on_ZigBee_technology.

[2] Stephen Farrell. "Low-Power Wide Area Network (LPWAN) Overview". In: 8376. RFC Editor, May 2018. DOI: 10.17487/RFC8376. URL: https://www.rfc-editor.org/info/rfc8376.

[3] Amina Seferagi Le Tian Serena Santi. "Wi-Fi HaLow for the Internet of Things: An up-to-date survey on IEEE 802.11ah research". In: *March 2021 Journal of Network and Computer Applications*. 2021, pp. 1–43. DOI: 10.1016/j.jnca.2021.103036. URL: https://www.researchgate.net/publication/350047434_Wi-Fi_HaLow_for_the_Internet_of_Things_An_up-to-date_survey_on_IEEE_80211ah_research.

[4] Jochen Seitz Nandeesh Basavaraju Naveen Alexander. "Performance Evaluation of Advanced Message Queuing Protocol (AMQP): An Empirical Analysis of AMQP Online Message Brokers". In: *2021 International Symposium on Networks, Computers and Communications (ISNCC)*. 2021, pp. 2–6. DOI: 10.1109/ISNCC52172.2021.9615705. URL: https://ieeexplore.ieee.org/document/9615705.

[5] Feng Miao Simin Long. "Research on ZigBee wireless communication technology and its application". In: *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2019, pp. 1–4. DOI: 10.1109/IAEAC47372.2019.8997928. URL: https://ieeexplore.ieee.org/document/8997928.

[6] Steve Vinoski. "Advanced Message Queuing Protocol". In: *IEEE Internet Computing*. Vol. 10. 6. 2019, pp. 87–89. DOI: 10.1109/MIC.2006.116. URL: https://ieeexplore.ieee.org/document/4012603.

[7] Quan Zhang Weisong Shi Jie Cao. "Edge Computing: Vision and Challenges". In: *IEEE Internet of Things Journal*. 2016. URL: https://ieeexplore.ieee.org/document/7488250.

[8] C. Bormann Z. Shelby K. Hartke. "The Constrained Application Protocol(CoAP)". In: *The Constrained Application Protocol(CoAP)*. 2014. URL: https://datatracker.ietf.org/doc/html/rfc7252.