# An Overview of Internet of Things

Cooper Fitzgerald, Daniel Fischer, Kevin Perez-Espinoza, Makayla dela Cruz

***Abstract*--The Internet of Things (IoT) is the biggest shift in modern technology, the frictionless communication between devices and systems has expanded the world's technological space rapidly. This paper explores the various environments while exploring the technical details within network management, application development and maintenance. The use of Wireless Ad Hoc Networks and Wireless sensor networks will be examined, while exposing the advantages and disadvantages. Additionally, the role of message brokers in IoT systems will be illuminated, and their significance in delivering information from devices to cloud servers. Along with the challenges faced when utilizing message brokers and the corresponding solution. Furthermore, this paper will discuss Machine-to-Machine (M2M) Communication. It will address the complexity of setting up an IoT Network configured with M2M Communication and the challenges proposed with managing the M2M Network. Lastly, This paper will also investigate the two well known protocols, MQTT and CoAP. MQTT and CoAP can be coordinated in order to achieve a cohesive strong distributed system/ This paper touches on real world examples of each of these technologies while showcasing the advantages and disadvantages of each.**

***Index Terms*–internet, network, client, server, LTE, TCP, UDP, protocol**

## I. Introduction

From smart appliances to home security systems to wearable technology, the Internet of Things plays a vital role in connecting the physical world to the digital world. The Internet of Things (IoT) is an interconnected ecosystem that allows everyday devices to seamlessly communicate with each other. Moreover, this network connects various physical devices, appliances, vehicles, and other objects that use circuits, software, and network connectivity to send and receive data over the Internet. Overall, the main goal of IoT is to allow these devices to communicate effectively within a centralized system as well as facilitate automation and data analysis with minimal human intervention [1].

The Internet of Things is made up of many technologies, each facing a unique set of challenges. First, setting up an IoT network with Machine-to-Machine (M2M) communication is discussed, including what devices are needed for an M2M network and the different ways they can be configured. Also discussed are the challenges that arise with managing M2M communication networks and how those challenges are being solved.

In addition to M2M communications are Wireless Ad-Hoc Networks (WANET) and Wireless Sensor Networks (WSN). WANET and WSN are very commonly used in the context of IoT. WANET is a type of Local Area Network that allows for two or more devices to connect to each other for communication. WSN is a network of small devices that communicate with an Ad-hoc Network. Ad-hoc networks are very easy to maintain and very cost-effective. A disadvantage is how easy eavesdropping packets are.

Next, message brokers. A message broker in the context of IoT is a program or architecture that mediates communication between applications, translating messages between these applications so they can be mutually understood, while the applications themselves can remain independent of each other. These brokers are used so applications can all use the same API (Application Programming Interface), rather than needing to integrate a custom API for each application that is being used in any given project.

Finally, a general solution for challenges within the Internet of Things is two critical protocols, Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). Implementing these two protocols contributes to the general goal of IoT which includes the connectivity, scalability, and efficiency of devices over a network. MQTT is designed for unreliable, high-latency, and low-bandwidth devices. It establishes efficient communication between clients that publish and subscribe to data through a message broker. On the other hand, CoAP is a web transfer protocol that operates on a client-server model. It is used for resource-constrained environments to communicate efficiently. Overall, these two protocols are implemented for various devices depending on the services and protocols needed to meet the needs of an application.

## II. Machine to Machine (M2M) Communication

The term Machine-to-Machine (M2M) Communication, is used to describe the way two machines communicate with each other on a network or the internet. Typically, these machines are remote sensors that send data through a network to a computer that can then be accessed by a human. The types of devices that use this technology can range from manufacturing equipment, allowing someone to know how well a system is performing, or where an issue might be, all

November 27, 2023

Cooper Fitzgerald is pursuing a B.S. degree in Computer Science and Popular Music Studies from Rider University.

Daniel Fischer is pursuing a B.S. degree in Computer Science and Cybersecurity from Rider University.

Kevin Perez-Espinoza is pursuing a B. S. degree in Computer Science from Rider University.

Makayla dela Cruz is pursuing a B.S degree in Computer Science and Cybersecurity from Rider University.

the way to washing machines that tell the user when their laundry is done, and even refrigerators that tell a user when they need to buy more of a certain item.

There are three types of Internet of Things (IoT) devices that use M2M. There are consumer devices, which are usually seen as smart home devices. That includes devices like the previously mentioned washing machine and refrigerator but also includes almost any home device able to be remotely accessed and controlled. There are also enterprise IoT devices, which are meant to decrease the amount of work required in an organization. They are meant to help a lot with planning and control and are used to increase the efficiency of the organization. A good example of this is the use in the medical field. IoT devices can be used to help monitor patients and staff so that more work can be devoted elsewhere when needed. Industrial IoT has the same goal of improving efficiency, but it is applied to places like factories and industrial facilities. They can be used to help determine and tune the efficiency of manufacturing equipment, and can also help alert workers when a part of a machine needs to be serviced or replaced.

With the growing popularity of IoT devices, there is also a growing number of challenges managing the amount of data and network traffic they bring to a network.

*A. Setting up an IoT Network*

The first step in knowing how to manage an M2M network is knowing how they are set up. An M2M network requires at minimum 4 items. A sensor, a network, a computer, and software for the computer to use to process and display the sensor data. M2M communication can happen between many types of networks, wired or wireless, but for the sake of staying on topic for this paper, we are going to focus on wireless networks, since most IoT devices are wirelessly connected. There are also many different ways to outline the steps of configuring IoT devices. The ways that made the most sense to me are the steps described in [2] and [3]. For this section, my description of the steps is heavily based on those two sources.

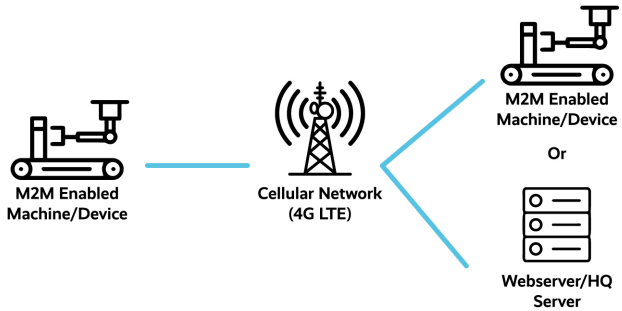


Fig. 2.1 Overview of an M2M Network [4]

*1) Registration:* The first step in managing an IoT M2M communication system is registering the devices. This involves connecting them to the internet in any way necessary. If lots of devices are being used, they may need to be connected and registered with the IoT core, which then that core connects to the network. Each device is different, but if on a secure network, the IoT devices or the IoT core's hardware Mac address will likely need to be whitelisted through the network configuration.

*2) Configuration:* The device or devices then need to be added to a group in the respective IoT software that is being used. Devices will typically be grouped based on their use or location. The configuration of devices should be saved so that in the event of a power failure or disconnection, they can automatically reconnect and reconfigure themselves. It is important to have an IoT policy for this step, that way it is clear what devices can and can't send data, and where it goes. A certificate is also important so that the devices can authenticate with each other, helping to ensure that unwanted devices aren't interfering with the IoT network.

*3) Monitoring and Collecting Data:* For data transfer, IoT devices use Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) because of their lightweight nature. They also can use Hypertext Transfer Protocol (HTTP) but because HTTP supports sending more information than MQTT and CoAP, it is usually not recommended for use in M2M communication. While the data is being monitored and transmitted, the devices themselves can be monitored to ensure that they are all working properly.

*4) Data Analysis:* Now that the data has been collected, it can be analyzed by the software of choice. Some devices have their own software, but can also work with IoT management software. Consumers can use apps like Apple's *Home App* or Google's *Google Home*. Enterprise and Industrial applications should use more robust software like Amazon's *AWS IoT Device Management*. At this point, things should also be done about the data being collected. Users can either choose to do something manually or they can set up actions to happen when a sensor senses something specific, like turning on the air conditioning if an IoT thermostat senses the temperature is too high.

*5) Updates and Maintenance:* The easiest part to forget about this whole process is updates and maintenance. While it is easy to just set things up and leave them alone, that also will leave them open to vulnerabilities. To avoid leaving devices open to vulnerabilities, users should be checking devices for updates and replacing devices once they have reached EoL (End of Life) status. Checking for updates can also bring devices more functionality and efficiency, so there is no good reason not to.

*6) Off-Boarding:* Once devices reach EoL status, or just no longer need to be used, the Off-Boarding process should begin. They should be deregistered and unauthenticated if they are still functional, that way if someone else gets access to them they do not have access to the system.

The steps outlined above may make it seem like a complicated process to set up an IoT network with M2M, and well, it is. There are a lot of factors to consider to make sure an M2M network is configured correctly and efficiently, but following those steps will help cover all of the bases to ensure that the network is set up in the best way it can and that it will continue to function properly as long as it is in use.

*B. Challenges in Managing M2M Communication Networks*

Even when set up correctly, there are many challenges to be faced with M2M Communication, and in this section, I am going to focus on a few of those challenges. The challenges discussed in this section will be scalability, data confidentiality, and security and authentication.

*1) Scalability:* IoT devices, especially those meant to move around to different locations, use cellular technology LTE. According to [5] this technology is not sufficient for IoT M2M communication. LTE was originally designed for Human-to-Machine (H2M) and Human-to-Human (H2H) communication, which require much fewer devices to be connected. M2M communication consists of a lot more devices than LTE networks are meant to handle, and if taken too far can overload the LTE cells [5].

*2) Data Confidentiality:* In [6] the question is asked, "Does the fact that the message is generated by a machine and intended to be processed by a machine change the applicability of the principle?" Specifically, it is not clear how M2M communication applies to the European Union's e-Privacy Directive, prohibiting the gathering and storage of user data over public networks. Due to it being such a new technology, there is not much legal definition of what M2M communication is. Because of this, there is not much regulation either. This is a problem when private information is being transferred through IoT devices using M2M communication.

*3) Security and Authentication:* M2M communication is also known for not being very secure. Due to the vast amount of devices and possible configurations of them, there are lots of places for things to go wrong. As outlined in [7] there are many ways a hacker can intrude on a M2M communication network such as exploiting a vulnerability found within one of the many security systems used in IoT devices. Once someone gains access to an M2M network, they can impersonate a user, and change, view, or erase important and sensitive information to the user. Authentication is important in M2M communication to ensure that even if an attacker gains access to the network, they are unable to do damage before being detected.

*C. Solutions to M2M Network Challenges*

Researchers are currently working on solutions to the challenges mentioned in the previous section. This next section will provide an overview of how they plan to address these challenges.

*1) Scalability:* To solve the issue of scalability, [5] proposes the idea of using group-based communication. To prevent networks from being overloaded with different requests coming from different places, it makes more sense for them to use a group connection to communicate with the LTE cell towers. The way that the researchers of [5] propose this would work is based on how devices first establish a connection with the cell tower. When the first device of a certain group connects to a cell tower, it will establish how the connection is made. As more devices are connected, they will signal a device that is already connected to get information on how its connection is set up and use the same configuration. With this efficient setup, the load on LTE cell towers will be much more manageable.

*2) Data Confidentiality:* According to the researchers of [6] the confidentiality principle in the European Union's e-Privacy Directive should apply to M2M communication. This is because, just like any other form of communication online, there is an expectation of privacy. The e-Privacy Directives purpose is to protect the privacy of internet users and so it should include all types of communications that involve them and their sensitive data. Even though a human isn't always directly in communication on an M2M network, they are still involved and should be protected.

*3) Security and Authentication:* To face the issue of lack of authentication in M2M communications networks, [7] suggests using blockchain technology to store data. The blockchain is described as "a distributed network that integrates asymmetric encryption, a time stamp, and the consensus algorithm" [7]. What this means for security and authentication is that it is very challenging for an attacker to view or make changes to any user's data. There are also nodes within blockchain that store ID and certificate information for the connected devices, which makes it challenging for an attacker to act as a device on the network. These features of a blockchain network make it much more secure than a standard M2M network.

## III. Wireless Ad-Hoc Networks and Wireless sensor Networks

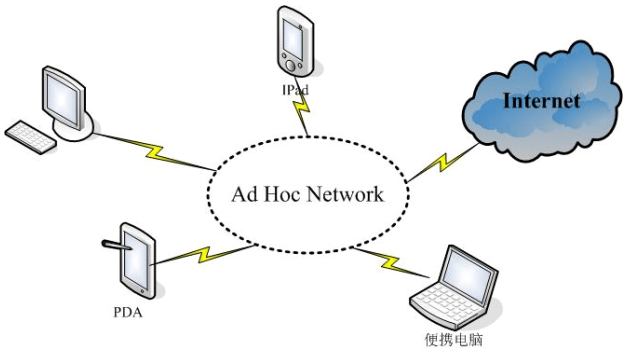


Fig. 3.1 Structure of Wireless Ad Hoc Network.

### *A. Definitions*

A Wireless Ad Hoc network also known as a WANET, is a local area network that enables two or more wireless devices to connect to each other without the need of a wireless router or access point. A wireless ad hoc network is used so devices can access each other's resources or data through a peer-to-peer network.[8] Essentially cutting out the need for a wireless router. When a wireless ad hoc network is set up there are two important things that need to be configured before it is ready for use. All devices must be set to use the same service set identifier (SSID) and they must be using the same wireless frequency channel number. An SSID is essentially a name for the connection. For example, A wifi connection in a typical home will have a name that is broadcasted and viewable by anyone who is trying to connect. [8] A wireless frequency channel number is the specific division of the network into specific frequencies. For example, an at-home network typically will have 2 frequencies, 2.4 gigahertz or 5.0 gigahertz. The difference between these two is that 2.4 GHz allows for a farther network travel at the cost of slower speeds while a 5.0 GHz network allows for faster speeds at the cost of distance.[9]

### *B. What is a Wireless Sensor Network?*

A wireless sensor network (WSN) is the transfer of data through a network that is absent of any physical cables. These sensors come in a variety of types. For example, there are sensors to read the temperature, light, speed, detect any motion and detect the current moisture in the air. A Sensor network has different parts to it. There is a base station where each sensor is connected to in order to transmit the data, and there are the actual sensor nodes that are interconnected through the base station. Each sensor node has its own processor built into the system so that it can perform tasks without relying on the base station. They also come equipped with their subset of sensors such as Sensor Sub-system, A Processing system and a Communication system. The Sensor Sub-system is used for the visual, audio and tactile values. The Processing System contains a Microcontroller, memory and its own operating system. The processing system is what receives the data and determines where the data gets sent to next and since this is a wireless system it can also send battery percentage information to the base station. Finally it contains a communication system which is used to exchange data from sensor to sensor with the use of radio transmitters.

Wireless sensor networks have a variety of different applications such as battlefields, Health Care and Internet of things. Internet of things in terms of Wireless Sensor Network is the connecting of physical devices on or over a network through the use of wireless sensors. [12]

### *C. Challenges of an Ad-hoc network*

A wireless sensor network has many challenges such as low speed communication. Cost-effectiveness, limited computational and communication resources. WSN cannot transfer data between nodes very quickly so it is really only useful for data that is not super time sensitive. For example, Reporting the weather to a website is not super time-sensitive because transmitting temperature data can be sent to a weather website with some delay. Another challenge of WSN nodes is the restriction of the battery and processor on board. The program running on the node must be very optimized in order to reduce wasted battery life and to limit the load being run which depending on hardware will slow down the data speeds. Scalability is a larger issue with WSN because the sensor nodes increase and the network might not be able to handle a huge amount of nodes which may cause a significant problem. Deployment is no trivial task. The sensor nodes have to be deployed in an optimal configuration where each node is able to transfer data efficiently. A Wireless Sensor Network also has an issue called the Travelling Salesman Problem (TSP). According to GeeksforGeeks.com, TSP is used to “find the shortest possible route that visits every city exactly once and returns to the starting point.” For example, if you have a drone transferring data from 8 nodes. The traveling salesman problem would be enacted in order to find the shortest path possible until the end. The traveling salesman problem is adapted to WSNs because it addresses the routing of data throughout the city (Nodes) in order to find the most optimal or close to optimal route from the source node to a destination node. Wireless Sensor Networks are very important in the context of the Internet of Things. They enable wireless connectivity among various different sensors and devices while being able to transmit real time information. In the event of a node failure, a WSN can redirect the data through an alternative path in order to keep continuous data flow and reliability. WSNs are integrated with various IoT platforms which creates a diverse compatibility list of devices and systems. This allows for a strong and effective IoT ecosystem. [15]

## IV. What is a Message Broker in the Context of IoT?

### *A. Definition*

A message broker in the context of IoT is a program or architecture that mediates communication between applications, translating messages between these applications so they can be mutually understood, while the applications themselves can remain independent of each other. In other words, a message broker is [16] “an intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver”. These brokers are used to validate, store, route and deliver messages between programs, acting like a post office or distribution center does in real life, receiving packages from various sources and then sending those packages out to various recipients.

### *B. How Do IoT Systems Rely on Message Brokers to Deliver Information?*

As stated in the definition, IoT systems rely on message brokers to deliver information by translating messages from different languages and platforms so that they can be

understood by both the sending and receiving applications. [17] The brokers themselves are usually modules within a messaging middleware or MOM(Message-Oriented Middleware) infrastructure. These infrastructures are used to take stress off of developers, as they are able to focus on the core logic of their applications, rather than having to write in a standardized language or manually create protocols to translate the messages themselves. Message brokers often rely on a FIFO (First In, First Out) queue to handle incoming and outgoing traffic, as messages are stored in the queue and sent out in the exact order that they were received in. Many systems in the IoT also rely on message brokers because they enable asynchronous communication. Asynchronous communication is messaging that does not require all of the involved systems to be online and connected at the same time in order to transmit and receive messages. Because of the usage of a message broker, messages are able to be stored and sent once the recipient is online and connected, as opposed to being sent out into a void if a direct connection between two systems was used, and one of the systems was not online. This is very important, as preventing data loss and ensuring security is one of the key factors of designing any communication protocol or application.

### *C. Types of Message Broker Models*

There are two basic message distribution patterns or messaging styles when it comes to message brokers: point-to-point messaging and publish/subscribe messaging. [18]Point-to-point messaging is used when each message needs to be sent exactly once, from exactly one sender to exactly one recipient. This is important for when message security and integrity is a high priority, with an example being financial transactions, since both the sender and the recipient need to know that they are the only two parties involved with the transaction, that the transaction will only happen once, and that the amount of money in the transaction will not be tampered with. Alternatively, when messages need to be sent to multiple interested parties, the publish/subscribe or “pub/sub” method is used instead. In this method of message distribution, recipients can choose to “subscribe” to any number of topics. Senders can then “publish” their messages to any of these topics, and all of the recipients that are subscribed to that topic will receive the message. This method has a one-to-many relationship between sender and receiver, whereas the point-to-point method has a one-to-one relationship between sender and receiver.

### *D. Challenges of Using a Message Broker in IoT Systems*

Message brokers, like any other program or application, are not without their drawbacks and costs. One of the primary concerns of a message broker within the IoT is scalability. As the number of devices in an IoT system increases, the message broker itself must be scaled in order to handle an exponentially increasing number of messages, without experiencing a significant increase in processing time. A similar concern is the added overhead that comes with a message broker. Whenever you add a device to a preexisting system, there is going to be unavoidable overhead time cost as messages are transmitted from one device to another, and this can be problematic if there are devices in the system that have low processing power, and performance may be greatly hindered as a result. Another primary concern is security, as a message broker is a prime target for cyberattacks since it transmits and receives data from so many different devices all at once. Because of this, message brokers need to have adequate security protocols built into them, in order to protect from otherwise devastating attacks. Message brokers must also be especially secure due to their central and integral role within a system. Since the message broker controls all of the traffic between devices, if the broker is compromised, then the entire network is put at risk, as different devices are no longer able to communicate with each other effectively. The final major potential problem with message brokers is their complexity. Message brokers by nature deal with a multitude of devices, often produced by different manufacturers, and in turn written in various coding languages. The broker must be programmed to not only be able to understand every one of these various languages, but also be able to translate each language into every other possible language. Programming these translation protocols becomes increasingly more difficult and time-consuming as more devices are added to the IoT system, and monitoring the message broker and the related traffic also becomes more complex. Regardless, the broker must be compatible with every device in the IoT system, no matter how many different manufacturers or languages there are.

## V. MQTT & CoAP

Two protocols, Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) play important roles in facilitating connectivity and communication between IoT devices. Specifically, both protocols contribute to the overall goals of IoT which are the connectivity, scalability, and efficiency of devices over a network.

Message Queue Telemetry Transport (MQTT) is a standards-based messaging protocol used for lightweight machine-to-machine communication (MCM). This protocol establishes a set of rules that defines how IoT devices communicate through publishing and subscribing to data over the internet [19]. This protocol is designed for networks that are unreliable, high-latency, and low-bandwidth devices as well as resource-constrained devices [20]. MQTT has become a standard for IoT data transmission as it provides many benefits. Firstly, it enhances connectivity within IoT ecosystems through asynchronous communication which allows for two devices to send and receive messages without a direct connection [21]. Next, MQTT is scalable because its implementation does not require a large amount of code; therefore, it consumes very little power in operations and it is able to connect with millions of IoT devices. Finally, MQTT requires minimal resources to be implemented on IoT devices, making it lightweight and efficient [22].

To continue, Constrained Application Protocol (CoAP) is a web transfer protocol that is also used for machine-to-machine (M2M) applications. It uses a client-server model based on the Representational State Transfer (REST) architectural style which means that it is resource-oriented, stateless, and implements a uniform interface. This protocol allows resource-constrained environments to communicate efficiently in the IoT ecosystem. CoAP also plays an important role in the IoT. First, it promotes intermittent connectivity as it supports asynchronous communication which allows devices with unreliable connections to exchange data. Next, its lightweight architecture allows it to handle many devices across IoT applications by enabling seamless scaling. Finally, CoAP has a minimalist design that conserves energy and resources and caters to the constraints of low-power devices [23].

Overall, MQTT and CoAP both utilize different protocols and provide different services that work to create a distributed IoT system that meets the needs of an application.

*A. MQTT Services, Protocols, and Applications*

Two services that the Message Queue Telemetry Transport (MQTT) protocol provides are publish-subscribe messaging and Quality of Service (QoS) Levels. MQTT uses principles of the publish/subscribe model where a message broker coordinates communication between devices that publish messages to topics and clients that subscribe to these topics to receive messages. First, the clients and brokers establish a connection, then the client publishes or subscribes to a topic, and the broker receives and filters messages for the clients using these topics. Then, the broker distributes the messages from the publisher to the subscribers. The publisher and subscriber do not directly exchange data as the broker decouples them by making sure these clients do not run at the same time, do not exchange IP addresses, and can send and receive messages without interrupting each other [22]. Figure 5.1 visualizes how clients publish and subscribe to topics by connecting to the MQTT broker.

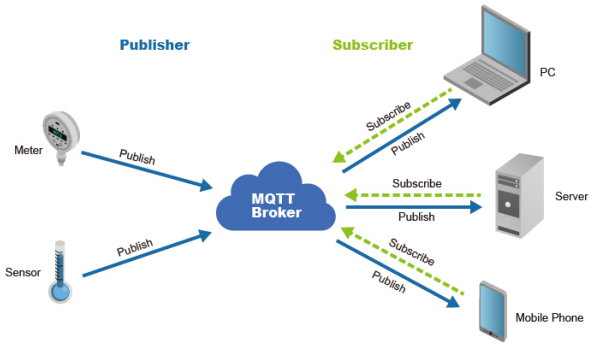


Fig. 5.1 Overview of MQTT publish-subscribe model [23]

Along with publish-subscribe messaging, MQTT also defines different levels of message delivery guarantee or Quality of Service (QoS) Levels. QoS provides the client with the ability to optimize their network usage by selecting a service level that will help them achieve a balance between efficiency and reliability. The three levels of QoS are at most once (QoS 0), at least once (QoS 1), and exactly once (QoS 2). To explain, in QoS 0, a message from the sender is not stored and the recipient does not confirm that it received the message. In QoS 1, the sender keeps a copy of the message until it receives a PUBACK packet so it is ensured to be delivered at least once to the receiver. Finally, QoS 2 is the highest level of service and it ensures that a message is delivered exactly once to the recipient through a four-part handshake. This four-part handshake is between the sender and receiver and involves a flow of PUBLISH, PUBREC, PUBREL, and PUBCOMP packets to ensure a message is delivered and the sender has received the confirmation [24].

The protocols that MQTT implements are the Transmission Control Protocol (TCP) and WebSocket (WSS). MQTT is implemented over TCP which allows for reliable and ordered communication. This reliable communication between the MQTT clients and brokers is established using TCP's connection [23]. Additionally, MQTT can also be implemented over WebSocket (WSS) which receives data directly into a web browser. With this protocol, a JavaScript client is defined to provide WSS for browsers so that additional headers are added to MQTT messages. WSS connection is bi-directional so communication is able to flow in 2 directions over the web [22].

There are many example applications of MQTT. For instance, MQTT was implemented with Facebook Messenger and Instagram messages to solve issues with limited internet bandwidth. This protocol enabled these applications to function effectively with varied internet connections. Also, MQTT is used for smart farming devices that monitor weather and soil parameters. Here, the protocol allows these devices to gather data and access the broker with a limited internet connection. Lastly, home automation devices that allow users to manage their homes through voice activation can connect to MQTT and respond to orders by sending messages to the broker. [25]

*B. CoAP Services, Protocols, and Applications*

One service that the Constrained Application Protocol (CoAP) utilizes to deliver efficient communication among IoT devices is a request-response model. This communication model is similar to HTTP where clients make CoAP requests to a server and the server responds with the requested action or data. The request is sent using two different messages, either confirmable or non-confirmable. Request methods such as GET, PUT, POST, and DELETE are used to handle the resource identification, manipulation, and exchange of data across the application layer. Overall, this method contributes to reliable communication in constrained environments [23]. Figure 5.2 visualizes how clients and servers interact with the request-response model utilized by CoAP.

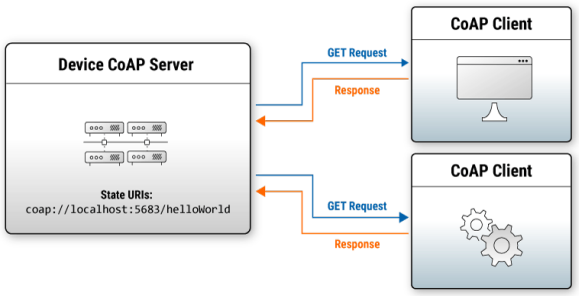


Fig. 5.2 Overview of CoAP request-response model [26]

Next, CoAP utilizes the User Datagram Protocol (UDP) as its transport protocol to move messages across a network. UDP is a connectionless and lightweight protocol that treats CoAP messages independently so before data exchange, there is no setup or teardown of connections. Since CoAP is intended to be used for constrained environments, operating over UDP allows for minimal overhead to conserve resources for devices with limited processing power. Overall, UDP works with CoAPs request-response model to enable communication between clients and servers [23].

CoAP is utilized in many real-world scenarios where communication between resource-constrained devices is needed. One example is healthcare wearables such as medical devices and fitness trackers. Devices such as these use CoAP to communicate health data to cloud servers and smartphones. Additionally, CoAP is used in smart city deployments to connect IoT devices such as waste bins. Specifically, this protocol can be used to notify collection services when the bins are full. Lastly, CoAP is implemented in industrial IoT where CoAP-enabled sensors send real-time data about machine performance to a central controller which can be analyzed for predictive maintenance [23].

## VI.CONCLUSION

In conclusion, the Internet of Things is one of the biggest innovations in modern technology, and its impact and importance in everyday life cannot be understated. The Internet of Things relies on machine-to-machine communication, which is achieved through the usage of systems like wireless ad-hoc networks and message brokers, as well as protocols like Message Queue Telemetry Transport and Constrained Application Protocol. All of these are tools that are designed to assist with communication between various devices, and they all serve as some form of middleman between said devices, as the ad-hoc networks and message brokers provide the road for the devices to communicate on, and the protocols like MQTT and CoAP serve as vehicles on that road, transporting messages throughout an IoT system.

## VII. REFERENCES

[1] P. Gokhale, O. Bhat, and S. Bhat, “Introduction to IOT,” vol. 5, no. 1, pp. 41–44, 2018.

[2] thomas, “How to Manage IoT Devices - JFrog Connect,” 15-Aug-2021. [Online]. Available: https://jfrog.com/connect/post/how-to-manage-iot-devices/. [Accessed: 19-Nov-2023]

[3] A. Takyar, “What is IoT Device Management and How it works?,” 2021. [Online]. Available: https://www.leewayhertz.com/iot-device-management/. [Accessed: 19-Nov-2023]

[4] “M2M Data Consumption and Expense Management,” 2023. [Online]. Available: https://mindglobal.com/case-studies/machine-to-machine-m2m-data-consumption-how-expensive-is-it-and-why-should-it-be-monitored. [Accessed: 26-Nov-2023]

[5] Y. Jung, D. Kim, and S. An, “Scalable group-based machine-to-machine communications in LTE-advanced networks,” vol. 25, no. 1, pp. 63–74, 2019, doi: 10.1007/s11276-017-1541-y. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=ofm&AN=134222390&site=eds-live&scope=site

[6] S. Storms, P. Valcke, and E. Kindt, “Rage against the Machine: Does Machine-to-Machine Communication Fall within the Scope of the Confidentiality Principle,” vol. 27, no. 4, pp. 372–408, 2019 [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=edshol&AN=edshol.hein.journals.ijlit27.24&site=eds-live&scope=site

[7] Y. Xie, Y. Wang, and M. Ma, “A blockchain-based authentication scheme for the machine-to-machine communication of a cyber physical system,” vol. 41, no. 4, pp. 4425–4430, 2021, doi: 10.3233/JIFS-189702. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=153410603&site=eds-live&scope=site

[8] (November,). Wireless Ad Hoc Network (WANET). Available: https://www.techtarget.com/searchmobilecomputing/definition/ad-hoc-network.

[9] “2.4 GHz vs. 5 GHz WIFI,” CenturyLink, https://www.centurylink.com/home/help/internet/wireless/which-frequency-should-you-use.html (accessed Nov. 26, 2023).

[10] *What is a Wireless Sensor Network?*. Available: https://edis.ifas.ufl.edu/publication/AE521.

[11] “Applications of ad hoc network and it’s problems,” GeeksforGeeks, https://www.geeksforgeeks.org/applications-of-ad-hoc-network-and-its-problems/ (accessed Nov. 26, 2023).

[12] N/A, “What is an SSID?,” www.kaspersky.com, https://www.kaspersky.com/resource-center/definitions/what-is-an-ssid (accessed Nov. 26, 2023)

[13] Holykell, https://www.holykell.com/news/What_are_Advantages_of_Wireless_Sensor_Network.html (accessed Nov. 26, 2023).

[14] J. E. Tito et al., “Solution of traveling salesman problem applied to wireless sensor networks (WSN) through the MST and B&B Methods,” NASA/ADS, https://ui.adsabs.harvard.edu/abs/2018SPIE10808E..2FT/abstract (accessed Nov. 26, 2023).

[15] N/A, “Traveling salesman problem using Dynamic Programming,” GeeksforGeeks, https://www.geeksforgeeks.org/travelling-salesman-problem-using-dynamic-programming/ (accessed Nov. 26, 2023).

[16]S. Hasitha, “Introduction to Message Brokers,” 2021. [Online]. Available: https://hasithas.medium.com/introduction-to-message-brokers-c4177d2a9fe3

[17]D. R. Ferriera, “Message Brokers,” 2013 [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-40796-3_4#citeas

[18]IBM, “What is a Message Broker?” [Online]. Available: https://www.ibm.com/topics/message-brokers#:~:text=Message%20brokers%20can%20validate%2C%20store,many%20of%20them%20there%20are.

[19] “Mqtt Essentials,” All Core Concepts Explained, https://www.hivemq.com/mqtt/ (accessed Nov. 24, 2023).

[20] EMQX, “What is the MQTT protocol and how does it work?,” www.emqx.com, https://www.emqx.com/en/blog/the-easiest-guide-to-getting-started-with-mqtt (accessed Nov. 24, 2023).

[21] K. Rupareliya, “Mqtt vs. COAP: An in-depth look at two leading IOT protocols,” MQTT vs. COAP: An In-Depth Look at Two Leading IoT Protocols, https://www.intuz.com/blog/mqtt-vs-coap (accessed Nov. 24, 2023).

[22] “What is MQTT?,” Amazon, https://aws.amazon.com/what-is/mqtt/#:~:text=MQTT%20is%20a%20standards%2Dbased,constrained%20network%20with%20limited%20bandwidth. (accessed Nov. 24, 2023).

[23] Radware, “What is Coap? understanding the constrained application protocol,” Radware, https://www.radware.com/security/ddos-knowledge-center/ddospedia/coap/#CoAPvsMQTT (accessed Nov. 24, 2023).

[23] “MQTT/MQTT Sparkplug,” ORing / Technology / Industrial VPN Router / MQTT/MQTT Sparkplug, https://connect.oringnet.com/en-global/tech/detail/94 (accessed Nov. 27, 2023).

[24] HiveMQ, “What is MQTT Quality of Service,” What is Mqtt Quality of service (qos) 0,1, & 2? – mqtt essentials: Part 6, https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/ (accessed Nov. 24, 2023).

[25] Z. Asim, “Applications of MQTT protocol with its benefits and comparison,” High Voltages, https://highvoltages.co/iot-internet-of-things/mqtt/applications-benefits-and-comparison-of-mqtt-protocol/ (accessed Nov. 24, 2023).

[26] I. Craggs, “Mqtt vs CoAP for IOT,” HiveMQ, https://www.hivemq.com/article/mqtt-vs-coap-for-iot/ (accessed Nov. 27, 2023).

**Cooper Fitzgerald** was born in New Brunswick, New Jersey on August 25th, 2002. He will be receiving a B.S in Computer Science with a minor in Popular Music Studies from Rider University in Lawrenceville, NJ in May 2024. He currently works as an intern at Wakefern Food Corporation in Edison, NJ, working in their POS and IT department, as well as working as a lifeguard in Princeton, NJ. He plans to continue working in IT and web development after graduation.

**Daniel Fischer** is from Howell, NJ and was born in 2001. He will be receiving a B.S. in Computer Science and Cybersecurity from Rider University in Lawrenceville, NJ in May 2024. He currently works as a Student Staff Coordinator at Rider’s IT department and plans to continue to work in IT after graduation.

He has helped develop various programming projects and has done research on developing and setting up a SIEM within Microsoft Azure. He has also received the IT Fundamentals certification from CompTIA. He is interested in working in IT because he enjoys troubleshooting and working with computer hardware. He also enjoys programming and is interested in working in software development.


**Kevin Perez-espinoza** was born in Long Branch, New Jersey, USA on November 10th, 2001. He will be receiving a B.S. degree in Computer Science from Rider University in Lawrenceville, in 2024. He is looking forward to graduation and is currently preparing for job interviews. Kevin has co-developed 2 coding projects with various uses in the medical industry. He hopes to be able to pursue a M.S. in Computer Science. He hopes to be able to teach computer science courses at Brookdale Community College.
He finds joy in playing video games and socializing with colleagues.

**Makayla dela Cruz** was born in Toronto, Ontario, Canada in 2002. They are a student at Rider University, Lawrenceville NJ, USA, and plan to graduate in 2024 with a B.S. degree in Computer Science and Cybersecurity. During their time at Rider, they have co-developed various coding projects as well as co-written a machine learning research paper. They have also received the IT Fundamentals certification from CompTIA. They hope to pursue a career in software development or web design after graduation.