ISSN:0975-9646

Internet of Things (IoT) Protocol Enabling Mechanisms

Michael Kinzel, Dr. Te-Shun Chou

College of Engineering and Technology, Department of Technology Systems, East Carolina University

E 5th St, Greenville, NC 27858, USA

kinzelm20@students.ecu.edu

CHOUT@ecu.edu

***Abstract*— The IoT protocol stack has not been standardized, but can be represented by a three, four, and five-layer abstraction. Each of these protocol stack abstractions can be understood by comparing the IoT protocol layer services to the equivalent services in the five layer internet protocol stack. Typical IoT protocols (OS4I, BLE, ZigBee, and LoRaWAN) vary slightly in the protocol layer stack architecture, but generally provide the same or similar services at each layer. However, each IoT protocol layer is unique to the specific IoT application; e.g., devices used in a LAN have different protocols than IoT devices used in a WAN. Standard IoT protocol services include range, openness, interoperability, network topology, and security. These service requirements drive protocol capabilities, such that each protocol is an enabling technology for each IoT device as well as the IoT Ecosystem in its entirety. The heterogeneity of IoT protocols have enabled the wide spread growth of the IoT.**

***Keywords*— IoT protocol, IoT protocol stack, IoT protocol layer services, IoT applications, IoT smart farming.**

I. INTRODUCTION

Intelligent applications have become pervasive within the home, industry, and society at large. Sensors and automated devices have contributed to smart homes consisting of security cameras, smart TV(s), gas sensors, smart meters, smart thermostats, etc. Smart housing grids interact with smart cities and water systems. The network of these distributed devices, sensors, and applications that can sense, interact, and control the physical world is referred to as the Internet of Things (IoT). The embedded and distributed aspect of intelligent applications within the IoT requires methods to ensure reliable and autonomous data exchange between devices. Communication protocols provide the means for this data exchange. A protocol "defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event" [1]. Standard and open protocols are defined by Requests for Comments (RFC) or by the Internet Engineering Task Force (IETF). Protocol standards are also formed by industry working groups and are usually also open. However, protocols may also be proprietary and unavailable for viewing and analysis.

The internet's most prevalent protocols include Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Protocol (IP) [1]. However, these common and open protocols are not as common within the IoT Ecosystem. Unlike traditional devices within the Information Technology (IT) domain, devices within the IoT Ecosystem are supplied by vendors and manufacturers with different technologies without agreed standards for networking protocols [2]. The resulting IoT Ecosystem has a heterogenous pool of devices, applications, and domains that have an equally heterogenous pool of protocols; each of which is designed to meet a specific IoT application. In addition to being heterogenous, IoT protocols can operate in different layers (or stacks) within the IoT Ecosystem.

There is a general knowledge gap on whether the disparate nature of IoT protocols is an enabling or potentially detrimental feature for continued IoT growth. This research paper addressed the question: does the large heterogeneity and lack of standardized protocol stacks in the IoT Ecosystem present future challenges for further integration of IoT in everyday usage? The conclusions found that due to the large variation in sensor and device technologies, diverse IoT protocols not only have enabled the growth of the IoT Ecosystem, but will continue to support greater pervasiveness into society.

This paper firstly presented a review of the commonly accepted five-layer internet protocol stack and listed common protocols for each service layer in Section 2. Section 3 introduced the IoT, generic protocol stacks used to layer IoT services, four common IoT protocols, and compared each of the four common IoT protocols based on the services within each layer. Section 4 presented a IoT case study to help conceptualize IoT protocols within real-world applications. This paper concluded that the wide differences in IoT protocols are not undesirable, but enabling technologies that permit the IoT Ecosystem to pervade many different processes and end-user applications.

II. INTERNET PROTOCOL STACK REVIEW

The internet protocol stack consists of five layers, to include the application layer, transport layer, network layer, link layer, and physical layer [1]. The five layers are commonly accepted and form a background to help understand the IoT protocol stack.

The application layer primarily resides on the end systems (clients or servers) and interacts with the process to provide

network services [1]. Common application layer protocols include HTTP for web pages, SMTP for email, and FTP for file transfer between end systems. The application layer also includes DNS.

The transport layer also resides on the end systems and provides the application layer with end-to-end service; i.e., provides the messaging between application end points [1]. The two most common internet protocols are TCP for connection-oriented services and UDP for connectionless oriented services.

The network layer resides in the end systems as well as routers (level three devices). The network layer includes IP, along with other message routing protocols that define how end systems and routers format the datagram fields [1].

The link layer resides in the end systems, routers, and switches (level two devices). Link layer services provide the mechanism to move a message from one node to another. Typical protocols include Ethernet for messaging over Unshielded Twisted Pairs (UTP), Wireless Fidelity (Wi-Fi) for wireless links, and Data Over Cable Service Interface Specifications (DOCSIS) for coaxial physical links.

The physical layer is used for bit transfer from one node to another. The physical layer includes copper conductors, fibres, and even air. Table 1 summarizes the five layers in the internet protocol stack.

TABLE 1
Five-Layer Internet Protocol Stack

Layer	**Service**	**Typical Internet Protocols**
Application	Defines types and syntax of messages. Defines the semantics of each field and how processes send and receive messages.	HTTP, HTTPS, SMTP, POP3, IMAP, FTP, DNS
Transport	Provides the logical connection between the applications and processes running on each host.	TCP, UDP
Network	Provides the logical connection between hosts, such that different hosts can communicate as if they were directly connected.	IPv4, IPv6
Link	Provides framing, link access, reliable delivery, and error detection and correction services. The link layer is implemented within an end-systems' network adapter or Network Interface Card (NIC).	Ethernet, Wi-Fi, DOCSIS
Physical	Provides the physical transfer of bits from one network adapter/NIC to another.	N/A

III. IoT

IoT is a category of networked computational devices that "sense, compute, communicate, and control the surrounding environment" [3]. IoT devices are often small, such that they are constrained by low power and low computational resources. Further, IoT devices may not always be on and communicate over lossy networks [3]. In addition to physical constraints, IoT devices are not always connected nor always transmit data. IoT message exchanges are characterized based on events and not continuous updates such that the data flow is often irregular [3]. The heterogenous and agglomerated nature of various devices and applications is not marked by a single technology or set of common protocols, but rather many different technologies working together [4].

The IoT relies on a large range of horizonal and vertical technologies. Advances in sensing devices, efficient computing, and data analytics have enabled the diffusion of IoT devices (horizontal technologies). Advances in networking protocols in the vertical IoT architecture layers have further enabled the plug and play interaction of IoT devices within the large horizon of other enabling technologies. The overall vertical and horizontal integration of IoT sensors, devices, and applications into society is called the IoT Ecosystem [4]

The IoT Ecosystem can be described in a seven-layer architecture that consists of the market, acquisition, interconnection, integration, analysis, application, and service layers [4]. The market layer is at the bottom and includes the application domain. This layer represents the integration of each device into the smart home, grid, or end-system. The acquisition layer is the second layer and includes the actual IoT devices, sensors, and intelligent applications. Application services within this layer are analogous to the application layer in the internet protocol stack [4]. Power and computational resources at this layer require networking services to provide reliable and autonomous data exchange between devices. The third layer (interconnection layer) is analogous to the transport, network, and link layers within the internet protocol stack [4]. The services within this layer provide the means for each device to communicate to the various processes used by the end-user. The remaining higher-level layers are outside the scope of the protocol services within the internet protocol stack and are not discussed further.

There is no single and agreed consensus for an IoT layered architecture for these services, but several proposals address generic IoT protocols by a three, four, and fiver layer protocol stack. Each of these abstractions of the IoT architecture are discussed next.

A. IoT Architectures

IoT protocols reside in different layers of the overall protocol stack. In order to understand how each protocol enables the IoT Ecosystem, the protocol's services must be harmonized to other protocol services at a similar layer. This is done by first understanding the generic IoT protocol stack. The generic IoT stack can be abstracted by a three, four, and five-layer stack. Table 2 provides the equivalent layers of each IoT layer compared to the equivalent internet protocol layer. Each of these three abstractions are further discussed.

TABLE 2
IOT ECOSYSTEM PROTOCOL STACKS

IoT Ecosystem	**3 Layer**	**4 Layer**	**5 Layer**	**Equivalent Internet Protocol Stack**
Service Layer	Application	Service	Business	N/A

IoT Ecosystem	3 Layer	4 Layer	5 Layer	Equivalent Internet Protocol Stack
Application and Software			Application	
Analytics		Platform		
Integration				
Interconnection	Network	Network	Processing Transport	Transport Network Link
Applications	Perception	Device	Perception	Application
Market	N/A	N/A	N/A	N/A

1) *Three Layer Stack:* The most basic architectural abstraction is the three-layer stack consisting of a perception layer, network layer, and application layer [4]. The lowest layer is the perception layer. This layer is comprised of the physical IoT devices and uses application-level services to enable each device to access the second layer. The second layer is the network layer, which provides the services for connecting each smart device to each other, as well as the back-end servers. This layer is responsible for the transmission of messages to and from each IoT device. This middle layer is equivalent to the transport, network, and link layers in the five-layer internet protocol stack. The third and upper layer is the application layer. The top layer provides the IoT processes to the end-user and is not discussed further, as it does not have an equivalent internet protocol stack.

2) *Four Layer Stack:* The IoT layered architecture can be described in a four-layer stack consisting of a device layer, network layer, platform layer, and service layer [2]. The bottom layer is the device layer. The device layer includes the physical IoT devices (sensors and actuators) as well as its own resource management (power and computational resources). Resource management is an important service, as most IoT devices having limited power. The second layer is the network layer. Services within this layer transmit messages between IoT devices and gateways. Equally important, this layer allows the physical mobility of the sensors and actuators within the lower layer. The platform layer provides the interface to the end-user. The back-end computation for the analytics necessary for IoT reside at the service layer [2]. Both the platform and service layers are outside the scope of the internet protocol stack and are not discussed further.

3) *Five Layer Stack:* A five-layer IoT stack expands upon the three-layer stack by diving the application layer into two layers, the business and application layer. The network layer is also segmented into two layers, the processing layer and transport layer. The resulting IoT architecture includes a bottom perception layer, transport layer, processing layer, application layer, and business layer [4]. The bottom perception layer in the five-layer stack is the same as the three-layer stack and represents the physical layer or actual devices. The transport layer resides just above the lowest layer and provides the transfer of data from the perception layer to the processing layer. The processing layer, application layer, and business layer take the data provided from IoT and perform various user level services such as storage, analytics, etc. The top three layers do not provide internet protocol equivalency and are not discussed further.

B. Common IoT Protocol Stacks

Although there are hundreds of IoT protocols, four common protocols were reviewed to compare the complete protocol stack at each layer. These include Open Stack For IoT (OS4I), Bluetooth Low Energy (BLE), ZigBee, and Long Range Low Power Wide Area Network (LoRaWAN). Each of these open IoT protocols is introduced and explained within its own protocol stack.

1) *OS4I:* OS4I is an IoT protocol stack that only uses open technologies including "IEEE 802.15.4, 6LoWPAN, UDP/TCP and CoAP/MQTT" [5]. OS4I uses the standard internet protocols Constrained Application Protocol (CoAP) and MQ Telemetry Transport (MQTT) as the application layer protocols. CoAP is an open protocol built specifically for low power devices and is very similar to HTTP, but offers the additional capability to observe, block, and recover. MQTT was developed by IBM, but is now an open standard [6]. MQTT minimizes the overhead in message sizes, which is advantageous for congested networks. The open protocols UDP and TCP are used at the transport layer. CoAP uses UDP and MQTT uses TCP. IPv6 is a standard network protocol used by the network layer. The Internet Engineering Task Force (IETF) standard for Low Power Personal Area Network (6LoWPAN) carries the IPv6 packets over IEEE 802.15.4. IEEE 802.15.4 is a wireless standard (physical layer) for 2.4 GHz, 915 MHz, and 868 MHz frequencies. The 2.4 GHz and 915 MHz are segmented into 16 and 10 channels respectively. 6LoWPAN is required to be used, because IPv6 packets are larger than IEEE 802.15.4 and must be fragmented or compressed [5]. Table 3 describes the OS4I protocol stack.

TABLE 3
OS4I PROTOCOL STACK

Generic Internet Protocol Stack	OS4I Protocols
Application Layer	CoAP and MQTT
Transport Layer	UDP and TCP
Network Layer	IPv6
Link Layer	6LoWPAN
Physical Layer	IEEE 802.15.4

2) *BLE:* Bluetooth Low energy (BLE) is a variant of the classic Bluetooth protocol developed for short range communications [5]. The Bluetooth version 4.0 specification describes BLE with a four-layer protocol stack. Within this stack, the link layer is the most unique amongst all of the IoT protocol stacks. In BLE, the link layer is responsible for scanning, advertising, initiating, and maintaining messaging with other devices and gateways. The BLE link layer supports six messaging roles to include an advertiser, scanner, slave, master, broadcaster, and observer. The advertiser and scanner roles are paired for sending broadcasts and sensing each of the 40 possible channels [7]. The master and slave roles describe how an advertiser device accepts a connection and how a master device is a scanner that initiates the connection. The broadcaster and observer role describe how

devices can send messages without any receipt (broadcaster), but also detect other broadcasters without sending any messages (observer). Beginning with the transport layer, BLE uses Generic Access Protocol (GAP) and Generic Attribute Protocol (GATT). GATT is a transport layer protocol that enables the application layer services to serve as both a client and server in a master-slave architecture. A device can act as a master when it's a central device, but act like a slave when it's on the periphery [5]. GAP is used concurrently with GATT to enable a device to join a network as an observer only, broadcaster, central, or peripheral. A broadcaster role enables the IoT device to implement unidirectional messaging. GATT is built on top of the network layer Attribute Protocol (ATT) and Security Manager (SM) protocol. ATT enables read and write functions between the layers. ATT manages the pairing session between the IoT devices and the gateway. The network layer protocol ATT is used specifically for the Logical Link Control and Adaption Protocol (L2CAP) link layer. The BLE L2CAP protocol is very similar to the classic Bluetooth L2CAP, but provides the multiplexing of the ATT and SM layers. The BLE link layer is a 2.4 GHz wireless frequency split into 40 channels. Three channels are used for pairing between devices, leaving the remaining 37 channels open for full duplex communication [7]. The BLE protocol stack is summarized in Table 4.

TABLE 4
BLE PROTOCOL STACK

Generic Internet Protocol Stack	BLE Protocols
Application Layer	N/A
Transport Layer	GAP and GATT
Network Layer	ATT and SM
Link Layer	L2CAP
Physical Layer	GSFK

3) *ZigBee:* ZigBee was designed to be a low power Personal Area Network (PAN) for devices with limited data [7]. Similar to OS4I, ZigBee is built upon the IEEE 802.15.4 physical layer. The ZigBee protocol uses IEEE 802.15.4 for both the link layer and physical layer, and therefore only specifies the protocol's top three layers. The application layer uses the Application Framework (APF) to define up to 254 Application Objects (APO) with a unique endpoint assignment number. Each APO is used to control a specific piece of hardware; e.g.: switch [5]. APO uses the Zigbee Device Object (ZBO) services to communicate between the APO endpoint assignment and its specific device. ZPO also provides security services similar to the SM protocol used by BLE. ZDO is considered both an application layer protocol and a transport layer protocol because it links each SDO through the network layer to the device. The Application Sub Layer (APS) serves as a link between the APF and the network layer, as well as provides the messaging link between APF and APS services. APS uses several different ZigBee alliance protocols that form the structure of message formats sent to the network layer. APS supports any device by any manufacturer, as long as the messaging conforms to published ZigBee APS message protocols [5]. The ZigBee Network (NWK) layer provides several services. Some of these include multi-hop routing, joining new networks, route discovery, and security. Table 5 presents the ZigBee protocol stack.

TABLE 5
ZIGBEE PROTOCOL STACK

Generic Internet Protocol Stack	ZigBee Protocols	
Application Layer	ZDO	APF
Transport Layer		APS
Network Layer	NWK	
Link Layer	IEEE 802.15.4	
Physical Layer		

4) *LoRaWAN:* The Long Range Low Power Wide Area Network (LoRaWAN) is specifically designed and maintained by the LoRa Alliance to enable low power devices to communicate over long distances [5]. Similar to previous IoT protocols, LoRaWAN uses HTTP and MQTT protocols for application and transport layer services. LoRaWAN uses LoRa for physical layer services and is a proprietary protocol making LoRaWAN a closed protocol. LoRa uses a 915 MHz frequency in North America (sub-gigahertz radio frequency band), which enables end devices and gateways to be separated between 5 km to 50 km (depending on the conditions). LoRaWAN resides on the network and link layers to define the network's system architecture as well as the specific messaging protocol. LoRaWAN enables larger distances based on a classification scheme; each class is full duplex [5]. Beginning with lowest Class A, each LoRaWAN device must be supported with the least amount of power as possible. However, limited power consumption also introduces latency, as power reduction is enabled by using only two downlink windows. In addition to the Class A device, LoRaWAN also supports a higher Class B device with larger power requirements. Class B devices are given additional downlink windows to reduce latency and for time-synchronization between the device and gateway. The time synchronization service allows the gateway to "know" when the LoRaWAN device is receiving messages. The highest class, Class C, reduces latency the most, but only supports higher powered devices. Latency is reduced by allocating a continuous downlink to each device [5]. The LoRaWAN protocol stack is illustrated in Table 6.

TABLE 6
LORAWAN PROTOCOL STACK

Generic Internet Protocol Stack	LoRaWAN Protocols
Application Layer	HTTP
Transport Layer	MQTT
Network Layer	LoRaWAN
Link Layer	
Physical Layer	LoRa

C. IoT Service Characteristics

IoT protocols are very heterogeneous, but can be generally compared to each other by their service characteristics. Service characteristics include range, openness, interoperability, network topology, and security [5].

1) *Range:* The IoT Ecosystem is based on wireless communication between each object. As such, the physical distance between an object and gateway is a critical characteristic that is tailored for specific applications. In order to review and compare each IoT protocol, range is abstracted in terms of the "area of action" the device can operate in [5]. There are three general areas of action: Personal Area Networks (PAN), Local Area Networks (LAN), and Wide Area Networks (WAN). Protocols used within a PAN are worn by a user and connect to a central gateway on the user themselves, supporting a maximum communication distance of a few meters. A typical example is wearable sensors that connect to a smart phone using the common Bluetooth protocol. Protocols for IoT devices used in a LAN are limited by the geographic location of the LAN and have a range of 100 m [7]. A typical example would be a house, office building, or property. IoT LAN protocols typically operate at 2.4 GHz, which is the standard Industrial, Scientific, and Medical (ISM) frequency band [5]. WAN protocols used by IoT devices enable communication up to several tens of kilometres. Protocols enabling WAN communication commonly use cellular technology that operate in the sub-GHz range. Each IoT protocol enables different ranges:

- OS4I: IEEE 802.15.4 uses the 2.4 GHz frequency band to support a range up to 100 m. IEEE 802.15.4 is a LAN based protocol that can also be used in a PAN.
- BLE: GSFK uses the 2.4 GHz frequency band to support a range up to 100 m and is both a PAN and LAN based protocol.
- ZigBee: Similar to OS4I and BLE, ZigBee is a PAN/LAN based protocol and supports a range up to 100 m.
- LoRaWAN: LoRaWAN is a cellular physical layer, permitting communication up to five km. However, the range can be extended depending on the power availability, environment, and congestion; thus, LoRaWAN can support long range networks [8].

2) *Openness:* The service characteristics of openness refers to the transparency of the IoT protocol. Standard protocols such as TCP and IP are open. Open IoT protocol stacks have published technical specifications, such that the protocols in the stack as well as the structure of the stack are transparent and available [5]. Closed protocol stacks have no published standards or specifications and are proprietary to the originating organization. Some IoT protocols make use of a hybrid approach and use open protocol stacks, but keep some specific details proprietary. When some, but not all information is known, the protocol stack is referred to as half-open [5]. Each IoT protocol can be ranked in terms of its openness:

- OS4I: OSI4I is fully open as all protocols and the protocol stack are published.
- BLE: BLE is half-open as it was developed and is maintained by the Bluetooth Special Interest Group (SIG).
- ZigBee: ZigBee is half-open as it was developed and is maintained by the ZigBee Alliance.
- LoRaWAN: The LoRaWAN physical layer uses LoRa, which is proprietary to the LoRa Alliance. As such, this protocol is considered closed.

3) *Interoperability:* IoT protocols may have unique protocols and protocol stacks, but still rely on IP. Interoperability reflects the ability of a IoT protocol stack to be used directly with IP or require a proxy for communication [5]. Each IoT protocol can be ranked in terms of its interoperability:

- OS4I: OS4I uses IPv6, so its fully interoperable and does not require a proxy for communication.
- BLE: BLE is not compatible with IP and requires a proxy.
- ZigBee: ZigBee is not compatible with IP and requires a proxy.
- LoRaWAN: LoRaWAN is not compatible with IP and requires a proxy.

4) *Network Topology:* IoT protocols can support many different network topologies. Common topologies include a tree, star, peer-to-peer (P2P), and cellular. The network topology structure depends on how the gateway controls each end device, and can be dynamic when devices, routers, or gateways come in or leave the network [9]. The traditional tree topology includes a single gateway for the entire network and forms the root. Each end device can attach to the network through a router and daisy chain, such that several end-devices communicate with each router and routers can communicate with other routers. A star is similar to the tree, but each end-device communicates directly to the gateway. Both the tree topology and star topology are considered traditional topologies, as they control the master-slave relationship between the gateway and end device. The gateway controls all communication. A cellular topology is simar to a star topology, but includes multiple gateways, such that each end-device can communicate with one or more gateways. IoT devices arranged in a cellular topology must deal with redundancy, as an end-device might send the same message to multiple gateways [5]. A Point-to-Point (P2P) topology is similar to the classic tree and star topologies, but includes multiple routers. Each router can communicate with each other router to provide multiple different paths for the end-device to reach the gateway. Each IoT protocol can be described in terms of the network topologies it supports:

- OS4I: OS4I is a wireless IoT protocol, but is limited by the IEEE 802.15.4 physical layer. IEEE 802.15.4 is not a cellular link, such that OS4I only supports star, tree, and mesh topologies.
- BLE: BLE is a wireless IoT protocol, but is limited by the L2CAP link layer's routing capabilities. L2CAP only supports star and mesh topologies. The star topology is supported by the master-slave roles. Mesh topologies are supported with up to 127 router hops [5].
- ZigBee: Similar to OS4I, ZigBee is not a cellular link and provides support for the star, tree, and mesh

topologies [8]. The star and tree topologies are supported by the parent-child relationship between each beacon and device. The mesh topology is supported using a routing discovery algorithm, but will revert to a tree topology if required resources are not available. The device specific configured topology affects the specific routing algorithm used [5].

- LoRaWAN: LoRaWAN fully supports a cellular architecture.

5) Security: Internet security typically involves the three dimensions of integrity, availability, and confidentiality. Unlike the traditional IT domain, the IoT Ecosystem places availability first, and then prioritizes integrity and confidentiality [10]. When translating security principles into protocol requirements, protocol services must provide resiliency to attacks and provide the ability to recover from a crash. Additionally, specific techniques such as device and data authentication must be supported. Access rights that provide client privacy are also a critical security feature the protocol must support [11]. Despite these three dimensions and specific principles for a secure protocol, the IoT Ecosystem has a much higher attack surface and therefore protocol transparency and continual improvements enable a more robust protocol (i.e., open protocols are often more secure). Each IoT protocol can be analysed by the security services it provides:

- OS4I: OS4I is based on a five-layer protocol stack, each of which provides security services including both encryption and authentication. The application layer services CoAP and MQTT use the transport layers for security. CoAP uses UDP, which includes Datagram Transport Layer Security (DTLS). MQTT uses the CTP protocol's Secure Sockets Layer (SSL) and Trusted Ticket Server (TTS). TTS provides authentication services [12]. As the network layer uses IPv6, IPsec is provided as a network security mechanism. IEEE 802.15.4 provides encryption using Advanced Encryption Standard-Counter with Cipher Block Chaining-Message Authentication Code (AES-CCM).
- BLE: Similar to OS4I, BLE provides multiple security services at different layers. The SM at the network layer provides security during pairing through authentication services using SIM Application Toolkit (STK). L2CAP (link layer) provides encryption using AES-CCM.
- ZigBee: The ZigBee protocols APS (transport layer) and NWK (network layer) provide security services. The two layers provide security through the use of a network and link key, both are 128 bit [7]. Additionally, ZigBee uses AES-CCM for encryption.
- LoRaWAN: LoRaWAN supports both encryption and authentication. Authentication is supported using 128 bit application and network keys. LoRaWAN uses Advanced Encryption Standard (AES) to provide encryption services.

D. IoT Case Study: IoT Applications in Smart Farming Communication

Smart farming uses IoT solutions to boost productivity and minimize waste. Sensors and final control elements (e.g.: valve actuator) collect, send, and receive data across wireless networks to information systems and data analytic services. Sensors measure variables such as temperature, humidity, light, pests, disease, and pressure to automatically take action or notify farm operators of any "necessary actions to be carried out at the right time, quantity, and place" [8]. Smart farming Wireless Sensor Networks (WSN) are based on sensor nodes and Single Board Computers (SBC) (e.g.: Arduino) to provide a geographically distributed surveillance capability linked to cloud-based data processing services. WSN(s) are not uniform nor standardized, but deployed for specific applications. Each of the four common IoT protocols OS4I, BLE, ZigBee, and LoRaWAN are used based on the unique service each provide.

Beginning with the process variable under surveillance, sensor selection is based on the specific requirements. Requirements in smart farming generally fall into crop monitoring, substrate monitoring, and environmental monitoring. Crop monitoring is supported by a diverse set of manufacturers and devices to measure crop canopy, growth, insects, and disease. Substrate monitoring measures the soil temperature, moisture, pH, and chemical properties such as nitrogen. Environmental monitoring measures the air temperature, humidity, solar radiation, rain, luminosity, pressure, wind speed/direction, and CO2 concentrations [8]. These agricultural process variables are found in many different farming scenarios, which can be grouped into open arable land, contained greenhouses, and orchards. With such a diverse set of process variables, applications, and farming scenarios, no single device, sensor, or communication technology can be used.

In smart farming, the star architecture is the dominant topology. In all three farming scenarios (arable land, contained greenhouses, and orchards), WSN(s) use BLE or ZigBee to connect each of the devices to a central node, which then uses LoRaWAN to communicate back to a single base station with internet access [8]. In large open arable lands, several groups of sensors are arranged in a star network and then daisy chained together using ZigBee to form a tree architecture. The lower power requirements of BLE and ZigBee enable a large number of sensors to be deployed with minimal cost. Although BLE supports a higher data rate (up to 24 Mbps) than ZigBee (up to 250 kbps), only ZigBee supports the daisy chaining of multiple stars, such that both protocols are equally prevalent. Smaller and denser networks in greenhouses make use of OS4I to provide direct access to the internet without the use of an IP proxy (OS4I uses IPv6). These more compact applications of WSN(s) are cost justified due to the higher crop yield per network area.

Using a single protocol such as OS4I would limit the range and supported topology of the WSN. Further, using only one protocol such as BLE would require higher powered devices with higher cost. The ability to use multiple IoT protocols to

support a diverse population of sensors has enabled smart farming to provide increased productivity while minimizing cost and waste in each of the three farming scenarios [8].

IV. Conclusions

This paper presented the five-layer internet protocol stack with its common protocols for each service layer. The five-layer internet protocol stack forms the basis to understand alternative protocol stacks used in the IoT Ecosystem. IoT has emerged as a dominant domain in digital communications and provides the ability for sensors and other end-devices to sense, analyse, and control the physical environment. The IoT is based on devices connected to the global network. Various IoT protocols were reviewed and compared to the basic internet protocol stack. Although there is no universally agreed IoT protocol stack, there is a three, four, and five-layer abstraction that generalizes all of the equivalent service layers to the five-layer internet protocol stack.

The IoT protocols OS4I, BLE, ZigBee, and LoRaWAN are widely distributed IoT protocols. Each of the four IoT stacks is slightly different and the various service layers provide unique capabilities. Capabilities differ across the six services of range, openness, interoperability, network topology, and security.

From the review of the four IoT protocols in context of a case study, it is clear that the heterogenous nature of the IoT requires many different protocols to support each application within the IoT Ecosystem. There is no single better or worse IoT protocol, as each has a specific niche for its intended application. PAN and LAN based IoT protocols have different characteristics than WAN based protocols, which is mainly driven by the messaging over the physical link. Although IoT devices are constrained by power and computation resources that drive many of the upper-level protocols within each stack, many of the different IoT protocol stacks rely on shared networking protocols with the equivalent five-layer internet stack (such as UDP and TCP). As such, although the IoT protocol stacks may all be different than the internet protocol stack, they all provide equivalent services for each process/device within its unique application.

This paper presented a knowledge gap on whether the heterogenous nature of IoT protocols is an enabling or potentially detrimental feature for continued IoT growth. The main conclusion is that the diversity of IoT protocols is as enabling as the IoT protocols themselves. It is the diversity of IoT protocols that has enabled IoT to fill in different needs, different applications, and different markets. The diversity of the IoT Ecosystem enables it to enhance all aspects of modern residential, commercial, and industrial domains.

References

[1] J. Kurose and K. Ross, Computer Networking, A Top-Down Approach, Harlow, UK: Pearson Education, 2021.

[2] S. K. Lee, M. Bae and H. Kim, “Future of IoT Networks: A Survey,” *Applied Sciences,* vol. 7, 2017.

[3] J. Mocnej, A. Pekar, W. Seah, E. Kajati and I. Zolotova, “Internet of Things Unified Protocol Stack,” *Acta Electrotechnica et Informatica,* vol. 19, no. 2, pp. 34-32, 2019.

[4] P. Sethi and S. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *Journal of Electrical and Computer Engineering,* 2017.

[5] J. Tournier, F. Lesueur, F. Mouel, L. Guyon and H. Ben-Hassine, “A survey of IoT protocols and their security issues through the lens of a generic IoT stack,” *Elsevier,* p. 100264, 2020.

[6] L. Tightiz and H. Yang, “A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication,” *Energies,* vol. 13, 2020.

[7] G. Kambourakis, C. Kolias, D. Geneiatakis, G. Karapoulos, G. M. Makrakis and I. Kounelis, “A State-of-the-Art Review on the Security of Mainstream IoT Wireless PAN Protocol Stacks,” *Symmetry,* vol. 12, no. 579, 2020.

[8] E. Navarro, N. Costa and A. Pereira, “A Systemic Review of IoT Solutions for Smart Farming,” *Sensors,* vol. 20, 2020.

[9] E. Haque, Asikuzzaman, I. Khan, I.-H. Ra, S. Hossain and S. Shah, “Comparative Study of IoT-Based Topology Maintenance Protocol in a Wireless Sensor Network for Structural Health Monitoring,” *Remote Sensing,* no. 12, 2020.

[10] Chief Information Officer, “DoD Policy Recommendations for The Internet of Things (IoT),” U.S. Department of Defense, Washington D.C., 2016.

[11] M. A. Razzaq, M. A. Qureshi, S. Ullah and . S. H. Gill, “Security Issues in the Internet of Things (IoT): A Comprehensive Study,” *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 6, 2017.

[12] A. Leicher, A. Schmidt, Y. Shah and I. Cha, “Trusted Computing Enhanced OpenID,” in *IEEE Explore*, 2010.