electronics

MDPI

Review

The Advents of Ubiquitous Computing in the Development of Smart Cities—A Review on the Internet of Things (IoT)

Jawad Ali [1], Mohammad Haseeb Zafar [2], Chaminda Hewage [2], Syed Raheel Hassan [3] and Rameez Asif [3,*]

[1] Department of Electrical Engineering, University of Engineering and Technology, Mardan 23200, Pakistan
[2] Cybersecurity and Information Networks Centre (CINC), Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff CF5 2YB, UK
[3] School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK
* Correspondence: rameez.asif@uea.ac.uk

Abstract: By leveraging ubiquitous computing and the Internet of Things (IoT), smart cities gain potential to provide a wider range of services. Different homogeneous and heterogeneous networking schemes and applications have been proposed in the literature to date. In these networking schemes, human and computer are connected for social, economic, physiological, and technological growth. However, there is a dearth of recent literature that incorporates recently proposed and operating techniques and technologies capable of enhancing the productivity of human and machine in IoT technologies. The role of this research is to investigate the protocols, followed by the advance frameworks for IoT, the characteristics and services that are being governed using IoT for establishing information-rich smart cities. To this end, likewise, physical layer, media access control, networking and applications protocols, and encapsulation standards of IoT for smart cities applications are critically reviewed. Certain open issues are discussed based on the literature collected that would improve the autonomous behavior, process control, device handling, and the QoS in smart environments.

Keywords: IoT; smart cities; ubiquitous computing; heterogeneous networking

Citation: Ali, J.; Zafar, M.H.; Hewage, C.; Hassan, S.R.; Asif, R. The Advents of Ubiquitous Computing in the Development of Smart Cities—A Review on the Internet of Things (IoT). *Electronics* **2023**, *12*, 1032. https://doi.org/10.3390/electronics12041032

Academic Editors: Sabrine Kheriji, Olfa Kanoun and Faouzi Derbel

Received: 31 October 2022
Revised: 14 February 2023
Accepted: 14 February 2023
Published: 19 February 2023

1. Introduction

The remarkable growth and outcomes of IoT technology are fueled by the continuous support by academia and industry. It is for this fact that the evolution of microtechnology [1], mobile networking [2], machine learning [3,4], and advancement in IoT [5] is of an expanding nature. The assessment done by Gartner in [6] provides the road map for the maturity of more than 2000 technologies. The technologies are grouped into 119 distinct areas including digital workplace, smart machines, 3D printing, enterprise mobile security [7–9], and connected homes [10–12]. IoT and wearable user interfaces, which are both based on machine learning, are likely to not experience inflation. Similarly, Gartner predicts that by 2025, 50% of industries would be using industry cloud platforms to accelerate their businesses.

The technologies based on IoT do not reside inside PCs and computers only. Ranging from a small light-dependent resistor (LDR), home appliances, traffic signals, air traffic control, weather systems, and sensory networks for pressure, humidity, and temperature, to the control system of a space shuttle's propulsion, all are considered in this autonomous machine-governed system. This concludes that the system of IoT comprises four basic sectors, which can be actuators, sensors, network, and computers. These sectors must take part in facilitating the user in development, as well as in sustaining human lives.

IoT transforms passive, data-driven devices, i.e., things that make exclusive decisions, into a ubiquitously communicating and in-phase system for making concurrent single decisions in a feedback mechanism. Thus, the more it sounds easy to use, the more it complicates the situation for its designers to loop all the quantities and things with one another. The heterogeneous nature of IoT does not support a standard model for its mode

of connectivity. Proper handshaking and making it continuous for uninterrupted data analytics and decision-making also requires more and more system renovation in IoT each time a new thing is considered.

The paper highlights the support of IP protocol stack by IoT, wirelessly identifiable things [13], edge computing [14], persistence computing and ambient intelligence, recognition-interaction of portable devices [15] and their continuous handshaking, small-talk on faultless connection between people and things, a smart city's components, self-healing properties in IoT, and the framework for smart cities with user perspective.

The organization of the paper is as follows. Section 2 encompasses the introductory concept of connected cities with respect to its market and technologies, data acquisition, device interfacing, connectivity and integration, data analytics, applications and software employed, and core services providers. Section 3 explains the importance of the physical and media access control layer in IoT. Section 4 includes the most widely used MAC layer IoT protocols in detail. Section 5 lists the networking protocols that are extensively used in IoT for the development of smart cities, along with their roles and limitations. Section 6 thoroughly explains IPv6-based encapsulation schemes for network layer. The IoT application layer protocols are thoroughly explained in Section 7 with the list of features that each application protocol provides. The discussion and open issues in the field of IoT with respect to the development of smart cities and concluding remarks are listed in Sections 8 and 9, respectively.

2. Connected Cities and Its Development

Several survey papers thoroughly inscribe the development of IoT and its application in smart cities. The work on connectivity protocol and standard procedures is given in [16]. This work explains the use of a developed model by the Internet Engineering Task Force (IETF). The IETF has enabled the IEEE 802.15.4 stack to connect with standard IPv4- and IPv6-based networks because IP-based networking is widely used. Low-power and lossy network routing (RLP) and constrained application protocol (CoAP) are discussed in this work.

IoT must provide high-level security for secure data and information exchange. Various security protocols are analyzed in [17]. These protocols authenticate the data and processes and defend the integrity, as well as confidentiality, of the IoT data. The work relates IoT security to only the MAC layer of the IEEE 802.15.4 standards because physical layers are using the ASM band of 2.4 GHz with 16 channels and are provided with a direct sequence spread spectrum (DSSS) for overcoming channel congestion and improving the signal-to-noise Ratio (SNR). On the other hand, the MAC layer starts from a 32-bit message integrity code (MIC) data authenticity up to 128-bit data authenticity and encryption. The physical frame for IEEE 802.15.4 is 128 bytes and can be four types of frames defined by MAC—data frame, acknowledgement frame, beacon frame, or MAC command frame.

IoT Market and Governing Technologies Overview

The IoT market is widespread and is continuously expanding because of its core feature of autonomous control. IoT services can be used in the energy sector, health care, education, transportation, military, aerospace, agriculture, environmental studies, etc. [18]. These separate yet common-to-man systems must be unified for maximum output, as well as to be in the reach of beneficiaries. Therefore, regardless of the application of IoT in a unique environment (module), the basic architecture of the service follows the same standard protocol stack.

The contribution of each stack level is given in Table 1.

Table 1. Components based on the consumer market.

Level	Component(s)	Area
Market [11]	Smart Home, Connected Homes, Smart Health, Smart Cities, Smart Grid and Energy distribution, Surveillance . . .	Personal, Local, Global
Data Acquisition [12]	Sensors and actuators: pressure, temperature and humidity sensing, GPS, measurements, smartphones . . .	Personal, Local
Interfacing [13]	Wi-Fi, Bluetooth, ZigBee, Near Field Communication (NFC), DECT/DECT-ULE, . . .	Personal, Local
Connectivity and Integration [13]	Data sensing, Economics and Stats, GIS . . .	Local, Global
Data Analytics [19]	Machine Learning, Statistical Analysis, Data mining, Big Data handlers . . .	Local, Global, Security, computing, Management
Applications and Software [20]	Clouds, Data centers, Local observatories, Software Defined Networking (SDN), Modular cities connectivity, Service Oriented Architecture (SOA) . . .	Global, Selective/User defined
Core Services [21]	Energy, Health, Entertainment, Industry, Production, Transportation, Communication . . .	Local, Global, sometimes personal

3. Physical (PHY) Layer and Media Access Control (MAC) in IoT

The physical link layer is used for wired or wireless connectivity of different devices. The connection can be token based or it can be managed by datalink layer. In both cases, permission is granted for packet transmission or reception by the media access control (MAC) schemes [22]. There are four main characteristics of the PHY layer that are considered for setting up a connection between a transmitter and a receiver in IoT that includes modulation scheme, transmission mode, channel encoding data rates/throughput, and MAC protocol.

3.1. Modulation Scheme

Modulation scheme and index depend upon the severity of the channel noise, as well as the type of application run on/for a node in IoT. Generally, IoT aims for low cost and complexity. For this reason, computationally expensive and advanced modulation schemes such as trellis-coded modulation, etc. are excluded at first. In the case of wider frequency bands for wireless connectivity, DSSS is employed, but since IoT nets are deployed with a massive number of nodes, DSSS is precluded because of regulatory constraint and narrowband interference. This will save channel bandwidth for other communications. DSSS increases channel bandwidth and requires more power for transmission [23]. For task-specific communication, direct burst modes are achieved using single channel communication with no spectrum hopping. It is a high-power low-bandwidth communication scheme.

3.2. Transmission and Network Modes in IoT

Transmission modes are selected with respect to the application constraints and requirements. In areas where deployment of devices is massive [24], it is recommended to use a low data rate and noise tolerant connectivity schemes. Similarly, a dense network uses low-power low-range communication. Sensors in combination with actuators use bi-directional data communication modes. Separate devices might use either transmitter or receiver, based on the application requirements. Similarly, mesh topology is generally employed for geographically scattered nodes, whereas for general home automation, industrial automation, etc., the central-coordinator-based star topology is preferred [25]. Z-Wave is used for geographically scattered IoT [26]. Its data rates are lower as compared to other IoT-enabling technologies, but its networking complexity is reduced because of its 232 fixed number of hops for data transfer and control in an IoT.

3.3. Media Access and PHY Layer

IoT is implemented using Wi-Fi, Bluetooth, ZigBee, and Z-Wave standards; thus, medium access varies, as well as the number of channels and frequency spectrum usage. Medium is accessed either using the non-beacon mode or the non-slotted CSMA/CA method, or the MAC is performed using a beacon-based slotted CSMA/CA mode. BPSK is used for lower data rates, whereas quadrature amplitude modulation (QAM) is used for elevated data rates in the order of 2n, where n is the number of bits per symbol [27].

3.4. Data Rates in IoT

The physical layer of IoT is responsible for variable data rate support. As mentioned earlier, the technological diversity of IoT makes it versatile for the efficient utilization of resources including bandwidth and battery constraints. For this reason, a single technology like ZigBee operates at three different data rates, i.e., @250 kpbs on 2.4 GHz, 40 kbps on 915 MHz, and 20 kbps on the 868 MHz channel. Transmission rates for different technologies that are employed in IoT are given in Table 2.

Table 2. IoT technologies comparison with respect to their data rates.

Technology	Sub-Category	Operating Frequency	Data Rate
Wi-Fi	802.11a	5 GHz	54 Mbps
	802.11b	2.4 GHz	11 Mbps
	802.11g	2.4 GHz	54 Mbps
	802.11n	2.4 & 5 GHz	600 Mbps
	802.11ac	5 GHz	1.3 Gbps
	802.11ah	2.4 & 5 GHz	347 Mbps
ZigBee	802.15.4	868 MHz	20 kbps
		915 MHz	40 kbps
		2.4 GHz	250 kbps
Z-Wave	G.9959	868.42 MHz	9.6 kbps
		908.42 MHz	40 kbps
Bluetooth 1	802.15.1	2.4 GHz	721 kbps
Bluetooth 2–4			2.1–3 Mbps, 25 Mbps (+HS Wi-Fi)
Bluetooth 5.0 (IoT LE)			2 Mbps
LoRaWAN (minimum and maximum spread factor (SF)-based data rates division)	LoRa:SF12 LoRa:SF7	125 kHz	250 bps 5.47 kbps
	LoRa:SF12 LoRa:SF7	500 kHz	980 bps 21.9 kbps

Table 3 depicts the general categorization of each MAC-layer technology with respect to availability, data transfer rate, bandwidth, mobility, range, and security.

Table 3. IoT technology categorization with respect to its physical characteristics.

Characteristic → Level ↓	Availability	Data Transfer Rate	Node Mobility	Range and Coverage	Data Security	Power Consumption	Size of Devices Included
Low	Bluetooth, RFID, NFC, Z-Wave, Zigbee, Sigfox, LoRa	Bluetooth, RFID, NFC, Z-Wave, Zigbee, Sigfox, LoRa	Wi-Fi, Bluetooth, RFID, NFC, Zigbee	Wi-Fi, Bluetooth, RFID, NFC, Zigbee	Wi-Fi, Cellular Network, Bluetooth, LoRa	Bluetooth, RFID, NFC, Z-Wave, Zigbee, Sigfox, LoRa	Bluetooth, RFID, NFC, Z-Wave, Zigbee, Sigfox, LoRa
High	Wi-Fi, Cellular Network,	Wi-Fi, Cellular Network	Cellular Network, Z-Wave, Sigfox, LoRa	Cellular Network, Sigfox, LoRa	RFID, NFC, Z-Wave, Zigbee, Sigfox	Wi-Fi, Cellular Network	Wi-Fi, Cellular Network

The smart city applications vary the requirements for communication, and most of the time, we do not use frequent data transfers. Table 4 gives a detailed overview of the communication specifications in enabling smart cities. Note that these specifications are for the Padova Smart City project [28].

Table 4. Smart city services for Padova city and its network type, constraints, and feasibility overview.

Service	Network Type	Traffic Rate	Tolerable Delay	Energy Source (Mostly Used)	Feasibility Complications
Structural Health	Wi-Fi + Ethernet	Every 10 min	30 min normal, 10 s alarm	Battery	Integration
Waste management	Wi-Fi + 3G/4G	Hourly	30 min	Battery + Harvesters	Integration
Noise Monitoring	BLE/ZigBee + Ethernet	Every 30 min	5 min	PV Cells	Implementation
Traffic Congestion	Bluetooth + Wi-Fi + Ethernet	Every 10 min	5 min normal, 10 s alarm	Battery + Harvesters	Realization and integration
Energy (production and consumption)	PLC + Ethernet	Every 10 min	5 min normal, alarm varies	Mains	Application
Parking	BLE/ZigBee + Wi-Fi + Ethernet	Application dependent	1 min	Energy Harvesters	Integration
Lighting	Wi-Fi + Ethernet	Application dependent	1 min	Mains	Modification of current system
Building Automation	BLE/ZigBee + Wi-Fi + Ethernet	Local: 30 min Remote: 10 min	Local: 1–10 s Remote: 5 min	Mains + Battery	Modification of current system

4. M2M-Based Division of MAC Layer Protocols

The designing of MAC layers is a rich communication technology field. Existing MAC protocols that are devised into contention-based, contention-less, and hybrid protocols have their advantages and complications that are inherited by IoT. The taxonomy of MAC layer protocols is given by Figure 1.

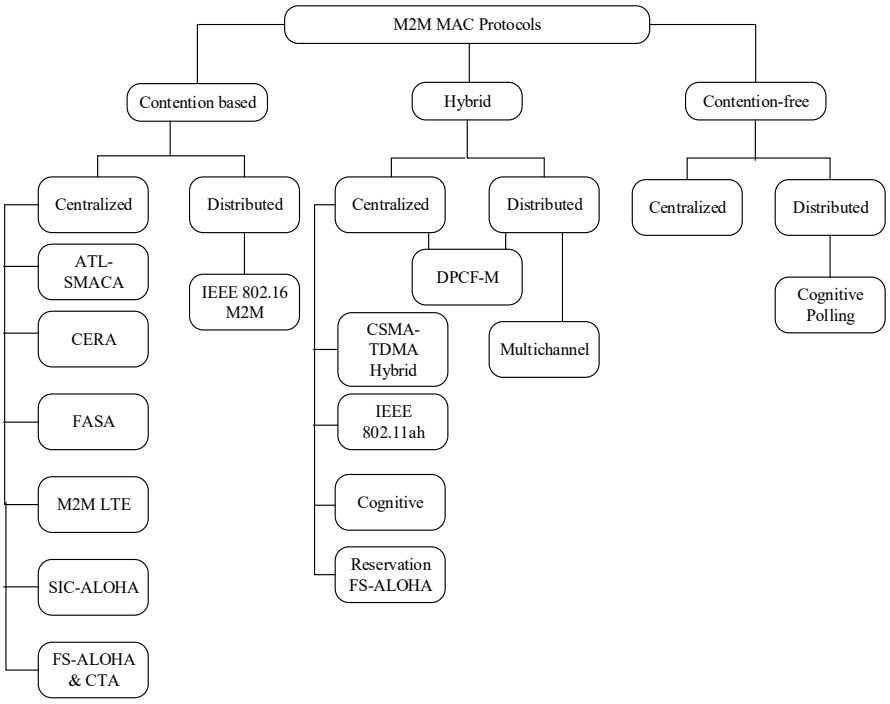

Figure 1. MAC layer M2M protocols.

To address the unique requirements and limitations of IoT, MAC layers for M2M protocols are studied. The most widespread MAC layer IoT protocols are explained in the upcoming subsections.

4.1. Distributed Point Coordination Function-M (DPCF-M)

A distributed point coordination function-M is proposed in [29] that addresses the energy constraints of M2M communication. This protocol is specifically designed for IoT functionality where two scenarios are considered. Local M2M communicating devices and gateway-capable nodes are exclusively enabled for short- and long-distance communication. This protocol uses the non-beacon CSMA/CA mode of IEEE 802.15.4. The gateway nodes also give facility of communication between remote servers using a cellular interface. This protocol outperforms the traditional 802.11 CSMA/CA in energy efficiency and throughput, but the introduction of additional hardware adds extra money to the cost of system upgrades. The protocol exhibits local communication collisions, as experienced by CSMA/CA, on the other hand.

4.2. CSMA-TDMA Hybrid MAC

This Scalable hybrid MAC protocol for M2M communication is proposed in [30]. This protocol divides time into frames and each frame is further divided into four parts, which are notification period, contention period, announcement period, and transmission period. These frame fragments reduce collisions between packets during their exchange but pose problems of extra delays and extra battery drainage.

4.3. Adaptive Multichannel Protocol for Large-Scale M2M

This protocol divides the available bandwidth into control channels and communication channels [31]. The time slots are also divided into estimation, negotiation, and data transmission phases. The estimation phase consists of a number of time slots in which the nodes send busy messages on the control channel if they either have data to send or if they hear a busy tone from other participating nodes. Based on the number of sent and received busy tones, statistical estimation is done for finding the number of active nodes on the network. The negotiation phase then assigns channels to the intended nodes for data communication and thus assigns channels based on the probability of a free channel calculated in the estimation phase. The transmission phase thus utilizes the assigned channel without any collision. The protocol uses mathematical estimations that might take extra time and cause extra delays. If nodes are not aware of each other, the number of active nodes may vary from device to device, which can cause extra hidden node issues.

4.4. Adaptive Traffic-Load-Slotted MACA (ATL S-MACA)

For IoT nodes not capable of carrier sensing, a slotted MACA is extended with the help of traffic-load estimation, collectively termed ATL S-MACA. The basic MACA scheme of RTS-CTS-DATA-ACK is slightly modified, and the RTS is adaptively controlled based on the estimation of traffic load [32]. The protocol finds the optimum traffic-load (Gopt) value, which is the maximum achieved load while using current traffic G for estimating it as Gopt/G for the RTS contention. The ATL S-MACA allows all nodes to send RTS packets at the beginning of each slot; thus, collisions are increased while not adding much to network performance in a constrained environment.

4.5. Code-Expanded Random Access (CERA)

CERA is the modification of an already-developed long-term evolution (LTE) protocol called dynamic random-access channel (RACH) resource allocation. The objective of CERA is only to support more devices, as compared to LTE, without increasing resources [33]. The approach utilizes the same orthogonal preambles as those of LTE, but a single phantom codeword is used for multiple frames, causing the preambles to be actuating in a virtually combined frame. When two nodes want an information exchange or channel access, they

select sub-frames from a singular preamble with a random probability. Collision occurs only when two nodes select the same sub-frame in same preamble. The approach is further enhanced by embedding a code work inside the virtual frame.

M2M is not bound to the above-mentioned random access techniques. Enhancement of IEEE 802.11ab for M2M communications [34], fast adaptive slotted ALOHA (FASA) [35], M2M using cellular networks [6], cognitive radio-based M2M communications in OFDMA [36], and polling-based channel access [37] are also implemented for efficient channel access and accomplishing application-constrained channel utilization.

5. IoT-Defined Network/Routing Protocols

5.1. Constrained Nodes and Environment

Characteristics that a normal network node often takes for granted, such as battery power, on-device memory, size of the node, its per unit cost, deployment mechanism, device initialization, and processing resources, are limited and thus to be managed in a constrained node(s). Environments/networks that are created by such nodes are optimized for specific objective(s) and do not support liabilities [29].

These networks use limited bandwidth, processing cycles, and limited communication to avoid depletion of constrained resources. The constrained resources vary with respect to network usage and device specification. Due to the unavailability of either the receiver or the internet, fragmentation is not desirable and constrained networks use small packets with relatively more achievable data rates and throughput. The link characteristics are also asymmetric and packet delivery is never at the same rate in most of the cases. Networks are lossy and nodes cannot rely on acknowledge-less packet transmission, yet must keep the channels and buffers free for other momentary information arrival/exchange. Advanced routing and data acquisition protocols are yet to be implemented in constrained networking because of its low memory and processing capacity.

5.2. Low-Power and Lossy Network (LLN)

LLNs typically comprise constrained nodes with limited energy, processing power, and memory. These nodes are connected to each other using a variety of protocols such as 802.11, 802.15.4, etc. The applications of LLNs are widespread. Ranging from a smart home, where a lightbulb is connected via Bluetooth and a handheld device or a refrigerator that is remotely locked against children, it goes to a big industry where automobile testing is conducted and real-time data are analyzed using a sensory network [30]. Environmental changes, military exercises, flood and disaster management, heating and ventilation systems, data collection and billing, energy and smart grid management, health care, assets tracking, and access control are facilitated using LLNs. The discussion proves the relation and key importance of LLN in IoT. A typical LLN comprises an edge router, a processing point, and connected constrained nodes.

5.3. Routing over Low-Power and Lossy Network (RPL)

An RPL [38] has comparatively less computational complexity and message overhead because it is a distance-vector and source-routing protocol. For the structure of WSN, especially in the use of IoT, the neighborhood must be enquired into for any substantial change in topology or any other restrained parameter. The same characteristics are possessed by distance-vector (DV) protocols. Each node has a vector of connected nodes and their state. This mechanism is used for calculating distances and path costs. DV has relatively less computational complexity because array update is the only process, and it uses distance-vector encounter alteration if any link must be updated. RPL has a self-healing ability, and it has a loop detection and destination-oriented direct acyclic graph (DODAG) repair ability, which is explained in the coming sections.

A node calculates the number of hops and the next node to which it transfers its data. The next hop is termed as “Direction”, whereas the hop count is termed as “Distance” in DV. A node can choose between many possible routes based on the priority of constraints

on the node. The minimum distance vector is maintained in the case of an isomorphic network. Again, the cost is calculated using different route characteristics for reducing cost and increasing networking efficiency.

The dynamic-source routing (DSR) protocol is a path-addressing-based mechanism [31]. Random neighborhood discovery is limited to each node and table information is shared using a data packet that is traversing through the network. The sender may partially embed the route information into the data packet. This mechanism makes the receiver know about all the hops a data packet has traversed before arriving. The mechanism works as a direct acyclic graph; thus, path information, when updated in the packet, takes credit for the reusability of the route. DSR is a destination-oriented protocol. A destination-oriented direct acyclic graph (DODAG) is sometimes used to represent the packet route. The DAG metric is calculated on node limits connectivity based on internal node state (workload, CPU, and memory), energy available, its type (powered or self-charging), and number of hops.

The link metric objects are updated to DAG using connectivity characteristics. These characteristics include throughput, latency, link reliability, and link color. Table 5 lists the node and link constraint objects, and their attribute descriptions. Table 5 clearly explains attribute assignment in the node and link object container. Simulation environments, such as Cooja, NS3 and NS2, OMNET, and OPNET, install these containers on nodes to ensure protocol implantation [28].

Table 5. DAG connectivity characteristics.

Container	Object	Attribute/Usage
Node	State	• CPU usage • Memory and Stack • Sensor state
	Energy	• Definition of power source (a) Powered (b) Battery (c) Scavenger
	Hop Count	• Maximum no. of hops • Total hops traversed by packet • Discovered nodes
Link	Throughput	• Available throughout with respect to environment and connectivity
	Latency	• Acceptable path latency
	Reliability	• Expected/average no. of transmissions. (ETX count). • Link Quality Level (LQL) (a) Unknown = 0 (b) High = 1 (c) Medium = 2 (d) Low = 3
	Color	• Values set by network admin to define and distinct a process

Object Function

RPL follows guidelines from object function (OF) to assign a child/children to a parent node. It also defines the basis of an optimized route. It is the OF that selects the nearest grounded node and metric to reduce the costs of a link. The link metrics are updated according to this instructing function.

5.4. Cognitive-Opportunistic Routing over Low-Power and Lossy Network (CORPL)

The DODAG is built the same way as RP in CORPL [31]. The main difference is an intelligent mechanism on node selection and utilization of node power. Multicasting is implemented using a forwarder set. In addition, coordination between nodes is used to select the best next hop to utilize the total network up-power efficiently. Each node possesses a forwarder set instead of the knowledge about the nodes from which it receives the packet for forwarding, as in RPL. A DODAG information object (DIO) is exchanged in a timely manner with the connected nodes to update the changes taking place in the parent node. Based on this update information, each neighbor updates the forwarder set, as well as prioritizes the information regarding the next hop utilization. It is notable that CORPL is a node-based cognition and sensing technique, and it does not guarantee channel utilization.

5.5. Channel-Aware Routing Protocol (CARP)

Channel-aware routing is normally used in shared media for distributed systems. These distributed systems are somewhat modular networks that coexist in the same geographical area but comprise different data and/or network characteristics/requirements [33].

Initially, the protocol was developed for underwater communication [34]. The packet size is relatively small and feasible for IoT applications including sensory networks. It is a semi-query-based data collection mechanism that inherently depends upon the path of the sink node from the sensor. Initially, a sink node traverses a packet from sink to all the supplier nodes of the network. In the forwarding phase, packets of the sensor are routed using hop-by-hop method to the end node.

Based on the historical successful packet deliveries, the paths are defined by the node for the next hop. There is no acknowledgement packet from the receiver side, so ultimately, a network floods the paths in its initial deliveries. Each next hop is determined independently, which allows specific channel selection. After a significant change in the network topology, the forwarding paths are all discarded, and then new paths must be declared by broadcasting "Hello" packets from the sink. Thus, the reusability of the previously discovered path is not utilized.

5.6. Enhanced CARP (E-CARP)

Enhanced CARP (E-CARP) [34], which allows the sink to store the previous data from the sensors, sends a ping packet using the same path to the sensor (sink to sensor). Upon receiving this request by the sensor, the sensor follows the same path to send more data. E-CARP reduces communication overheads by feeding the media and hops from re-broadcasting the same type of information more than once during an application.

6. IoT Encapsulation Network Layer Standards

IoT prefers IPv6 for its unique property of identifying each connected device and setting different payloads, protocols, and many other standards defined by the IPv6 header. The problem arises when the availability of a channel is limited or the constraints limit long headers. In such a scenario, IEEE, ITU-T, and IETF seek protocols for secure fragmentation and encapsulation of the IPv6 header. The section comprises of such network-layer encapsulation protocols currently employed by IoT community.

6.1. 6LoWPAN

A fixed IPv6 has 320 bits (320 octets). The 16-bit payload section defines the number of bits attached to the header. This makes it go from 1 byte up to one less 4 GB (jumbo

gram). IPv6 over a low-power wireless personal area network (6LoWPAN) is a widely used standard. The underlying protocol is IEEE 802.15.4, which can convert large data streams into small packets, thus converting IPv6 headers into small chunks that do not exceed 128 bytes. Fragmentation header (11) of 6LoWPAN handles this compression and breakdown. Various topologies are being supported by 6LoWPAN that include mesh and star in abundance. These topologies are also supported by different 6LoWPAN headers that include mesh header (10) and dispatch header (01). No header (00) is used for single-hop data delivery. Here, any frame that does not follow 6LoWPAN specifications is automatically discarded by this receiver [35].

Data frames, once defined by a unique 6LoWPAN header, are multi-casted in the neighborhood where each connected node receives and analyzes the frame and does processing according to the header type.

Characteristics of 6LoWPAN

- 6LoWPAN [35] has a long sleep mode, enabling smart battery consumption and network sustainability;
- Different length addressing enables network scalability and variable packet length for a constrained environment;
- Its relatively lower bandwidth consumption makes it scalable and supports reliable packet delivery because of low SNR, and the reliability is not managed by any other algorithm, so low bandwidth facilitated it;
- Mobility is inherently supported in 6LoWPAN for its dynamic channel assignment and local addressing [6];
- The deployment costs are relatively low and there is no massive processing over sensor nodes.

6.2. 6Lo

IPv6 over a network of resource-constrained nodes (6Lo) covers data-link layers that are not addressed by 6LoWPAN and 6TiSCH (discussed in the coming section). IEEE 802.15.4 and IEEE 802.15.4e do not cover protocols and they are covered in 6Lo. Bluetooth technology, near-field communication (NFC), industrial sensory network, and 1 GHz Wi-Fi (802.11ah) are modified for the development of 6Lo [39]. Recent advancements in 6Lo are tabulated here in Table 6.

Table 6. 6Lo Frontiers.

Technology	Description and Characteristics	Area/Application
IPv6 over Bluetooth Low-Energy Mesh Network [36]	Enabled in Bluetooth v4.1, 6LoWPAN, Peer-to-Peer comm., Fragmentations, link-local addresses, stateless IPv6 auto-configuration, neighbor discovery, header compression	Mobile healthcare, networked computing, headsets and enhancements, notebooks, sensing smart meters, appliances, etc.
IPv6 over IEEE RS-485 Master Slave/Token Passing (MS/TP) networks [40]	Master/Slave/Pole control frames, 256/2 addresses for master and slave, uses LoBAC encapsulation	Used for supporting large payloads in building automation and control networks within 1 km diameter
IPv6 over DECT/ULE [37]	Low-Power Operation, Extended Security (64 bit AES), dedicated radio frequency link	Home automation, climate control
IPv6 over NFC [41]	Peer to peer, read/write mode, and card emulation modes, 80% packet success rate, 1 to 10 cm range	Payments via phone, passive and active tags based electronic token system, control NFC devices with active tags
IPv6 over IEEE 802.11ah [33,37]	Uses extended ISM band (1 GHz), typically with wider range and low SNR, multi hop relay and mesh networking, Supports MIMO	Smart sensors and meters, backhaul aggregation using 802.11ab routers/gateways, extended hotspot range and cellular offloading using sub-1 GHz band
IPv6 over Wireless Networks for Industrial Automation Process Automation (WIA-PA) [35,39]	Improved stack architecture, auto-configuration	Connects industry to the internet, connecting individual process data together for management in industry

6.3. LoBAC

It is the only protocol that is wired on 6Lo PHY. The 6Lo Z-Wave, BTLE, and NFC are based on this proposal [36].

6.4. 6TiSCH

The IPv6-based time-slotted channel-hopping (TSCH) mode is an extension of the IEEE 802.15.4 standard for industrial automation and process control. It converges operational technology (OT) with information technology. The time-slotted channel hopping makes 802.15.4 (TSCH) introduce reliability to the communication in terms of reduced multi-path fading because of its narrow bandwidth channels [42]. It also extends battery life on constrained nodes. A specific portion of bandwidth is allotted for the communication of a node with its neighbor for a fixed interval of time. The standard does not guarantee any scheduling mechanism, hence it is not allowing maximum network throughput in its current form. When neighbors use data for the same application, scheduling poses a relatively bigger network optimization problem and reduces the flexibility of the process. They must be programmed to access media with respect to the information gathered from surrounding nodes. In such a case, a centralized scheduling mechanism can be proposed, but to reduce power consumption, the centralized node must adopt some constraint-based distribution of decision making to one of its high-residual-energy neighbors.

6.5. IPv6 over G.9959

ITU-T G.9959 specifies transmission and receiving for short-range narrow-band radio communication. It is a PHY/MAC layer channel-assignment mechanism. Link quality for receiving frame, selection of frequency band, data frames, power and mode management of radio transceivers and clear channel assessment are collectively managed by its two services, the PHY data-service access point (PD-SAP) and the PHY management service interfacing layer transmission entry (PLME). The PHY layer wraps the data in radio frequency (RF) frames. Channel numbering and frequency assignment are also done by G.9959 [43].

IPv6 over G.9959 supports mesh routing and route-over routing for TCP, UDP, and ICMP application types. Header compression of these IPs is done using RFC6283 6LoWPAN standards. G.9959 is responsible for logic link control (LLC) segmentation and reassembling (SAR) [44], media access control (MAC), and physical interface.

6.6. IPv6 over Bluetooth Low Energy (BLE)

IPv6 provides larger pools of addresses. Local areas nowadays use WLAN that in the majority comprise Wi-Fi-based connectivity. Though the technology is cheap, it consumes much battery power and abruptly drains power. On the other hand, Bluetooth technology gives almost the same range with sufficiently less power drainage [36]. This allows the local NAT fields to be occupied with Bluetooth connectivity rather than using Wi-Fi. Body area networks (BANs) majorly consist of sensing nodes that require less bandwidth and throughput. IPv6 over BLE is a smart choice for such low data-rate transmissions. Bluetooth watches, gears, computer accessories, multimedia, and maybe industry-grade equipment use IPv6-powered BLE. IPv6 enables each node to directly talk to the internet.

The following are the issues that need to be addressed before bringing this technology to the consumer level.

- Bluetooth is available in almost every handheld smartphone. This makes the Bluetooth-based network relatively less secure and more vulnerable.
- Only enterprise routers are enabled with low-power Bluetooth connectivity and bringing IoT into use would need more plugging of Bluetooth hubs into our normal 802.11 routers.
- IoT mainly integrates using multi-hop topologies, and thus, every device would be served as a repeater to transfer its data to the main BLE hub, which is a software-based problem. This problem must be finalized before implementing BLE for dense networks to avoid battery drainage and expand the physical access range.

7. IoT-Defined Application Protocols

The application layer for the IoT is also thoroughly explored in [45]. The study conducted examines IETFs CoAP, IBMs MQTT, HTML 5s Web socket, XMPP, RESTFUL Services, and AMQP for the application-layer interconnectivity of IoT. These platforms are based on existing technologies and partially fulfill IoT requirements [46].

7.1. Constrained-Application Protocols (CoAP)

It is a request/response protocol. The internet geeks are familiar with get-and-post-based HTTP (hyper-text transfer protocol) from the internet protocol (IP). CoAP was documented by the Internet Engineering Task Force (IETF) in 2014 [47] for constrained, i.e., low-power and small battery size, devices. CoAP enables edge devices and nodes to connect and share information with centralized systems using lower power. CoAP is a two-layered protocol that combines the characteristics of UDP with TCP. The first layer is the messaging layer that declares the connection's reliability as UDP or somewhat like UDP. The second layer is a request/response layer used for interaction between the communicating nodes. Note that CoAP can be used both within a single WSN-based IoT and outside the network when eternal nodes are essential for process control and/or data.

CoAP packets/messages are of six distinct types, depending upon the mode of communication and information integrity. These message formats consist of nonconformable messages, conformable messages, acknowledgment messages, response messages, and reset messages. The messages are explained underneath.

7.1.1. Conformable Message

This type of message assures the transfer of the message to the sender. An acknowledge message propagates back to the sender, confirming the safe arrival of the packet at the receiver side. This message protocol is thus useful in terms of data integrity, but the bidirectional communication flow is an overhead on the network, as well as on battery constraints. Because IoT-based WSN always have intermediate nodes for a communication mesh, this type of messaging service cannot be used for applications where nodes do not possess infinite/sufficient power.

7.1.2. Nonconformable Message

For avoiding the overheads discussed in the above section, a nonconformable message is not bi-directional communication. CoAP may use error detection schemes for avoiding and discarding the erroneous data on the receiver side, yet the sender node is not aware of the packet loss and spoofing.

7.1.3. Acknowledgment Message

When the information in a conformable or nonconformable message packet is not consistent, the receiver sends an acknowledge/warning message for a resend. This type of message comes under this category. Acknowledgment messages can relatively reduce the network overhead when used with nonconformable messages; hence, the network power usage is almost between the conformable and nonconformable messaging schemes.

7.1.4. Response Message

Response can be either used separately from the acknowledgment message or it can be embedded within the response. So, based on these scenarios, the response messages are divided into two sub-categories, i.e., separate response and piggyback response.

7.1.5. Separate Response

When an IoT node needs information from another network node/centralized database, it sends a GET command using CoAP to the addressee node. The receiver of the message thus posts the required data using the same CoAP to the sender. When the network power constraints are not optimized, the network uses two separate messages for responding to

the query. First, it sends an acknowledgment message to the sender that includes information about the query the receiver (server/node) has encountered. The response is then posted as a separate packet back to the sender with a totally new header. Such messaging is known as separate messaging.

7.1.6. Piggyback Response

Here, the receiver directly includes the query result in the acknowledgment message, and thus, two separate messages are avoided. If the query is lost in the network instead of reaching the server, the server acknowledgment is not received by the sender, and hence, the query is to be made again. If the query is answered and it reaches the receiver, the receiver may consider it as tempered data, therefore embedding acknowledgment in the response is a good mechanism for assuring network security. It also decreases network overhead and consumes less bandwidth, as well as energy, with an increase in the CPU's processing time on both sides.

CoAP applies its reliability mechanism to reduce network overhead and processing on each node and conserves bandwidth and battery. The two-conceptual-layered protocol is based on representational state transfer (REST), which supports request/response models such as HTTP and HTTPs and thus enables both secure and nonsecure information exchange. The messaging layer is controlled by this request response layer giving reliability to the data transfer and queries. CoAP also uses low memory.

The latency is inherited from REST, which inherits it from the UDP working mechanism. Packets are often discarded by network nodes and the data field is also limited. These limitations readily prove the inability of the CoAP over REST for complex or continuous big data types.

7.2. Messaging Queue Telemetry Transport (MQTT)

As the name indicates, MQTT is a queuing-based protocol that requires minimum bandwidth because of its design [48]. Its low computational cost and open-source code make it versatile amongst constrained IoT applications. It is a client–server communication protocol that is based on publish/subscribe protocol. Unlike the request/response mechanism in CoAP [49], the MQTT focuses on onetime subscription or one session registration and then unidirectional data flow from the server side to its client(s). Most of the IoT nodes use this protocol for machine-to-machine (M2M) communication. The protocol serves the TCP/IP layer for its communication. Data integrity and session are monitored in MQTT; thus, the data received are reliable and error free.

MQTT supports unicast, multicast, and broadcast messaging. That is, it can flood the network with the desired message, or it can securely transfer the message to any node/server on demand. A timely update feature is used for subscriptions, which reduces network overheads. Notification mechanisms are also employed for abnormal activity in the network or emergency situations.

The quality-of-service level is achieved by considering an MQTT protocol such as TCP/IP. The server node, when subscribed to for a specific task, publishes messages based on the specifications of the subscription. These subscriptions can be of three kinds, each explained below.

7.2.1. Exactly One Delivery Subscription

Due to its highest level of QoS, the protocol must ensure the packet delivery using an acknowledgment message after its delivery. A route is traced from the server to the client for message delivery, and when the link is established, the packet is sent. Another mechanism is a "waiting for delivery report", in which the server waits after specific information is shared with the client.

7.2.2. At Most One Delivery Subscription

The receiver/client may not receive the packet at all in this type of subscription. This facilitates the sender to reduce its power consumption. On the other hand, acknowledgment from the receiver is also not required. The queuing is not required when a message is being sent to a single user. A message can be broadcast in such a case without any delivery report requirements by the server.

7.2.3. At Least One Delivery Subscription

As the title indicates, the subscription must assure that the client gets the required data. The difference between at least one delivery subscription and exactly one delivery subscription is that in at least one delivery of a message, there may or may not be duplicate packets reaching the client from different routes. Exactly one message delivery discards the packets as they are received. Exactly one delivery has more precision than this type of delivery subscription.

MQTT was developed by IBM in 2013 after studying the impact of CoAP; thus, it outperforms CoAP in many characteristics. It requires less computational power and memory as compared to CoAP. The MQTT-based IoT has comparatively lower latency in packet delivery, and it utilizes the network bandwidth efficiently because of its queuing mechanism. MQTT is also dedicated to simple and small data packets. It is used in networks that repeatedly require the same type of data from network nodes; thus, it possesses high data as well as sampling rate. This limits its application to only simple networks. As based on TCP/IP, the MQTT cannot be used for real-time applications as well.

7.3. Extensible Messaging and Presence Protocol (XMPP)

Extended messaging is dealt with in the dynamic host control protocol (DHCP) on a network layer of the OSI model for IP. Thus, XMPP [50] is the most advanced form of IoT networking based on the characteristics of DHCP, i.e., the message length can be extended for a single packet based on the network specification and data requirement [17]. Furthermore, each node is dealt with as a separate entity using three identification marks commonly known as node ID, source ID, and cache type. These attributes are used by the concentrator/gateway that serves as a primary router for the local IoT. The attributes are unique for each node within the local network, so they can be reached by a client outside the network.

XMPP registers things on demand with a concentrator. The attributes are accompanied by data as string and data as numeric tags [47]. The process registers each node that is being communicated in the IoT; thus, tagging is important because the registry might be searched for certain terms and comparisons should be made according to the tag type accordingly for fast routing and packet tracing. The process allows things to be registered that are only known by the owner of the things. Security and data integrity is thus never breached when using this mechanism.

Tags or meta data are used to avoid re-registration of the same node with the owner node. In addition, as the network can be formed by any node, the owner can thus dedicate itself for a specific group of nodes it is surrounding. XEP documentation suggests five different architectures for an XMPP-based network.

- **xep-0000-IoT-BatteryPoweredSensors:** this protocol defines the way for handling power constraints by the IoT. Such a protocol is widely used in intermittently available network nodes.
- **xep-0000-IoT-Events:** this extension gives privileges to network nodes for defining their set of events for subscription and response.
- **xep-0000-IoT-Interoperability:** IoT may or may not consist of the same types of devices. This protocol definition is used for setting priorities on defining communication rate and type between different devices that are working on slightly different technology/data.
- **xep-0000-IoT-Multicast:** it defines multicasting methods for efficient transmission of data between sensors in a network.

- **xep-0000-IoT-PubSub:** publications that are the result of subscriptions by anodes for specific data in specific formats are efficiently done using this method.
- **xep-0000-IoT-Chat:** user-friendly human-to-machine interface is designed using this tool. The basic architecture of IoT is used as the underlying platform for meaningful information exchange because of human-defined queries. Automatic response is also provided by using chat options in IoT XMPP.

XMPP is relatively simple because it uses high-level XML scripts. Though it is the most suitable protocol for heterogeneous IoT, yet it has four major drawbacks. Its commands are more specified, and thus, it uses more CPU power for processing each command. The bandwidth requirement can be controlled by the underlying XML scripts but cannot be as optimized as other protocols can be. QoS is not guaranteed, and the communication is restricted to text data. Visualization is relatively easy for users by using XMPP. Interoperability and bidirectional communication between man–machine are the key features of XMPP.

7.4. Representational State Transfer (REST) Mechanism

REST is a better alternative of CoAP for its cache capabilities, HTML support, and stateless architecture. REST was developed in 2000 for defining network resources and addressing these resources for connectivity. REST is an application or style of software architecture employed for IoT, which is the reason why it is often termed as RESTful architecture or REST-style applications. IoT that is designed for web applications such as multimedia devices in smart homes and other data-communication-based devices that publish or get orders from web servers uses RESTful application architecture. Cloud computing, remote computing, and service-oriented architecture all prioritize REST. Restful API serves as the interface between two computer systems that is used for information exchange securely over the internet.

The main characters of RESTful application are as follows.

7.4.1. Distributed Resources

Based on the functionality and state, pools of resources are listed using REST. For example, a network is divided into three pools. The first pool sends sensor data, the second one receives the data, and the third pool does the calculation and sends it to actuators. In the example, REST will uniquely address the resources. After that, it will assign states and functionality to each node of the network and based on that functionality, the publish/request or process orders will be managed.

7.4.2. Software Commands

As RESTful architecture usually uses HTTP commands, so GET, PUT, POST, and DELETE commands are used internally by the IoT nodes. The web-based architecture allows the pools to receive HTTP requests using their unique addresses and to act accordingly.

7.4.3. Protocol

It is a client/server, stateless, supporting caching layered protocol. The caching capability makes the nodes computationally expensive. This layered architecture supports a range of communication technologies because it is a software-based solution to the IoT problem. Caching also makes the IoT delay-tolerant.

REST is the most widely used service over IoT and the internet by implementing HTTP. It supports different types of applications and has security protocols such as TLS/SSL. The maximum payload limit for a single API is 45 MB, whereas the URL size is limited to 7 KB. HTTP is readily used for secure data transfer and messaging, both inside and outside the network. IT is easily implementable, and interaction of man-to-machine is easier.

7.5. Advance Messaging Queuing Protocol (AMQP)

AMQP has the tendency to support heterogeneous networks of nonwearable sensors, so its potential customers are industry machines and businesses. It supports messaging

publish/post in different technologies. Its reliability is enhanced by keeping the messaging to a fixed data and format, which is also the requirement of industry. The security, as well as performance, increases when messaging and requests are kept to a fixed format. The receiver knows the pattern of the message and can extract the data field from a fixed message structure. It is highly scalable for its heterogeneous networking and publish/subscribed approach [51].

The communication reliability has a level equal to TCP/IP. This means that the protocol for delivery of data for at least once, exactly once, and at most once guarantees the communication reliability. The protocol spans multiple operating systems simultaneously, and thus, the complexity in developing different industrial-level applications is reduced. AMQP is the backbone of modern data and acquisition systems and supports industrial IoT that includes the autonomous unmanned operation of machines and the self-healing capabilities of systems.

Other proprietary messaging protocols are limited to single-user decision making for information exchange. Thus, integration for business partners is limited. Nodes must limit their functionality if they want heterogeneous platforms-based networking. AMQP can integrate hundreds of systems with quick solutions such as moving a ledger to the cloud or to some other network node, sharing daily meeting schedules, and making an appointment next week without disturbing the routine work. IoT becomes more flexible while using AMQP, and the addition of new clients is on the fingertips [51].

AMQP-based architecture possesses additional features that are not supported by HTTP client/server communication for the request/publish mechanism, whereas HTTP has certain features that are eliminated in AMQP for reliability, performance, security, and scalability. Some of these features are tabulated as under (Table 7).

Table 7. Features that optimize AMQP for scalability, security, and homogeneous networking solutions.

Protocol Feature	AMPQ Status
Caching Read	Eliminated
Put	Eliminated
Delete	Eliminated
Content Filtering	Added
Header (Typed)	Added
Transactions	Added
Simple Authentication (SASL)	Added
Symmetric Protocol	Added
Socket Multiplexing	Added
Out-of-order Notification	Added
Server-initiated Transfer	Added
Store and Forward	Added
Publish and Subscribe	Added
Defined Error Recovery	Added
Well-defined Address	Added
Content-based Routing	Added
Credit-based Flow Control	Added

According to Gartner [6], an AMQP core can bundle up properties of protocols such as TCP/IP, SMTP, and HTTP and, thus, can form a standard IoT-based communication protocol for achieving software as a service (SaaS) and platform as a service (PaaS). This might reduce costs, as well as setup time, for deploying certain web-based business mod-

els. AMQP has all the ammunition to increase opportunities in technology, business, government, and industry.

Comparison with MQTT

IBM's MQTT is a single-core protocol, and this cannot compete with AMQP, which is the result of a collaboration of much user-driven firmware (UDF) [52].

1. MQTT lacks durability and crashes without possessing archives for retrieval and recovery.
2. User must communicate simultaneously because no queueing mechanism is defined by the protocol in MQTT.
3. Windows Communication Foundation (WCF) and Java Message Service (JMS) are not configurable in MQTT, but AMQP supports Linux, Windows, and Solaris platforms for server configuration, Verbose, and messaging.
4. Transaction monitors and databases support XA-Transactions for its guaranteed features of update on all the connected data upon any update. MQTT does not provide support for its memoryless broadcasting features and lack of queuing.
5. Kerberos authentication, which is based on active dictionary (AD) key underpinning, has the facility to avoid the usernames and passwords traveling via network. The mechanism uses tickets to get entry to the remote server or database, which is based on an encrypted hash containing time stamps. A domain controller (DC) then receives the access request and grants a ticket-granting ticket (TGT), which can be used for the application server as login and session key. The facility of Kerberos is limited to AMQP; thus, private keys can be attacked in MQTT.
6. There is no flow-control mechanism and selective acknowledgment to pause and resume applications in MQTT.
7. Multiplexing is relatively easy in AMQP because MQTT supports the at most one, exactly one, and at least one message delivery schemes. These schemes may be valid for a single receiver, but the result may be different when much information is incorporated for different network nodes, resulting in network overhead and inefficient use of bandwidth.
8. Multiplexing in MQTT can cause flooding, and error control is also affected.
9. AMQP has flexible topology, and the information can traverse the internet or any broker AMQP.
10. AMQP supports ASP.NET, Java, and PHP, while MQTT is dedicated for small data packets and frequent server replies and data updates.

If we compare AMQP with other protocols such as RESTful, MQTT, XMPP, and CoAP, AMQP proves to be more advanced and prone to develop more for its modular inter-platform secure connectivity. Despite this, AMQP lacks support for constrained environment and real-time applications, which is indeed important from an IoT perspective. It also does not support an automatic route-discovery mechanism and IoT node-discovery mechanism.

7.6. Web Socket

Web socket is a low-latency client/server communication that is standardized on HTTP for TCP. This protocol allows the client to get real-time updates from the server without requesting a new session, till the session is ended. As the uninterrupted communication of IoT devices with remote servers is very important, the web socket interface provides TCP-based communication of IoT nodes with remote servers. This communication is available for data updates on the server and from the server. Web socket provides a full duplex communication channel for remotely connected web servers and nodes. The data that are communicated using asynchronous client/server communication start exchanging without publish/subscribe and request/response mechanisms [52].

The remote host starts its session after building a handshake from the client; this server gets busy only after communication is started. The client, on the other hand, can draw

routes to the server without any external third-party tool. The session terminates when both server and client agree upon it.

MQTT and XMPP are usually for client-to-server communications, and web socket is fairly new in this field. However, web socket seems to be more powerful, as the literature suggests. Web socket uses a single web-socket library at the client end for initiating web-socket-based real-time communication. The library can be used using programming language like Java or Python.

Web socket is efficient for the following reasons.

1. There is no real-time lag or polling interval while in communication with server.
2. There is no need of bidirectional authentication for each message. A request for each packet is not necessary while connections are set up beforehand.
3. Data are only sent when they are needed, avoiding network congestions and bandwidth deficiency.
4. After the client connects with a remote server, no headers are used for communication purposes and only data fields are shared.

7.7. Data Distribution Service (DDS)

IoT application is for certain QoS. These QoS include reliability/durability, security/encryption, and queueing/prioritizing. The data that are transmitted between the server and the receiver are known as topics. The methods of DataReader and DataWriter in subscriber and publisher, respectively, generate the data on the topic. Fundamentally, DDS is all about data. It does not control nodes, machines, or processes. The applications of DDS are designed for tagging and assignment in data and their secure or unrestricted access [52].

DDS is an interoperable data exchange protocol that is used for real-time secure data transactions. It works on publish/subscribe message delivery time, durability, and security. If the subscriber wants the publisher to send the data in less time, the subscriber can embed the path/route information in the message. The publisher can also find the shortest path to the subscriber for achieving QoS. Similarly, updating security protocols makes the receiver a less visible intruder, and data distribution can also be done with some encryption for secure information exchange.

This protocol is a good choice for IoT and machine-to-machine communication models if QoS is the main concern. DSS consists of a local database that is the collection of all topics handled by the domain. This database is termed the global data space (GDS). This local store gives the illusion to the connected applications that they can access the whole data (in actuality, the whole data concerning the application is also known as the ticket). Nodes connected to this local store possess data that are instantaneously/timely needed. The copies of data are respectively sent to the application after the application subscribes to it.

Global space shares many different technologies such as sensors, mobile communication, embedded systems, and Windows- and Linux-based platforms and cloud applications. The communication is fast and with low latency because tickets can be accessed at any instant by any communicating node with attached QoS specification. DSS always provide dynamic discovery of publisher and subscriber and are somewhat called plug 'n' play-mechanism-based applications.

The architecture makes it scalable and the administrative layer can control its subsequent fogs (data spaces) and edges (connected machines). Only this protocol can add any type of node to an existing pool of IoT and provides a simplified framework for distributed system development [52].

The following are some of the qualities of DDS for which modern IoT supports its core development.

7.7.1. Modeling and Reusability of Modules

Once a topic is created on GDS, it can be used by other developers while they are writing their application code for certain nodes in IoT. This increases the reusability of the

build modules. Each topic consists of information that is domain-specific; thus, getting information is quite easier.

7.7.2. Development of IoT

The network development is similar because the GDS allows input and output data for any application that points toward a domain-specific topic. This allows the network modelers to focus on the outcome rather than setting up basic functionalities of the IoT.

GDS can also supply filtered data and this facility relatively eases the task of nodes with respect to computational complexity and robustness. DDS also supports redundant customers, i.e., the same data are already available on GDS for supply to remote clients without increasing application complexity.

7.7.3. System Design with Ease

DDS allows devices to share a common server resource as GDS. This model separates both time-based processes and frequency-based processes in a modular manner. Local databases use a data-centric approach to use the GDS rather than multiplexing information for many connected users and evaluation of data based on requests. It is this onetime configuration for data reusability and system components that can be added in a plug 'n' play manner.

7.7.4. Enhanced Yet Simplified Security

DDS API allows a "many-to-many" session update algorithm with channel security. Simple DDS searches data by either name or value and makes connections. Secure DDS documents queries and enforces certain security schemes that are already developed and coded in API for establishing a secure channel and system. DDS also supports secure multicasting. The publishers and the subscribers are connected to the topic, while the topic name and its configuration are used as security terms for information exchange. This enforces dataflow between a known publisher and a known subscriber without any intruder or data misplacement.

Table 8 gives the qualitative comparison of different IoT communication protocols. It clearly provides evidence that DDS is a relatively better choice for decentralized fault tolerant networking and data security with configurable encryption and encoding.

Table 8. Qualitative comparison of DDS-, AMQP-, CoAP-, and MQTT-based IoT networking.

	Transport	Communication	Connectivity Nodes	Network Discovery and Routing	Data Security	Data Centricity	Data Prioritizing	Fault Tolerance
DDS	TCP/UDP	Publish/Subscribe, Request/Reply	D2D, D2C, C2C	Yes (Content-based routing)	DTLS	Declared Encoding	Data-based transport properties assignment	Decentralized
AMQP	TCP/IP	Peer-to-Peer messaging in same technology	D2D, D2C, C2C	Yes	Kerberos Ticket based, TLS	Encoding	None	Decentralized
CoAP	UDP/IP	REST (Request/Reply)	D2D	Yes	DTLS	Encoding	None	Decentralized
MQTT	TCP/IP	Publish/Subscribe	D2C	No	TLS	Encoding	Session-based queuing	Single Point Failure based (SPoF)

8. Discussion and Open Issues

8.1. Discussion

The IoT enables expansion of smart cities by employing protocols at various levels. Application-layer protocols, physical and MAC layer protocols, and IPv6-based networking-layer protocols are designed in such a manner that certain task-oriented solutions are provided and enhanced. Although the developed and designed protocols are sufficient for incurring the current needs of services that the smart cities are intended to own, the

protocols need certain enhancements for the expanding usage of data, as well as of data-driven IoT nodes.

According to Gartner's speculation, the IoT will ultimately take 50% of the management share in industry. This means that the current Industry 2.0 and Industry 3.0 will be reshaped by the end of this decade, i.e., 2030. Thus, researchers should be curious not only about the increased demand for data communication channels, but they must also address the issues that arise with the generation of a large amount of data traffic in terms of its collection, storage, analyzing, and even discarding or replacing. Processing power will exponentially increase both at the end nodes and also at the management and data servers.

As the work focuses on autonomous systems, application layer protocols are thoroughly examined for their use, as well as for the complications that may arise while dealing with large-scale data and wider network scenarios in smart cities. The sensing and data acquisition technologies that are used in smart cities require expandable solutions and relatively constant network throughput. Such networks also use adequate processing power and energy that must be addressed on the application level via certain constrained energy-based application protocols.

Constrained nodes and environment, low-power and lossy networks, channel-aware and data-aware routing, and state-, energy-, and hop-count-based networking are explored. Various metrics of the stated ingredients are listed in the work that may undergo deep research to collaborate with the engineers of Industry 3.0, Industry 4.0, and smart cities.

Finally, the addressing of a large quantity of nodes and on-time delivery of information is explored via encapsulation technique perspectives inside 6Lo frontiers. The area of applications ranges from mobile healthcare, smart metering, enhancement to visual and audio communication, building automation, home automation and climate control, payment systems, secure communication for credit cards and NFC devices, network backhaul aggregation and cellular offloading, and finally industry management and automatic functionality. These areas and applications use heterogeneous modes of communication, while the data centers and logical networks stay unchanged in the majority of cases. For such scenarios, the physical network must cope with the increase in data over the network, as well as more interfaces of IoT nodes.

8.2. Open Issues in Application-Defined IoT Protocols

The IoT applications can be developed by using various protocols across different components of the aggregate solution. Several techniques are used that combine good attributes of application-, physical-, and network-layer protocols [9].

1. IoT-based sensing and acquisition technologies are connected to data stores on the back end; these applications need to support big data using MongoDB, Hadoop, Postgres, or ScaleDB, a variant of MySQL for high velocity storage. These need accessing techniques development and addressing.
2. MQTT and AMQP are different technology stacks or protocols. They both use messaging bus. It is, however, possible to take MQTT and then go from service to server, but AMQP is more popular for server-to-server scenarios. Therefore, applications are to be developed by swapping MQTT and AMQP.
3. If there is an anomaly in the data or an event that requires more than a single poll, then it is possible to instantiate a streaming scenario that generates a time series or a stream of data using MQTT. Thus, bandwidth needs to be optimized for carrying such sessions.
4. IoT can be distributed in geographically separate regions and the decision making can be time consuming. A need for modular connectivity is experienced in the applications where the data from one region is in real time, as well as in delayed use in another region.
5. With continuous sessions of information exchange, the algorithms must be adaptive if battery-powered nodes are considered. Such nodes should be then treated with battery-constrained accessing protocols, such as CoAP with slight modifications.

6. Mobility, reliability, security, privacy, energy efficiency, scalability, management, availability, and interoperability issues are in the rack toward the development of smart cities.
7. Recent additions of blockchain technology, AI, machine learning, non-fungible tokens (NFTs), augmented reality, massively multiplayer online gaming (MMOGs), real-time video summarization, and decision marking through big data analytics must be well-addressed in the development of smart cities.
8. Certain security risks are involved with the deployment of IoT while expanding smart cities. These risks may include but are not limited to data breaches at local networks, networks being compromised by intruders, eavesdropping, network congestion by unwanted traffic, distributed denial of service (DDoS) attacks, etc. Irradicating and limiting such risks should be kept as the focal point of advanced research in expanding smart city applications and the development of Industry 4.0 applications, as well as equipment.
9. In addition, big data can be collected and sold back as analytics, and yields can be increased through smart monitoring, and so on.

8.3. Open Issues in IoT for Smart Cities and Its Solutions in 6G

The 6G system can be equipped with the necessary support to significantly improve the performance of smart cities [53]. It will provide the required infrastructure and data transmission rates to support massive deployment of smart sensors and fully functioning devices (FFDs). The 6G also will enable real-time processing of data, as well as data analytics in IoT. The following are the potential open issues that will drive the attention of future research in smart cities and IoT with respect to 6G: 6G requires the support of a large number of IoT nodes and enough bandwidth to support these devices. The capacity of smart cities is directly associated with large-scale device and data-rate support in 6G.

1. As IoT requires low-power usage-based networking, future 6G will need this support for enabling low-power wide-area networking (LPWAN).
2. Real-time IoT-based control and information exchange systems for critical operation in industries require low-latency-based communication. Thus, to enable low latency and improve reliability, 6G will need ultra-reliable low-latency communication (URLLC).
3. Multi-access edge computing (MEC) is the support required for enabling end nodes in remote IoTs to perform various computations, the results and status of which require multiple access support; 6G will need this MEC to achieve appreciating edge computing.
4. The 6G system requires modular connectivity and network slices to support exclusive IoT tunnels and dissociated processes.
5. IoT devices are exposed to intruders in a network because of its constrained messaging protocols; 6G will need to provide a mechanism to support the encapsulation of such barely protected data and deny unauthorized usage.
6. The 6G system will require an advanced support and management mechanism to efficiently operate IoT networks in smart environments.
7. Technologies such as BLE PANs, Zigbee, LoRaWAN, etc. require integration for the smooth operation of various transactions and network/process integrations; 6G must ensure this interoperability of the new advancements with such existing technologies.
8. Smart cities and environments require dynamic configuration for their ubiquitous operation. To enable such dynamic and retainment, 6G must possess the required tools to enable the automatic configuration of a network and its on-demand reorganization.
9. Generally, IoTs require various levels of QoS. To improve the operation of IoTs in smart environments, 6G needs the support of advanced QoS mechanisms.

9. Conclusions

This work has explored the role and current status of various MAC/PHY layer technologies and protocols, networking-layer protocols, and application-layer protocols in the

development of smart cities. Technologies that include Wi-Fi, cellular networks, Bluetooth, RFID, NFC, Z-Wave, Zigbee, Sigfox, and LoRa support LoRaWAN, 6Lo, BLE, 6loWPAN, Wi-Fi, HaLow, etc. Based on the information collected through Table 3, we conclude that the general availability of Wi-Fi and cellular networks has remained unbeatable. Moreover, technologies such as Z-Wave, SigFox, and LoRa have the potential to reduce the responsibility of cellular networks for low-data-rate-based applications. The data security concerns in cellular and Wi-Fi networks are also adequately addressed in Z-wave and SigFox for wide area communication. On the other hand, these MAC layer protocols are responsible for handing internet protocols such as IPv4- and IPv6-based TCP and UDP connections. These networking protocols support application layer protocols such as MQTT, CoAP, XMPP, AMQP, DDS, LWM2M (lightweight M2M), OPC UA, STOMP, HTTP/HTTPS, etc. Operations like smart grid, smart building management, smart agriculture, smart healthcare, smart public safety, smart energy management and scheduling, smart waste management, smart water management, smart lighting, and smart transportation systems are only possible if the backbone of the city is supported by the aforementioned physical layer infrastructure, MAC layer, and application-layer-protocols-based networking schemes.

We conclude that the application-layer protocols, MAC layer, physical-layer protocols, and network-layer protocols play importance roles in enabling smart cities. The application layer enables the infrastructure to develop and implement specific applications such as smart lighting, supply chain management, energy management, traffic management, etc. Such applications directly improve smart cities. Likewise, the MAC layer enables efficient communication for the stated applications for which the physical layer is responsible for the provision of the required infrastructure for wireless and wired communications. The need of 6G is also realized with time as its development aids the expansion of smart technologies and smart environments for its growing connectivity and enhanced data rates.

Despite this, the current advancements in IoT for smart cities, such as the use of massive connectivity environments, large inputs, and utilization of big data analytics, edge computing and machine learning, are helping to gain command over the challenges faced in the expansion of smart cities. This way, smart cities will become more intelligent, productive, and sustainable.

Author Contributions: Conceptualization, J.A., C.H. and M.H.Z.; methodology, J.A.; software, J.A.; validation, J.A., C.H. and M.H.Z.; formal analysis, J.A.; investigation, J.A.; resources, S.R.H., R.A.; data curation, J.A.; writing—original draft preparation, J.A., C.H.; writing—review and editing, J.A., C.H., R.A., S.R.H. and M.H.Z.; visualization, J.A.; supervision, C.H., M.H.Z.; project administration, M.H.Z.; S.R.H. and R.A.; funding acquisition, R.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the internal research start-up funding, reference: 1012606FA1, from the University of East Anglia (UEA), Norwich, UK.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ogbodo, E.U.; Abu-Mahfouz, A.M.; Kurien, A.M. A Survey on 5G and LPWAN-IoT for Improved Smart Cities and Remote Area Applications: From the Aspect of Architecture and Security. *Sensors* **2022**, *22*, 6313. [CrossRef]
2. Liu, A.-C.; Tu, P.-T.; Langpoklakpam, C.; Huang, Y.-W.; Chang, Y.-T.; Tzou, A.-J.; Hsu, L.-H.; Lin, C.-H.; Kuo, H.-C.; Chang, E.Y. The Evolution of Manufacturing Technology for GaN Electronic Devices. *Micromachines* **2021**, *12*, 737. [CrossRef]
3. Bzai, J.; Alam, F.; Dhafer, A.; Bojović, M.; Altowaijri, S.M.; Niazi, I.K.; Mehmood, R. Machine Learning-Enabled Internet of Things (IoT): Data, Applications, and Industry Perspective. *Electronics* **2022**, *11*, 2676. [CrossRef]
4. Rodrigo, F.M.; Alberto Luiz, A. Challenges for development and technological advancement: An analysis of Latin America. *Inf. Dev.* **2018**, *35*, 413–420.
5. Bellini, P.; Nesi, P.; Pantaleo, G. IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies. *Appl. Sci.* **2022**, *12*, 1607. [CrossRef]
6. Sheng, Z.; Yang, S.; Yu, Y.; Vasilakos, A.V.; Mccann, J.A.; Leung, K.K. A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wirel. Commun.* **2013**, *20*, 91–98. [CrossRef]

7. Bahaddad, A.A.; Almarhabi, K.A.; Alghamdi, A.M. Factors Affecting Information Security and the Implementation of Bring Your Own Device (BYOD) Programmes in the Kingdom of Saudi Arabia (KSA). *Appl. Sci.* **2022**, *12*, 12707. [CrossRef]
8. Downer, K.; Bhattacharya, M. BYOD Security: A Study of Human Dimensions. *Informatics* **2022**, *9*, 16. [CrossRef]
9. Pallavi, S.; Smruti, R.S. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 9324035.
10. Basarir-Ozel, B.; Turker, H.B.; Nasir, V.A. Identifying the Key Drivers and Barriers of Smart Home Adoption: A Thematic Analysis from the Business Perspective. *Sustainability* **2022**, *14*, 9053. [CrossRef]
11. Martin, S.; Martin, H.; Sacha, H.; Marko, S.; Klaus, W. Towards In-Network Security for Smart Homes. In Proceedings of the ARES 2018: International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018; pp. 1–8.
12. Al-Turjman, F.M. Towards smart e-health in the ultra large-scale Internet of Things era. In Proceedings of the 23rd Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME), Tehran, Iran, 24–25 November 2016; pp. 102–105.
13. Rodrigo, R.; Javier, L.; Masahiro, M. A survey and analysis of security threats and challenges. *Future Gener. Comput. Syst.* **2018**, *78*, 680–698.
14. Williams, L.; Wood, P.; Marsden, J. Protocol for Interaction between Wireless Devices and Other Devices. U.S. Patent 9,891,867, 13 February 2018.
15. Bhagya, N.S.; Murad, K.; Kijun, H. Towards sustainable Smart Cities: A review of trends, architectures, components, and open challenges in Smart Cities. *Sustain. Cities Soc.* **2018**, *38*, 697–713.
16. Granjal, J.; Monteiro, E.; Sá Silva, J. Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1294–1312. [CrossRef]
17. Karagiannis, V.; Chatzimisios, P.; Vázquez-Gallego, F.; Alonso-Zarate, J. A survey on application layer protocols for the Internet of Things. *Trans. IoT Cloud Comput.* **2015**, *3*, 11–17.
18. Joseph, B.; Andreas, J.; Paul, D. An Empirical Analysis of Smart Connected Home Data. In *Internet of Things—ICIOT 2018, Proceedings of the Third International Conference, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, 25–30 June 2018*; Springer: Berlin/Heidelberg, Germany, 2018.
19. Taivalsaari, A.; Mikkonen, T. A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Softw.* **2017**, *34*, 72–80. [CrossRef]
20. Angelakis, V.; Tragos, E.; Pöhls, H.; Kapovits, A.; Bassi, A. *Designing, Developing, and Facilitating Smart Cities*; Springer: Cham, Switzerland, 2017.
21. Al-Sarawi, S.; Anbar, M.; Alieyan, K.; Alzubaidi, M. Internet of Things (IoT) communication protocols: Review. In Proceedings of the 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017; pp. 685–690.
22. Roca, D.; Milito, R.; Nemirovsky, M.; Valero, M. Tackling IoT ultra large scale systems: Fog computing in support of hierarchical emergent behaviors. In *Fog Computing in the Internet of Things: Intelligence at the Edge*; Springer: Berlin, Germany, 2018; pp. 33–48.
23. Kim, H.S.; Ko, J.; Culler, D.E.; Paek, J. Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2502–2525. [CrossRef]
24. Byers, C.C. Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks. *IEEE Commun. Mag.* **2017**, *55*, 14–20. [CrossRef]
25. Saha, H.N.; Mandal, A.; Sinha, A. Recent trends in the Internet of Things. In Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–4.
26. Silva, B.N.; Khan, M.; Han, K. Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Tech. Rev.* **2018**, *35*, 205–2220. [CrossRef]
27. Shen, B.; Naveen, C.; Ru, W.; Xingshe, Z.; Shiwei, W.; Wen, J. Deadline-aware rate allocation for IoT services in data center network. *J. Parallel Distrib. Comput.* **2018**, *118*, 296–306. [CrossRef]
28. IoT Software Development Guide. Available online: http://postscapes2.webhook.org/internetof-things-software-guide/ (accessed on 10 July 2021).
29. Amin, F.; Abbasi, R.; Khan, S.; Abid, M.A. An Overview of Medium Access Control and Radio Duty Cycling Protocols for Internet of Things. *Electronics* **2022**, *11*, 3873. [CrossRef]
30. Tripathi, J.; de Oliveira, J.C.; Vasseur, J.P. A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks. In Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 17–19 March 2010; pp. 1–6.
31. Khan, A.A.; Rehmani, M.H.; Reisslein, M. Requirements, Design Challenges, and Review of Routing and MAC Protocols for CR-Based Smart Grid Systems. *IEEE Commun. Mag.* **2017**, *55*, 206–215. [CrossRef]
32. Melike, Y.; Cagri, G.V.; Etimad, F.; Laila, N.; Nadine, A.; Ian, F. Akyildiz, "Channel-aware routing and priority-aware multi-channel scheduling for WSN-based smart grid applications. *J. Netw. Comput. Appl.* **2016**, *71*, 50–58.
33. Angelo, C.; Gianluca, D.C.; Chiara, P. R-CARP: A Reputation Based Channel Aware Routing Protocol for Underwater Acoustic Sensor Networks. In Proceedings of the 10th International Conference on Underwater Networks & Systems (WUWNET '15), ACM, New York, NY, USA, 22–24 October 2015; Volume 37, Issue 1.
34. Diego, M.M.; Ioannis, P.; Baijian, Y. Internet of Things: Survey on Security and Privacy. *ACM Inf. Secur. J. A Glob. Perspect.* **2017**, 1–16.
35. Xiaonan, W.; Zhengxiong, D.; Dong, W.; Sun, Q. Mobility management for 6LoWPAN WSN. *J. Comput. Netw.* **2018**, *131*, 110–128.

36. Lynn, K.; Martocci, J.; Neilson, C.; Donaldson, S. Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks. RFC 8163. *rfc-editor.org*; IESG: Hong Kong, China, 2017.
37. Choi, Y.H.; Hong, Y.G.; Shin, M.K.; Kim, H.J. Method for Transmission of IPv6 Packets over Near Field Communication (NFC) and Device Operating the Same. U.S. Patent 9,877,144, 23 January 2018.
38. Sang-Hyun, P.; Seungryong, C.; Lee, J.-R. Energy-Efficient Probabilistic Routing Algorithm for Internet of Things. *J. Appl. Math.* **2014**, *2014*, 213106.
39. Darroudi, S.; Gomez, C. Bluetooth low energy mesh networks: A survey. *Sensors* **2017**, *17*, 1467. [CrossRef]
40. Mariager, P.; Petersen, J.; Shelby, Z.; Van de Logt, M. Transmission of IPv6 Packets over Digital Enhanced Cordless Telecommunications (DECT) Ultra Low Energy (ULE). RFC 8105. *rfc-editor.org*; IESG: Hong Kong, China, 2017.
41. Gomez, C.; Paradells, J.; Bormann, C.; Crowcroft, J. From 6LoWPAN to 6Lo: Expanding the Universe of IPv6-Supported Technologies for the Internet of Things. *IEEE Commun. Mag.* **2017**, *55*, 148–155. [CrossRef]
42. Dujovne, D.; Watteyne, T.; Vilajosana, X.; Thubert, P. 6TiSCH: Deterministic IP-enabled industrial internet (of things). *IEEE Commun. Mag.* **2014**, *52*, 36–41. [CrossRef]
43. Callaway, E.; Gorday, P.; Hester, L.; Gutierrez, J.A.; Naeve, M.; Heile, B.; Bahl, V. Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks. *IEEE Commun. Mag.* **2002**, *40*, 70–77. [CrossRef]
44. Hunkeler, U.; Truong, H.L.; Stanford-Clark, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks, Communication Systems Software and Middleware and Workshops. In Proceedings of the 3rd International Conference on COMSWARE, Bangalore, India, 6–10 January 2008; pp. 791–798.
45. Yassein, M.B.; Shatnawi, M.Q.; Al-Zoubi, D. Application layer protocols for the Internet of Things: A survey. In Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco, 22–24 September 2016; pp. 1–4.
46. Peter, W.; Ronny, K. Internet of Things–Discovery. XEP-0347. XMPP Standards Foundation: Victor, NY, USA, 2017.
47. Tara, S. *Internet of Things Protocols and Standards, Networking Protocols and Standards for Internet of Things*; Washington University: St. Louis, MI, USA, 2015.
48. Thangavel, D.; Ma, X.; Valera, A.; Tan, H.X.; Tan, C.K.Y. Performance evaluation of MQTT and CoAP via a common middleware. In Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.
49. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120. *rfc-editor.org*; IESG: Hong Kong, China, 2011.
50. IoT Daily. Available online: https://iot-daily.com/2015/03/19/4-main-must-haves-for-the-physical-layer-of-internet-of-things-wireless-connectivity/ (accessed on 5 February 2018).
51. Rajandekar, A.; Sikdar, B. A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications. *IEEE Internet Things J.* **2015**, *2*, 175–186. [CrossRef]
52. Luzuriaga, J.E.; Perez, M.; Boronat, P.; Cano, J.C.; Calafate, C.; Manzoni, P. A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks. In Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2015; pp. 931–936.
53. Pajooh, H.H.; Demidenko, S.; Aslam, S.; Harris, M. Blockchain and 6G-Enabled IoT. *Inventions* **2022**, *7*, 109. [CrossRef]