JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AN EMPIRICAL STUDY ON SYSTEM-LEVEL ASPECTS OF THE INTERNET OFTHINGS (IOT)

MADDINENI VYSHNAVI[1], KOYINANA.MEGHANA[2], MADAKA MEGHANA[3]
Student[1], Department of MCA, Andhra Loyola College, Vijayawada
Student[2], Department of MCA, Andhra Loyola College, Vijayawada
Student[3], Department of MCA, Andhra Loyola College, Vijayawada

ABSTRACT:
The Internet of Things (IoT) is a combination of sensor, embedded, computing, and communication technologies. The goal of the IoT is to give seamless services to anything, at any time, and in any location.IoT devices are everywhere, ushering in the fourth revolution of disruptive technologies following the internet and Information and Communication Technology (ICT). According to the research and development community, the impact of IoT on society will be greater than that of the internet and ICT, which promotes the well-being of society and industries. Addressing the prevalent system-level design characteristics such as energy efficiency, robustness, scalability, interoperability, and security challenges results in the utilization of a prospective IoT system. This article summarizes the current state of the artfor the IoT functional pillars.

Its growing applications aim to encourage academics and researchers to create real-time, energy-efficient, scalable, dependable, and secure IoT systems. This document outlines IoT architecture, as well as the current state of IoT architectures. Highlights of IoT system-level challenges have been highlighted inorder to design more powerful real-time IoT applications. Millions of devices exchange data using many communication standards, and interoperability is a serious challenge. This study gives a detailed review ofthe current state of communication standards and application layer protocols utilized in IoT. Cloud, Cloudlet, Fog, and Edge computing paradigms enable IoT by providing services such as data offloading, resource and device management, and so on. An in-depth examination of Edge Computing in IoT using several edge computing designs is presented in this study.

INDEX TERMS IoT, pillars of IoT, emerging IoT applications, IoT application requirements, IoT architecture, IoT application layer protocols, computing paradigms (edge, fog, cloudlets & cloud), privacy& security, and platforms for IoT.

NOMENCLATURE

AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
APIs	Application Interfaces
BAN	Body Area Networks
BLE	Bluetooth Low Energy
BS	Base Station
CART	Classification and Regression Tree
CHAID	Chi-squared Automatic Interaction Detector
CRP	Challenge-Response Pair
CoAP	Constrained Application Layer Protocol
DDS	Data Distribution Services
DTLS	Datagram Transport Layer Security
GIS	Geographical Information System
HTTP	Hypertext Transfer Protocol
IDPs	Intrusion Detection and PreventionSystem
IEC	International ElectrotechnicalCommission
IoT	Internet of Things
ISO	International Standard Organization
ITU	International TelecommunicationUnion

ITS	Intelligent Transportation System
KNN	K-Nearest Neighbor
LTE	Long-Term Evolution
MAC	Medium Access Control
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NS3	Network Simulator-3
OS	Operating System
PDR	Packet Delivery Ratio
QoS	Quality of Services
QUEST	Quick, Unbiased, Efficient, StatisticalTree
RSSI	Received Signal Strength Indicator
SVM	Support Vector Machine
TLS	Transport Layer Security
TIS	Traveler Information System
WiFi	Wireless Fidelity
WWW	World Wide Web
XMPP	Extensible Messaging and PresenceProtocol

I. INTRODUCTION

The Internet plays a significant role in information transmis-sion [1]. However, the technology is moving towards data col- lection, analysis of the data, and controlling devices remotelyvia the Internet rather than just sharing the information, and this results in a new technology called IoT. IoT is an intercon-nection of various physical devices to collect, control, ana- lyze, and share data in real-time [2]–[4]. IoT aims to enhancethe quality of life [5]. The motivating factor for the extensivegrowth of IoT technology is that most of the manufactur- ing industries, service providers, and software industries are investing more and adopting IoT technologies more swiftly.Forbes estimated that the transformation of the hypothetical concepts of iot to reality started during the year 2015 [6]. As per the survey, the number of internet-connected objects are likely to be 75.44 billion [7] and the economic growth of the IoT technology will range from $2.7 TO $6.2 trillion by 2025 [8]; this shows the impact of IoT technology on society. Figure 1 illustrates the number of devices predicted to be connected to the internet by 2025, and these devices generateapproximately 80 Zettabytes of data [9]. Figure 2 representsthe research progress in the area of IoT [10]. IEEE Digital Library (IEEE Xplore) is considered for the search and selec-tion process. IoT or Internet of Things is used as a search keyword. The number of conferences and journal papers areretrieved by limiting the year field of the IEEE Xplore.

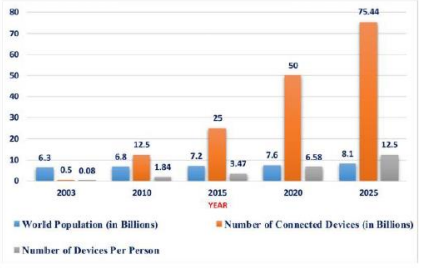


FIGURE 1. Number of devices predicted to be connected to the internet by 2025.

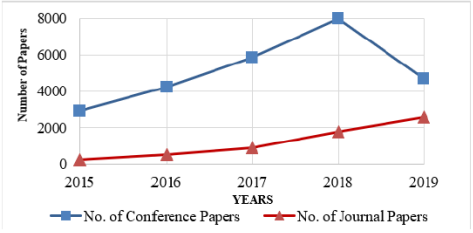


FIGURE 2. IoT research trend from 2015 (Source: IEEE digital library).

IoT has an impact on most of the applications such as healthcare, intelligent homes, smart farming, factory automa-tion and industry 4.0, intelligent transport systems, smart cities, infrastructure monitoring, retail industry, environmen-tal monitoring, smart water, and power grids, etc., [11]. Figure 3 shows the diversity of IoT applications.

IoT technology has traversed a long way (almost a decade) and offers numerous opportunities. Many IoT components and devices with multiple communication standards, messag-ing protocols, computation technologies, and security algo- rithms are under development. For example, different vendors such as ARM, Atmel, Silicon Labs, Texas Instruments, Intel, NVIDIA, Samsung, etc. are the primary producers of IoT components/ chips and development tools with their stan- dards. BLE, ZigBee, WiFi, Z-wave, Sigfox, NB-IoT, 5G, etc. are the communications standards used by IoT devices for short and long-range communication. IoT applications use application/ messaging protocols like MQTT, CoAP, DDS, XMPP, RESTful APIs/ HTTP to send and receive messages. Also, IoT applications utilize computing technologies like Edge, Fog, Cloudlets, and Cloud TO store, process, and ana- lyze the data. In IoT, data plays a significant role in making decisions, which is highly private and sensitive.

Figure 4 illustrates the hybrid architecture of the IoT ecosystem. From Figure 4, it is being noticed that

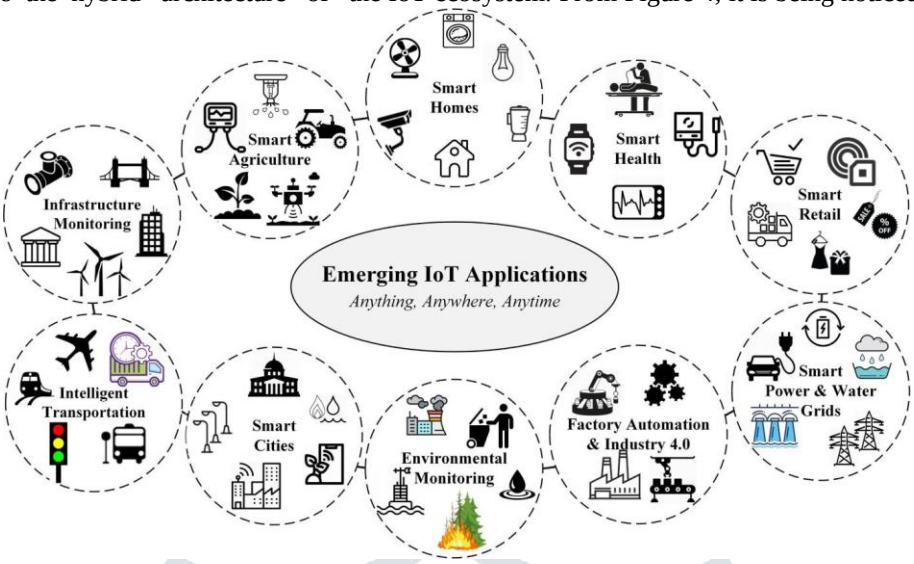


FIGURE 3. Applications of IoT.

computation and communication technologies are the pri- mary requirements of an IoT system. It also shows how the above technologies have enabled the development and deployment of IoT systems. Figure 5 illustrates the detailed deployment scenario of IoT. IoT technology comprises of End-node, Edge, Fog, Cloudlets, and Cloud computing tech-nologies, which bring the intelligence to IoT. In general, sensors and signal-conditioning circuits are connected to the end-nodes; these nodes are used when the application needs to deploy a massive number of high-density sensors tocapture the physical parameters from the environment. Gen- erally, a group of end-nodes forms a cluster. The edge-nodes acts as a cluster head, which collects the raw data fromthe end-nodes and preprocesses the data. Also, edge-nodes are accountable for reliable data transmission using short and long-range communication standards. Introducing the edge-nodes at the edge of the network provides the different deployment scenarios based on the services required by the application. Scenario 1: In this, the edge-node forwards the data to the gateway (acts as fog node) for further processing,Scenario 2: In this, mobile edge nodes access to the servicesoffered by the cloudlets, and Scenario 3: The edge nodes directly access the cloud services. Further, Gateway or BS acts as a fog node, which does the fog computing. Data processing and information extraction are the more common operations performed at the fog level. This information is also used to provide essential services to the underlying devices (Edge and end nodes), including the estimation of optimized communication costs. Also, it establishes the communication either to cloudlets and cloud infrastructure to store and deliver the more service-oriented value(s) ata higher level. Generally, cloudlet services are used when IoT devices are highly mobile. For example: Streaming the High Definition (HD) video/ movie in an autonomous car. To share data effectively, IoT devices (end nodes, edge nodes, fog node, and cloudlets) use either short-range (RFID, BLE, Zigbee, Z-Wave, Thread) or long-range (LoRaWAN, NB- IoT, Sigfox, 5G, Telensa, Ingenu) communication standards. In IoT, both computation and communication resources need to be utilized efficiently and carefully since IoT devices are resource constraint devices and pose many challenges. The system-level aspects like energy-efficiency, robustness, het-erogeneity and interoperability, and other aspects like data & device management and QoS parameters need to be consid- ered while realizing the potential and mature IoT system. Additionally, privacy and security issues should be handled by the IoT system design phase.However, some of these issues are solved in traditional internet technologies. How- ever, it is impractical to apply those solutions/ mechanisms directly to IoT systems; this is due to the useof multiple sensors, communication standards, protocols, and computing technologies along with the demand-driven services in a constrained environment. IoT devices require novel light- weight algorithms with fewer memory requirements and less computation complexity. This paper deliberates the system-level design issues in the development of an interoperable, secure, scalable, reliable and energy-efficient IoT system.

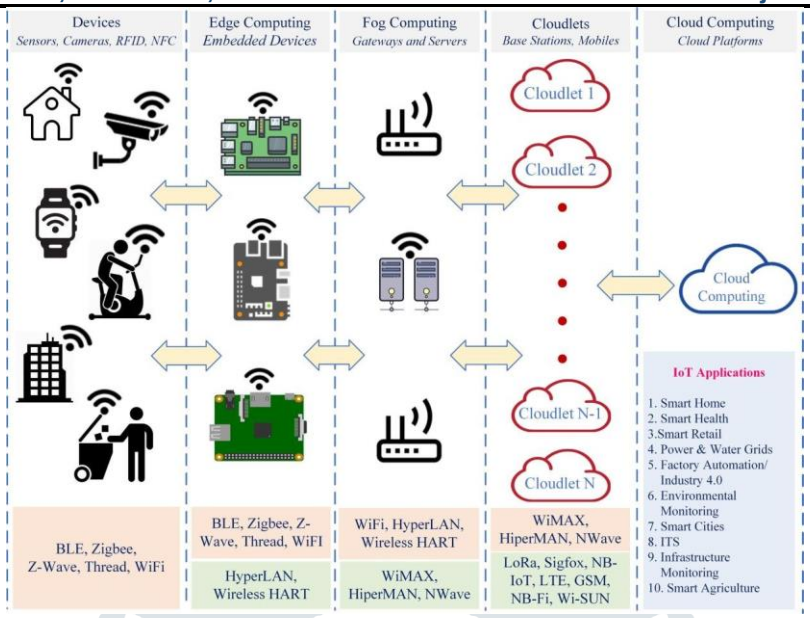


FIGURE 4. Hybrid architecture of IoT.

A. COMPARISON OF RELATED SURVEY PAPERS

Several survey papers have been published which cover the different characteristics of IoT technology. Dhanvijay and Patil [12] discuss IoT healthcare networks, architectures, WBAN technology, security issues, challenges, and open issues in the IoT healthcare system. In [13], the authors have presented the existing short and long-range commu- nication standards. The authors also provided their views on the emerging communication technologies like Com- pressive Sensing, Non-Orthogonal Multiple Access, Massive Multiple-Input Multiple-Output (mMIMO), and ML-based random-access protocols. In [14], the authors address the IoT enabling technologies, recent advancements in IoT commu- nication standards (BLE), and application protocols (MQTT, CoAP, HTTP). The authors also presented various IoT appli- cations and security issues. Nauman *et al.* [15] provided a comprehensive study on Multimedia IoT (M-IoT) with a focus on architectures (Agent, SDN, Fog, and AI-based architecture), protocols (Routing and PHY-MAC protocols), and different M-IoT applications. Also, the authors have discussed the importance of the Quality-of Experience (QoE) and QoS features in M-IoT. In [16], authors have investigated the trends in IoT access control and performed a detailed analysis of existing authorization frameworks in IoT. The authors have observed that there is no generic access control mechanism/ system for IoT applications. In [17], the authors have focused on fundamentals of trust modelling, the impor- tance of trust information fusion, possible trust threats affect- ing the IoT applications, and IoT architecture for modelling the trust. The authors in [18] provided the advantages of fog computing over cloud computing (Characteristics), fog architecture, hardware, and software platforms for fog com- puting, and its use in various IoT applications, and future research directions. In [19], the authors have deliberated the IoT architecture, security issues, and requirements. The authors also provided the taxonomy of the IoT authentica- tion protocols, detailed analysis of them, and problems that need to be considered while developing novel authentication schemes. Da Costa *et al.* [20] provided a detailed analysis of machine learning algorithms applied to spot the intruders in IoT networks. The authors also provided information related to the different datasets available to conduct research in the field of security. In [21], the authors carried out an exhaustive survey on the state of the art of IoT security. The authors used applications like smart home, healthcare, and smart grids to review privacy, forensics, and other security challenges.

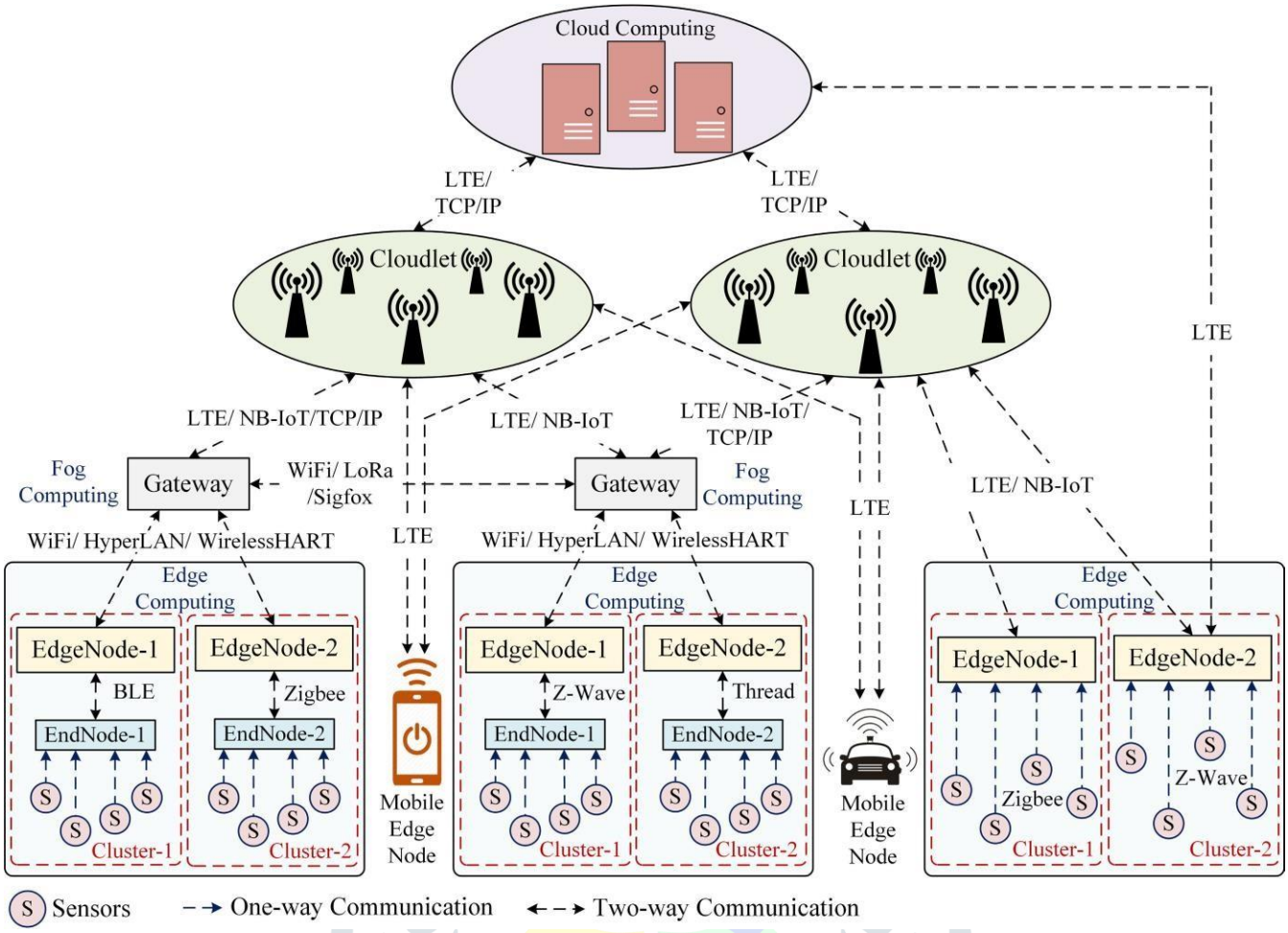


FIGURE 5. IoT deployment scenario.

The authors in [22] performed a detailed analysis of trust management techniques (E-Lithe, GTRS, TWGA, TrustCEP, TTBBS, etc.) for IoT. Based on the review, the authors have classified the various techniques with the pros and cons. In [23], the authors provided the taxonomy of the machine learning techniques. Also, the authors have provided a detailed analysis of statistical, data mining, and machine learning techniques used in the different IoT applications, and a comparison of various ML algorithms. The authors also reviewed the different computing architectures used by IoT applications. Marietta and Chandra Mohan [24] performed a detailed analysis of the WSN and ad hoc routing protocols by considering the characteristics like bandwidth, topology, scalability, and mobility of the nodes. The investigation of these routing protocols is based on the least transmission time and shortest path. Finally, the authors have provided the characteristics of the IoT routing protocols. In [25], the authors presented the taxonomy of the security require- ments and analyzed IoT security attacks. Also, the authors have analyzed few security solutions such as Intelligent Security Framework, Chaos-based Privacy Preservation, etc. In [26], the authors offered the review of currently available testbeds like MoteLab, ORBIT, Trio, Indriya, Flocklab, Sen-sorscope, TWIST, etc. for WSN and IoT applications. They have performed a detailed analysis of testbeds based on the characteristics. Lu and Da Xu [27] discussed the cybersecu-rity issues in detail, along with the cybersecurity attacks tax-onomy and middleware layered cybersecurity architecture. The authors also highlighted the different security schemes like Host identity, Datagram Transport layer, and Capability-based access control scheme. In [28], the authors have dis- cussed the architecture, protocols (Physical, Transport, and Application Layer protocols, WSN and IoT routing proto- cols), scalability, and privacy and security concepts in IoT. Table 1 depicts the summary of the survey papers in the IoT area.

B. MOTIVATION

IoT applications are gradually evolving day by day. How- ever, the development of a scalable, reliable, energy-efficient architectures and secure IoT system remains challenging in practice. In literature, the authors have discussed the various aspects of IoT. However, no works have presented a detailed analysis of the IoT system design issues. Therefore, increased recognition and more attention need to be imparted to this area. Hence, this survey provides a comprehensive study of system-level aspects and enabling technologies of IoT with a focus on interoperability, application layer protocols and security.

TABLE 1. Summary of the survey papers.

Ref. & Year	Applications	IoT Challenges	System Design Aspects			Computing Paradigms	Protocols	Security & Privacy
			Architecture	Communication	OS			
[14] 2020	Smart Home, Smart Retail, Smart Agriculture, Industry, Environment, Smart Health, Smart City	×	×	BLE, Zigbee, WiFi, LTE, 5G, Sigfox, NB-IoT	×	×	MAC Layer (CSMA, CSMA/CA, TDMA, CDMA)	×
[15] 2019	Healthcare, Industrial IoT, Home Automation, Smart Cities, Emergency Care, Smart Agriculture	×	×	×	×	×	Application Layer (MQTT, CoAP)	✓
[16] 2020	Road Management, Surveillance, Industrial application, Healthcare	×	Three, Middleware, SDN & Cloud based Layered Architecture	Zigbee, WLAN, NB-IoT, BLE, LoRaWAN	×	Cloud, Fog & SDN Computing	Routing and PHY-MAC Protocols	×
[17] 2019	Smart Homes, Smart Health, Smart Building, Connected Vehicles, Manufacturing	×	Middleware Architecture	Zigbee, WiFi, BLE, LoRaWAN, Z-Wave	×	×	HTTP, CoAP, MQTT, Restful, XMPP	Access Control
[13] 2019	Healthcare	Scalability, Privacy & Security	Middleware Architecture	×	×	×	×	✓
[19] 2019	Intelligent Transportation System, Public Safety, Smart Grids, Industry 4.0, Smart Homes	Security	Middleware Architecture	×	×	Fog Computing	×	×
[20] 2019	Smart Grids, Vehicular Networks, Smart Homes, Mobile Applications,	×	Three, Five and Middleware Architecture	×	×	×	×	Authentication
[23] 2020	Automotive, Environmental Monitoring, Agriculture, Healthcare, Industrial, Retail, Banking, Supply Chain, Smart Homes, Smart City	×	×	×	×	Cloud, Fog & Edge Computing	×	×
[26] 2019	Smart Home, Healthcare, Transportation	×	Middleware Architecture	Zigbee	×	×	6LowPAN, RPL, CoAP	✓
[27] 2020	Smart Homes, Smart Health, Smart Transportation, Retail, Industry 4.0, Smart Agriculture	Security	Middleware Architecture	RFID, BLE, Zigbee	×	×	Application Protocols (MQTT, CoAP, HTTP)	×
This Survey	Smart Homes, Smart Health, Smart Farming, Intelligent Transportation, Factory Automation/ Industry 4.0, Smart Retail, Smart Cities, Environmental Monitoring, Infrastructure Monitoring, Smart Power and Water Grids	**System Level Challenges** Architectures, Reliability, Interoperability, Self-Diagnosis & Adaptability **Other Challenges** Availability, Mobility, Scalability, Device and Data Management, Privacy & Security	Three, Five and Middleware Architecture	Communication Models and Standards RFID, Zigbee, Z-wave, WiFi, NB-IoT, BLE, LoRaWAN, Sigfox, 5G	Real-Time Operating Systems and IoT Protocol Stack	Cloud, Cloudlets, Fog and Edge Computing & its Architecture	Application Layer (MQTT, CoAP, XMPP, DDS, Restful APIs)	General Attacks, Mitigation Techniques

C. CONTRIBUTIONS TO THIS SURVEY

The rapid growth of smart devices and the use of various technologies in IoT poses countless research opportunities. In reality, there are no large-scale IoT applications till date. Industries, research organizations, and academicians need to address the current research issues and development of standards in IoT. This survey paper aims to provide an extensive understanding of the current research status of IoT system-level aspects. It also provides the research directions in emerging IoT application domains, architectures, communication standards and application protocols, computing paradigms, and RTOS. In summary, this work aims

- To provide deeper insights into the IoT system to the readers
- To describe the functional pillars of IoT
- To give an overview of emerging IoT applications, its classification, characteristics and requirements
- To review the IoT application requirements and challenges

The remaining part of the paper is organized as follows. Section II provides functional pillars of IoT. Section III delib-erates emerging and popular IoT applications. Section IV reviews the IoT system-level challenges and current research state. Section V analyses the different IoT architectures, the current state of the art, and issues that need to be consid-ered during architectural design. Section VI and VII provide a detailed analysis of communication standards and applica-tion layer protocols used in IoT, respectively. Section VIII provides the in-depth analysis of computing paradigms (Edge, Fog, Cloudlets, and Cloud computing architectures), different edge architectures, and areas where further research is required. Section IX provides a detailed analysis of privacy and security issues, mitigation techniques, and what needs to be addressed in the future. Section X highlights the plat-form support for IoT, and protocol stack supported by them. Section XI lists open research issues and future directions that need to be addressed. Finally, Section XII provides the concluding remarks.

II. PILLARS OF IoT

IoT increases the quality of life by providing numerous appli- cation services to the users. Sensing & Actuation, device identification, communication, computation, application ser-vices, management, and security are the major functional pillars of an IoT system. Figure 6 illustrates the functional pillars of IoT.

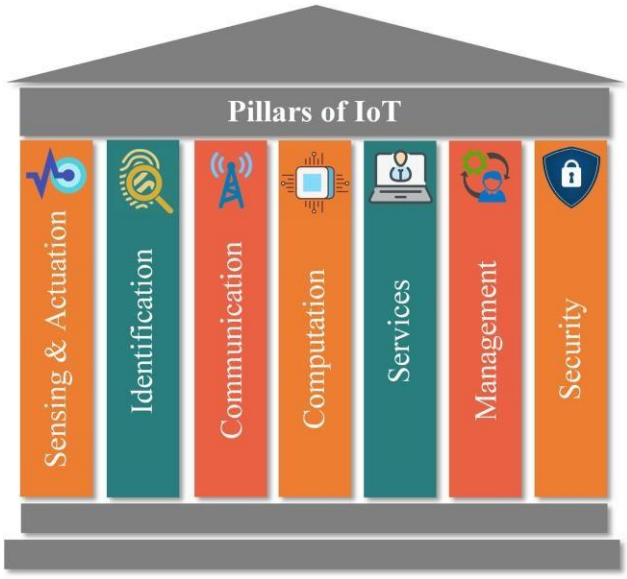


FIGURE 6. Pillars of IoT.

1) SENSING AND ACTUATION

In IoT, sensing is all about collecting various data from the environment using different sensors. The IoT sensors can be acoustic, chemical, biological, pressure, thermal, wearable, implantable sensors, actuators, RFID Tags, etc., [29], [30]. The data collected from these sensors are processed and analyzed to actuate and improve decision making in IoT. IoT is a collection of various types of sensors, which have their own requirements. To maintain the standardization among the IoT sensors, IEEE Electronic Engineering Association introduced the IEEE1451 smart transducer protocol stack for the development of smart sensors [31].

2) DEVICE IDENTIFICATION

Devices are the central pillar of IoT; these are also called as objects/ things, which are the essential sources of data [32]. Object identification is used to identify the entity of interest in an IoT application. Objects are used to perceive, operate, monitor, and control the IoT application services. In IoT, devices need to be identified uniquely to provide applica- tion services with an increase in security. Object Identifier (OID), Electronic Product Code (EPC), and Universal Unique Identifiers (UUID) are commonly used to identify the devices within the network [11]. Figure 7 depicts the various identi-fication standards used in IoT.

a. OID: It is an object identification mechanism collabora- tively developed by ITU-T and ISO/ IEC. This mechanism uses the hierarchical structure to assign unique identifica-tion numbers to the objects. Levels of the OID tree are referred to as an arc. The highest arc in the tree is the root, which does not contain any name, and comprises of three management organization such as ITU-T, ISO, and ISO-ITU-T (Jointly managed by ISO and ITU-T). These management organizations are uniquely identified using the values 0, 1, and 2, respectively. The second arc repre-sents the category of the data, which belongs to (For exam- ple, administration, recommendation, questions, etc.), and the third arc represents the country code. The one M2M system prefixes the OID with the device identity [33]. An OID mechanism was introduced to eliminate the inter-operability at the device identification level [34].
b. EPC: It is a unique identifier used to identify the devices around the globe. EPC is used to track the object infor- mation within the EPCglobal network. EPC is 64-bit(I, II, and III) and 96-bits long. Typically, a 96-bit EPC comprises of a header (8-bit)- used to identify the EPC ver-sion number, EPC manager (28-bit)- used to identify the manufacturer of the item, object class (24-bit)- identifies the type of product made by the manufacturer and serial number (36-bit)-identifier for the individual item. The first two fields are assigned by the EPCglobal, and the remain-ing fields are assigned by the EPC manager [11], [34].
c. UUID or Globally Unique Identifier (GUID): It is a 128-bit number used to identify an object uniquely. The probability of reproducing the same UUID is almost zero. The general format of the UUID comprise of five groups, such as 8-4-4-4-12 bits. The sample UUID is as fol- lows 67080a16-331b-4b85-b7d5-db3121568d14. UUIDs are widely used in databases like MySQL, MariaDB and etc. as database keys [35]. Also, UUID are used to iden- tify the services provided by the BLE devices in IoT network.

For example, in V2V communications, connected vehi- cles are uniquely identified using ISO 3779 standard. ISO 11784 is used to identify the animals uniquely in smart farm-ing and animal husbandry. Similarly, ISO/ IEC 15459 is used to identify the products in the supply chain [36].

3) COMMUNICATION

The aim of this element is to share the captured data using different communication standards across the globe, and a detailed description of this is presented in Section VI.

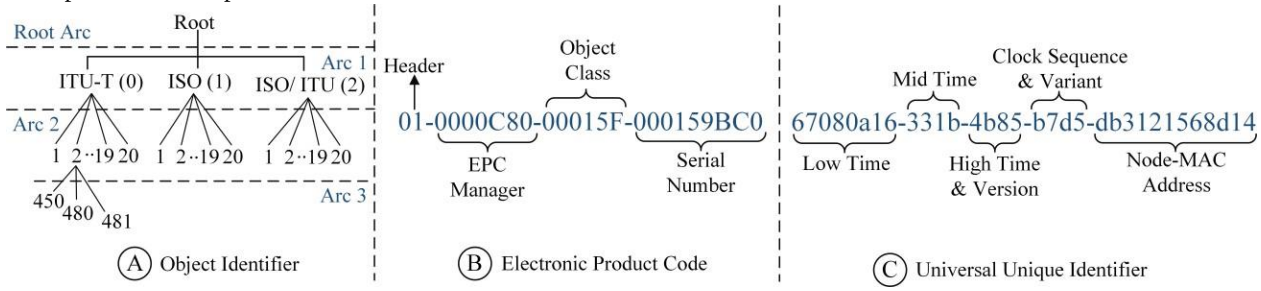


FIGURE 7. Identification standards in IoT.

4) COMPUTATION

Processing units such as microprocessors, microcontrollers, SoCs, and FPGAs and software packages act as a brain of IoT. The operating systems like TinyOS, RIOT, Ubuntu Core, Raspbian, Contiki OS, etc. provides the networking and IoT application development environment [37]. Cloud, cloudlets, fog, and edge platforms facilitate the data storage, analysis, and ubiquity in IoT [38].

5) SERVICES

Device Monitoring, Device Controlling, Data publishing, and Device Discovery are the services offered by an IoT system.

6) MANAGEMENT

This aims to manage the IoT users, roles of the users, and access the IoT services. The managing of users includes adding and removing the user from the group. It also controlsand records user activities.

a. Device Management: Various operations like - authen- ticating, configuring, monitoring, and maintaining the objects in the network are described in this block [39], [40].
b. Service Management: IoT provides the following services [41]
   i. Information Aggregation Service: The primary concern of this service is to sense, collect, and store the data from multiple environments and process the same to extractinformation through IoT infrastructure.
   ii. Collaborative Aware Services: Based on the aggregated data received, this service forms the better decision and makes reactions. The effectiveness of this service is based on the reliability of the infrastructure and com- putation cost.
   iii. Ubiquitous Services: This provides the Collaborative Aware Services at any-time and anywhere to anyone.

7) SECURITY

This element intends to secure the Devices, Communica- tion, and Services of an IoT system from modification and unauthorized access. A detailed analysis of this element is presented in Section VIII.

Security in Devices: IoT device security is the processof protecting the objects connected over a network from the attacker [42]. The typical lifecycle of IoT objects consists of Booting (Loading of Firmware), Initialization (Connection Establishment and Data Collection), Oper- ation (Desired Functionality of the Object), and Update (Installation of New Firmware and Rebooting). The entirelifecycle of IoT objects should be protected by employingsecurity algorithms.

a. Security in Communication: This provides end to end protection of the communication channels between the objects. Most of the objects in the IoT system uses wire- less communication technologies, and these channels areeasily prone to various types of attacks [43].
b. Security in Services: Securing the application services andits data from unauthorized access and modification by employing lightweight security algorithms.

III. EMERGING IoT APPLICATIONS

The applications like Smart Homes, Smart Health, Smart Farming, Smart Shopping, Intelligent Transportation, Factory automation & Industry 4.0, Environmental & Infrastructure Monitoring, Smart Cities, Smart Water & Power Grids etc. use IoT technology to increase the quality of life of human beings [44]. Figure 8 depicts the broad domains and sub- domains of emerging IoT applications.

A. SMART HOMES

Smart homes aim is to augment the resident's quality oflife [45]. The smart home is an integration of various domainslike - home automation, air quality monitoring, health care, surveillance, and smart gardening.

- **Home Automation:** Home automation enables the res-idents to control home appliances like Air Condition- ers, Fans, Washing machines, Refrigerators, Toasters, Coffee makers, Personal computers, Smartphones, etc. remotely using the internet [46].
- **Indoor Air Quality Monitoring:** It is essential to supervise the quality of the air inside the home oroffice, as people spent most of their time at home orin the office. Indoor air is more toxic than outdoor air. According to the survey, the use of stoves, tobacco, anddetergents increases the levels of CO2, NO, and NO2.

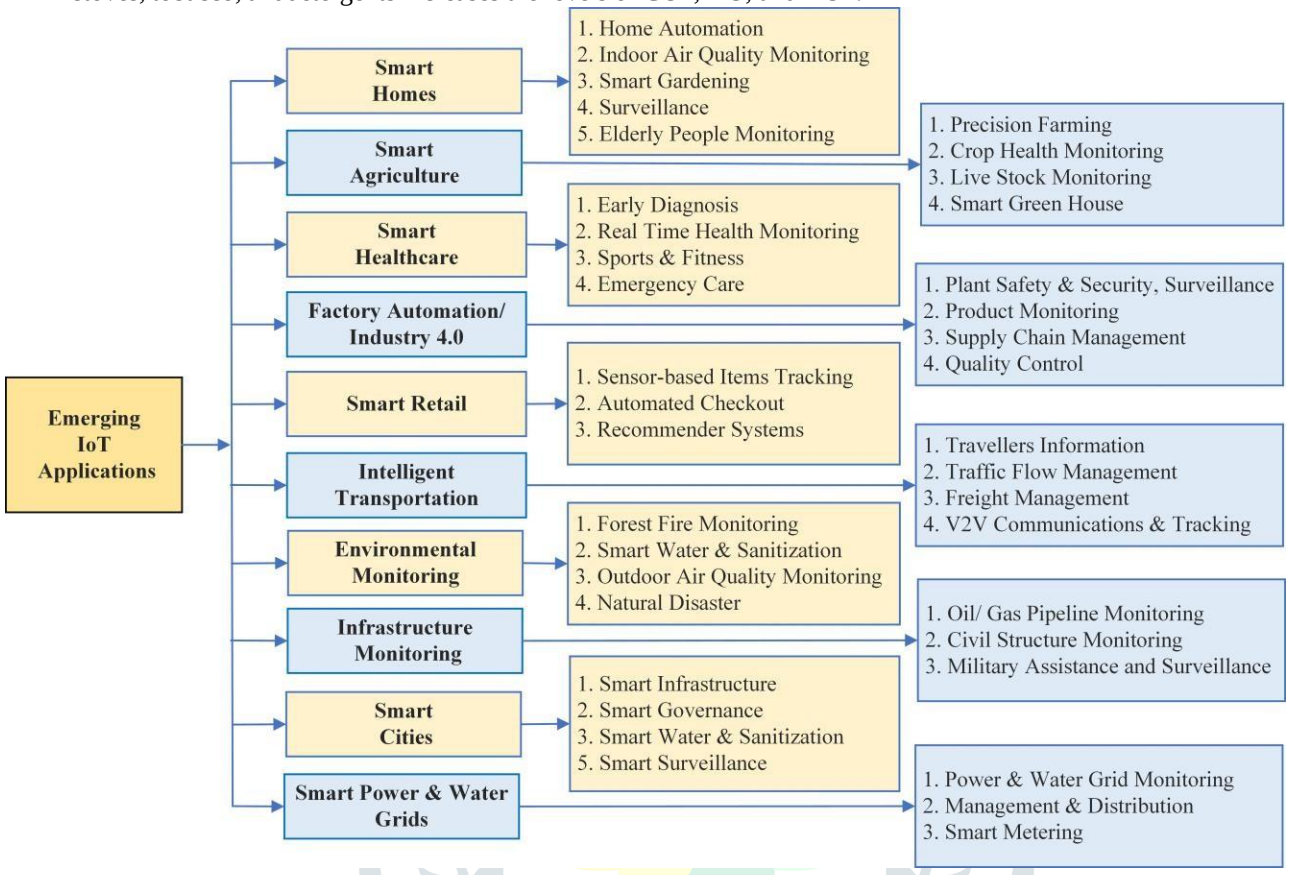


FIGURE 8. Broad application domains of IoT.

The researchers have noticed that children are suscep- tible to mental illness because of these pollutants [47]. IoT facilitates monitoring of the quality of the air and alerts the residents.

- **Smart Gardening:** In this busy life, most of the peopledo not have time to take care of the garden. Garden- ing improves the mental health of the residents andair quality in the surrounding habitation. The use ofIoT in smart gardening allows people to monitor the plants and provides nutrition and water from time to timeremotely [48].
- **Surveillance:** The surveillance using IoT aims to provide safety and security for the people in smart homes [49]. Integration of sensors, human-computer vision techniques, and IoT technology identify the anomalies and intruders, which provide real-time alertsto the residents [50].

- **Elderly People Monitoring:** The population of elderly people is expected to be increased by 16.7% in the year 2050 [51]. Elderly people suffer from chronic dis-eases and conditions like blood pressure, diabetes, heart diseases, and cancer. The advancements in biomedi- cal sensors and IoT allows the monitoring of elderly people health remotely; this increases the average life expectancy of elderly people [52].

- Real-time video analysis of elderly people needs to provide identification and classifications of multiple activities such as gait activities, fall detection, facial expression, sleep cycles, etc. Automated health assessment techniques also need to be improved. The smart home concept comprises of various sensors, actuators, and devices like Alexa, Fitbit, wireless speakers, Google assistance, Laptops, and so on; most of these devices are portable and wearable devices, which are used for long-term monitoring purpose. Hence, communi- cation and computation algorithms should be energy effi- cient. The devices used in smart homes are enormous in number and highly heterogeneous; maintaining the interop- erability between them is also a major concern and tracking these devices and handling data generated by them are still challenging. Surveillance cameras should provide real-time alerts to

the residents by recognizing and analyzing the multiple crime activities within a short period; this requires high-performance IoT systems to meet the real-time applica-tion requirements.

B. SMART HEALTH

Smart health aims to provide immediate medical service by reducing the cost of the medication to increase the quality of health in humans and pets [53]. Healthcare includes various sub-domains such as fitness programs, remote monitoring of health, chronic disease diagnosis, and monitoring of elderly people. Various sensors, medical, and imaging devices are used to acquire data from the human body.

- **Early Diagnostics:** Early diagnosis help the doctor to find the diseases and disorders in patients at earlier stages. Integration of sensor technologies (Wearable Sensors and Implantable Sensors) and BAN (Collects the real-time data like- sugar levels, BP, ECG, etc. from the patients) [54], and IoT facilitates the doc- tor to monitor and analyze patient's health conditions remotely. Early diagnosis plays a vital role in observ- ing Alzheimer's disease, Parkinsonism, malignancies and other health conditions [55]. The early diagnostics feature of IoT leads to better treatment and extended existence.
- **Real-Time Health Monitoring (RTHM):** The clinic-centric medical services have been changed to patient-centric medical services; this is due to the advancements in health monitoring devices and IoT. Real-time health monitoring plays a crucial role in monitoring the health of the sick and geriatric people on a timely basis remotely. The RTHM can be adopted by patients who are suffering from various diseases like cardiac and respiratory diseases. Also, patients with conditions like bradycardia, tachycardia, arrythmia, etc.
- **Sports and Fitness:** Most of the people in this digital era are working professionals and engaged in different activ-ities, due to which they have a habit of ignoring health and physical fitness [56]. Staying fit and maintaining the diet is very significant for patients and sportspersons. Smart devices like smartwatches, smartphones, smart gloves and shoes track the features like motion, calo- ries burnt, analysis of sleep cycles, heart rate, etc. IoT enables fitness trainers and coaches to send recommen-dations to the sportspersons and patients based on the data received which ultimately improves the fitness of patients and athletes.
- **Emergency Care:** Emergency medical care is a network of patients, hospitals, health workers and emer-gency vehicles. This provides services to the citizens during natural disasters like earthquake, building col- lapse, accidents, fire accidents, floods, war fields, and others. Emergency care also provides services like locat-ing nearby hospitals and emergency vehicles using the Global Positioning System (GPS) technology [57].

The critical requirements of smart health applications are data acquisition, real-time data processing, maintenance of the patient records, immediate intelligence, effective commu- nication, availability of the data, and privacy and security. Also, the use of Telehealth is increasing due to the inevitable situations like the COVID-19 pandemic, etc. which poses other requirements like seamless real-time video streaming. Interoperability among medical devices is also a signifi- cant concern; the researchers should emphasize on devel- oping faster, energy-efficient, and interoperable algorithms. Medical devices are resource-constrained and running the conventional security algorithms; it is very costly in terms of both energy and memory point of view, and maintaining the privacy of patient's health information is also challeng- ing. The absence of real-time data processing may lead to unavoidable situations and even sometimes to death. Efficient big data analytics solutions are required to maintain a vast patient's health records.

C. SMART FARMING

The agriculture industry is playing a vital role in nourishing every individual on earth [58], [59]. Most of the farmers still follow the traditional methods in farming. The use of traditional farming techniques causes soil erosion, a decrease in the yield of the crop, waste of water, waste of fertilizers, etc. IoT plays a crucial role in reducing a few problems in agriculture.

- **Precision Farming:** The factors like soil, climate, flora, and water directly affect the growth of the crop, and these factors change from one place to another. Preci- sion farming manages these factors efficiently to more crop using minimal resources. Precision farming is a combination of sensors and associated software. Sensors are used to gather real-time data from the fields, and the software is used to analyze data and manage the available resources [60].
- **Crop Health Monitoring:** Monitoring the health of the crops at regular time intervals increases the quality and quantity of the crops. Generally, most of the crops are majorly affected by fungal and bacterial diseases. IoT devices like drones and tractors are the conventional vehicles used in the agricultural fields to supervise the healthiness of the crop. These vehicles are equipped with various sensors and infrared cameras to collect crop health data and give the initial indications of the diseases. Data analysis and Image analysis techniques are used to predict the health of the crop [61].
- **Smart Greenhouse:** Nowadays, the greenhouse is one of the most popular and fastest-growing farming tech- niques. In this technique, plants are covered with the green-colored covers or glasses. The IoT based green- house enables the farmers to produce multiple crops at a time by maintaining the optimal conditions inside the cover [62]. This empowers the farmer to produce a high yield with better quality.
- **Live Stock Monitoring**: IoT based livestock monitor-ing enables real-time collection and analysis of animal data, which improves the health and productivity of the animals [63]. Integration of sensor and IoT technology allows the farmers to get the information on trespassing, digestion, grazing behavior, and other vitals of livestock. Various sensors are used in agriculture to

measure soil and environmental factors, integrating these multiple sensor val- ues help in crop management. Several deep learning models have been developed to identify and analyze the crop diseases and weeds at the earliest stages. However, deep learning models should consider environmental factors to analyze the severity level of diseases and weeds. Effective prediction algorithms are required to predict the changes in the environ- ment and market trends. Drones and robots are more common in smart agriculture. The design and development of low- cost intelligent drones and robots are essential to support regular agricultural activities (Monitoring the crops and live stocks). ML algorithms play a vital role in precision feeding by analyzing the grazing behavior of the livestock, which provides the required nutrients at the right time. Efficient algorithms to analyze the behavior patterns of the livestock are also necessary.

D. INTELLIGENT TRANSPORT SYSTEM (ITS)

ITS is one of the major applications of IoT, which comprises of different forms of transportation like Road, Air, Sea, and Rail. ITS offers services like Traveler Information, Traffic Management, Electronic Payment, Information Exchange, Freight Management, and V2V communication [64]. These services improve the human being's quality of life in terms of transportation using IoT technology.

- **Traffic Flow Monitoring and Controlling:** It is one of the significant elements of the ITS [65]. An increase in vehicle density in the cities is making traffic monitoring and controlling essential. Monitoring and controlling the congestion using traffic signaling system is the target
of this application. Cameras mounted on the roads are used to gather the congestion data, and present situations are analyzed using ML and AI algorithms [65]. This analyzed information is shared among the travelers.
- **Traveler Information System (TIS):** TIS offers infor- mation about the location, traffic regulations, routes, emergency services, safety, and warning measures to the traveler's [66]. TIS is meant to offer more precise infor- mation for travelers, thereby reducing the various hurdle in real-time [67]. TIS is a combination of technologies like WWW and GIS [68]. WWW allows travelers to retrieve required data anywhere at any time, and GIS is a system used to store, analyze, and represent the geographical data graphically.
- **Freight Management System (FMS):** IoT enables the gathering of real-time information [69] about the freight along with its transportation chains like the sea, and air. The use of IoT technology in FMS increases operational efficiency, maintenance, reduces electric consumption, and adapt to the environmental changes [70]. The FMS is divided into categories like Commercial Vehicle Oper- ation (CVO) and Advanced Fleet Management System (AFMS). The CVO systems are designed to exchange safety information, electronic certificate supervision, and automated wayside inspection. AFMS is used to automate and reduce the complexities in the freight car-riage operations [71].
- **V2V Communication:** The trend is now moving from the use of traditional vehicles to independent vehicles (Autonomous vehicles). Vehicle to Vehicle (V2V) com- munication plays a key role in the sharing of reliable information between the vehicles and the infrastruc- ture [72]. V2V communication provides 360-degree information about the vehicles and associated infras- tructure [73]. IoT alerts the drivers in the case of road hazards, accidents, traffic congestion, and emergencies in V2V communication [74].

Real-time data collection is challenging, inaccurate, and unreliable certain times in ITS; this is due to the high mobility of the vehicles. Real-time accident detection, traffic flow monitoring, and prediction are difficult due to the randomness of occurrence. Effective automatic incident detection and classification using roadside surveillance cameras make ITS more valuable. Several algorithms have been developed, such as SVM, KNN, ARMA, ANN, CNN, and many more to detect and analyze the incidents. However, accuracy is the prime concern in prediction algorithms. Data privacy and security are essential for ITS since people use the various ser- vices provided by the government and private organizations which collects the individual's and vehicle's private informa- tion such as location details, etc. Therefore, the development of privacy laws and efficient security algorithms need to be implemented to escalate the security feature. This application mainly demands the availability of the communication net- works and ruggedness in the IoT system.

E. FACTORY AUTOMATION & INDUSTRY 4.0

Industry 4.0 is the new stage of the industrial revolution that uses real-time data, machine learning algorithms, automation techniques, augmented reality, and interconnectivity tech- nology to improve the overall efficiency of manufactur- ing [75], [76]. Industry 4.0 is also referred to as Industrial IoT (IIoT). Industry 4.0 provides benefits like 1. shorter develop- ment cycle, 2. efficient resources utilization, 3. Decentraliza- tion to take swift and better decisions, and 4. flexibility in product development [77]. Factory automation and industry
4.0 provide services like plant safety and security (Surveil- lance), product monitoring, supply management, and quality control.

- **Plant Safety and Security (Surveillance):** Manufac- turing industries face a massive range of dangerous cir- cumstances, which cause severe injuries to the workers and manufacturing plants [76]; this leads to a substantial financial crisis and human loss. IoT based surveillance systems capture possible hazardous situations and ana- lyze the cause of injury to the workers and manufac- turing plants. The IoT based surveillance systems are capable enough to inspect the situation like fire acci- dents, caught in machine, overexertion, and slip & fall accidents.

- **Product Monitoring:** The product development life cycle is shorter in industry 4.0, and it needs to be monitored carefully. The product monitoring using IoT helps in reducing the bottlenecks in the production, optimizing the production process, and prevents the problems before manufacturing [78]. **Supply Chain Management:** IoT play a fundamental role in supply management by implanting sensor intelligence to the products and machinery. These sensors are connected together and also to the Internet. This advanced connectivity empowers real-time information tracking and sharing of the product information, which elevates the supply management operation [79].
- **Quality Control and Management:** Quality control and management ensures that the product is free from defects. The quality of the products and services is crucial to attaining sustainable economic success [80]. In this, IoT is used to monitor and analyze the quality of the products at critical points and alerts if it founds substandard quality materials [81].

Still, most of the companies are making efforts to adapt to Industry 4.0 but, Industry 5.0 is the future, which enables the collaboration between the people and machines to achieve business milestones. Device management is an essential requirement with the support of middlewares, which ensures the smooth operation of the machines in Industry 4.0. Scalable device management techniques are in demand. Efficient algorithms are required for the robots to understand the surrounding environment and exchange the information efficiently with the other robots and human workers in Industry 5.0, which increases safety in highly complex tasks.Rarely, security threats may lead to manufacturing and operational disruption, which might lead to a financial crisis. Industry 4.0 requires effective threat detection and system recovery algorithms, along with self-debugging and reconfigurability in the system.

F. ENVIRONMENTAL MONITORING

The growth of population is higher in the urban area in all over the globe which results in urbanization. Air and water pollution is increasing, and meteorological conditions are degraded due to the urbanization [82]. The standard parameters like atmospheric temperature, humidity, pressure, quality of the air and pollutants like carbon dioxide and carbon monoxide are being monitored [83].

- **Forest Fire Detection and Mitigation:** It has been predicted that 80% of the forest loss is due to the forest fire. IoT can be used to build the early fire detection models and alert the fire brigades to take necessary measures [84].
- **Smart Water and Sanitization:** Both developed and developing countries are facing water distribution and sanitation service problems. IoT solutions like digital meters and geographic information systems increase the effective utilization of water and sanitation services [85], [86].
- **Outdoor Air Quality Monitoring:** Industrialization and urbanization also lead to air pollution in recent years. Poor quality in the air increases lung, heart and skin diseases. IoT technology allows the evaluating of toxic and flammable gas proportions along with the concentration of air pollutants regularly [87]. **Natural Disaster Management:** Natural disasters like earthquakes, hurricanes, floods, etc. are more common, which results in loss of habitats, lives, and properties. IoT technology plays an essential role in providing services like early warning systems, immediate responses to the victims, emergency medical care, etc., [88], [89].

Sampling, analysis, spatial and temporal errors are more common at the time of measuring the environmental parameters. Decision making is efficient when these errors are eliminated from the dataset. These sensors produce a heterogeneous and enormous volume of data. Therefore, intelligent algorithms are required to clean and preprocess these types of data; this increases the analytical capabilities. Further attempts are essential to implement light-weight algorithms for big data analytics in environmental monitoring. The lifespan of the sensors used in environmental monitoring is reducing due to the meteorological properties. Efforts should be made to increase the durability of such sensors. Along with the above challenges, security issues also need to be considered.

G. SMART CITIES

A smart city aims to advance the urban economy, infrastructure and governance, quality of life (Physical, Mental, and Financial well-being), social and environmental smartness [90], [91].

- **Smart Infrastructure:** Infrastructure is the basis for the successful implementation of smart cities. The infrastructure like intelligent electric grids, water grids, and transport infrastructure constitutes towards better and modern societies [92]. Smart infrastructure improves the understanding and control of the operation and optimizes resource utilization in a city.
- **Smart e-Governance:** It enables faster decision making in government organizations and schemes, transparency in the governing agencies, and accessibility of the public services [93].
- **Surveillance:** Surveillance plays a vibrant role in a smart city, which reduces the rate of crimes. In a smart city, surveillance cameras record the individual's activity in the crowded areas to detect and prevent crimes, where public safety officers cannot position; this increases public safety [94].
- **Smart Community:** It provides the essential services like energy and fuel management (green buildings and renewable sources of energy), water management (water quality monitoring, leakage identification, and smart meters), waste management, etc. to the residents [91].

Real-time data processing and analysis, reliable communication, availability, device management are the key enablers for the success of smart cities. Improved Self-learning algorithms are much needed to recognize and analyze the multiple events in real-time videos, detecting the damages in the buildings, real-time traffic flow monitoring, and control. Also, efficient mechanisms

are essential to enhance on-demand services such as transportation, vehicle parking, and emergency ser- vices. Generally, the IoT system needs to handle video, audio, and signal processing efficiently while maintaining resourceconstraints.

H. INFRASTRUCTURE MONITORING

The purpose of infrastructure monitoring is to collect real- time information related to civil structures such as buildings,bridges, monuments, tunnels, railway tracks, manufacturingand construction process to avoid the risks [95].

- **Oil/ Gas Pipeline Monitoring**: Pipelines are used widely for effective transportation of crude oil and gas;these pipelines need to withstand severe environmental conditions. Monitoring failure in oil and gas pipelines may lead to severe ecological disasters and financial losses. IoT smart sensors are used to monitor the oil/ gaspipelines for cracks and leakages [96].
- **Monitoring of Civil Structures:** Monitoring of civil structures involves various stages such as detecting, locating, identifying the types of damages, and quantify- ing the severity of the damage. Civil structure monitor- ing looks for parameters like stress, displacement, cracksin civil structures like bridges, railway tracks, buildings,monuments, etc., [97].
- **Military Assistance and Surveillance:** IoT also plays a major role in military operations to identify the sub- ordinates during the search and rescue operations [98]. Traditional military bases can be converted to smart mil- itary bases by deploying the IoT devices; this increases the safety by automating the security screening tests. Intelligent surveillance systems allow armed forces to recognize threats swiftly with higher accuracy.
- **Aircraft Health Monitoring System:** A large number of public and military aircrafts have been surpassed their design lifetime [99]. It is essential to monitorthe health of the aircraft structure to increase oper- ational efficiency. Modern aircraft engines compriseof 5000 sensors and generate roughly 10GB of dataper second [100]. IoT plays a key role in aircraft struc- tural monitoring by collecting data from the sensors andanalyzing them. Analysis results in efficient decision making and communicating the decisions to the aircraftauthorities.

Data acquisition and fusion, real-time data processing and analysis, device management, reliable connectivity, and autonomous robots are significant in infrastructure monitor- ing. Efficient deep learning algorithms need to be developed to detect the damages in real-time. Intelligent robots need tobe designed to depute in rescue operations. Efficient sensor data acquisition and fusion techniques are the major require-ments, which fasten and simplifies decision making. This application requires massive deployment of IoT devices in the fields. The massive IoT device deployment requires clus- tering, network availability, and medium to high computa- tional requirements. Efficient device management techniquesand light-weight data mining algorithms make device and data management easier in infrastructure monitoring systems. This application also prone to privacy and security attacks; the IoT system should be tampering free.

I. SMART RETAIL

Smart retail improves the customer's experience by using communication, information, and augmented reality tech- nologies at the time of shopping. Generally, smart retail shopscomprise of RFID tags, interactive displays, recommended systems, and self-cash desks [101].

- **Sensor-Based Stock Maintenance and Tracking:** Smart sensors mounted on the smart shelves allow the customers to find items with no efforts. Generally, RFIDtags are attached to the items, which are linked to the store computer system; this allows the store owners to keep track of the items [102].
- **Automated Checkouts:** This allows the customers to pay for their purchases with little or no time. Auto- mated checkouts are relying on linear or barcodes for the identification of the items and their price. Automatedcheckouts use self-checkout terminals or tunnel systems. In the self-checkout terminal, customers need to scan theitems. In the tunnel system, cameras scan the barcodes of the items. Some retail shops allow their customersto scan the items while customers are moving by using mobile applications [102].
- **Recommender Systems:** Recommender systems sug- gest the set of items based on the potential interest of the customers. In this, potential customer interests are predicted based on the previous purchases data [103].

In the future, technologies like Virtual Reality (VR), Augmented Reality (AR), and intelligent robots play a crucial role in enhancing the overall customer's shopping experience. These technologies bring new features like virtual trial rooms, provide additional information about the products, search for the best deals of the day, check for the availability of items, and so on. The retailers have access to the customer'sdata for various purposes; this raises privacy and security issues. The development of advanced methods to analyze the real environment to create a virtual environment are the primary requirement. Efficient algorithms to analyze the cus-tomer's mood swings based on the availability of the data areessential.

J. SMART POWER AND WATER GRIDS

The power grid is an interconnection of power producers, distributors, and consumers. Producers gather the electrical energy from various sources like solar, wind, thermal nuclear, and hydroelectric power plants and distribute them among the consumers using power transmission lines. Multiple sensorsand control devices are used to monitor the activities of producers, consumers, and distributors; this saves energy, management, and distribution cost [104].

A smart water grid is the networking of water reservoirs (grids and micro-grids), distribution centers, and purifica- tion plants, which are equipped with the sensors and controldevices to manage and monitor the water distribution system.

TABLE 2. Classification and characteristics of IoT applications.

Characteristics	Emerging IoT Application									
	High Data rate Applications					Moderate Data rate Applications		Low Data rate Applications		
	Smart Health	**Smart City**	**Factory Automation/ Industry 4.0**	**Infrastructure Monitoring**	**Smart Power and Water Grids**	**Intelligent Transportation**	**Environmental Monitoring**	**Smart Homes**	**Smart Farming**	**Smart Retail**
Internet Connectivity	WiFi, 3g, 4g, BLE, Zigbee, Z-Wave	WiFi, GSM, Satellite Communication	BLE, Zigbee, WiFi, GSM Satellite Communication	WiFi, GSM Satellite Communication	WiFi, GSM, Satellite Communication	WiFi, GSM, Satellite Communication	WiFi, GSM Satellite Communication	WiFi, GSM, BLE, Zigbee, Z-Wave	WiFi, GSM, Satellite Communication	RFID, BLE, WiFi
Network Size	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (>10000 Nodes)	Very Large (< 10000 Nodes)	Very Large (<10000 Nodes)	Very Small (10 Nodes)	Large (Up to 1000 Nodes)	Small (Up to 1000 Nodes)
Analysis	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distribute	Local & Distributed	Local	Local & Distributes	Local and Distribute
Acceptable Delay	Seconds	Seconds to minutes	Seconds to Days	Seconds	Seconds	Seconds to minutes	Seconds to Minutes	Seconds to Hours	Seconds to 1Day	Hours
Type of Data	Historical, Stream	Stream	Stream	Stream	Stream	Stream	Historical, Stream	Historical	Historical, Stream	Historical
Computing Paradigm	Edge, Cloud	Edge, Fog, Cloud	Edge, Cloud	Edge, Fog, Cloud	Edge, Fog, Cloud	Edge, Fog, Cloud,	Edge, Cloud	Cloud	Edge, Cloud	Edge, Cloud
Technologies Used	SP, IP, AI (DL)	SP, IP, VP, AI (ML), Digital Twin	AI (DL) Robotics	SP, IP, VR, AR, Robotics, Digital twin	SP, Control algorithms, AI	SP, IP, VP, VR, AI	SP, IP, ML	SP, IP, ML	SP, IP, VP, Robotics, Drones	IP, VP, Drones, VR and AR, AI, Robotics
Location awareness & Sharing	✓	✓	✓	✓	✓	✓	✓	-	-	✓
Privacy and Security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sensors used	Bio, Pressure, Temperature, SPO2, Chemical Sensors	Environmental, Vehicular and Infrastructure Sensors	Cameras, Temperature, Gas, Fire Sensors	Accelerometer, Displacement, Strain, Inclinometer, Temperature, Laser Sensors	Accelerometer, Displacement, Strain, Inclinometer, Temperature, Laser Sensors, Current and Voltage Sensors, Leak noise localizers and leak noise correlators, Water Level Sensors	Cameras, Ultrasonic, LiDAR, RADAR, GPS, Infrared, Fuel Sensors	Moisture, Pressure, Tilt, Rain, Voltaic Organic Compound sensors, and Geno Sensors	Accelerometer, Temperature, Activity Sensors	Chemical, Temperature, Pressure, Tensiometer, Airflow Sensors	Cameras, Ultrasonic, NFC

SP – Signal Processing, **VP** – Video Processing, **Global System for Mobile Communication (GSM)** – 3G/ 4G/ 5G, **IP** – Image Processing, **AI** – Artificial Intelligence, **VR**-Virtual Reality, **AR**-Augmented Reality, **ML**-Machine Learning, **DL**–Deep Learning

- **Power and Water Grid Monitoring:** Power grid mon- itoring performs activities like power loss monitoring during distribution, load balancing, and metering func- tion to enable safe and efficient power delivery to the customers. Similarly, water grid monitoring comprises monitoring the parameters like pressure, quality of the water, flow, and leakage 105].
- **Management and Distribution:** Power and Water management systems continuously monitor the usage of electrical energy and water resources, respectively. This monitored data is analyzed and used to provide on-demand services to consumers; this ensures adequatelevels of power, and water resources are available to theconsumers [106].
- **Smart Metering:** It records the electrical energy (volt-age levels and power factor) and water usage details automatically and communicates details to the con- sumer. Data analytics is performed on this data to enhance their operational efficiency potentially [107].

Most of the power grids use the Remote Terminal Unit (RTU) to monitor and control the devices in the smart grid. However, integration of the Phasor Measurement Unit (PMU), commu-nication, and advanced computation technologies assist the better monitoring of the power grids. Autonomous control of the power grid substation is a primary requirement; to achieve this requirement, efficient predictive algorithms need to be developed to understand the customer's electricity demands. Efficient sensor fusion techniques are required to maintain the water quality and quantity to maintain the water flowin smart water grids. Low power water grid monitoring and management techniques are of high priority. Additionally, efficient prediction models need to be developed to predict the water necessity for the urban areas. In the future, both powerand water grids may be integrated; this is due to reducing theenergy consumption in households and industries.

Table 2 describes the typical characteristics and classi- fication of IoT applications [23], [108]. IoT applications are classified based on data rate such as high, moderate, and low data rate applications. In most cases, high data rate applications include signal, image, and video processing technologies. For example, A typical High Definition (HD) video of resolution 1920*1080 pixels, color depth of 24 bits,and 30 frames per second require a 1920*1080*24*30 = 186.624 Mbps data rate. Similarly, to transfer an HD image over a network with 1920*1080 pixels resolution and color depth of 24 bits requires a data rate of 1920*1080*24 = 6.221 Mbps. In moderate data rate IoT applications, the major data sources are images, audio, accelerometer sensor, etc. andrelated data, which requires preprocessing. For example, an audio signal with a sampling rate of 44100 Hz, and each sample consists of 16 bits per sample; if we consider stereo, two channels will be used. The total data rate 16*44100*2 = 1.41 Mbps. Normally, the low data rate application comprises of discrete data values with floating-point representation, andthe sampling rate of these values is less than ten samplesper second, i.e., the required data rate is 4*8*10 = 320 bps.

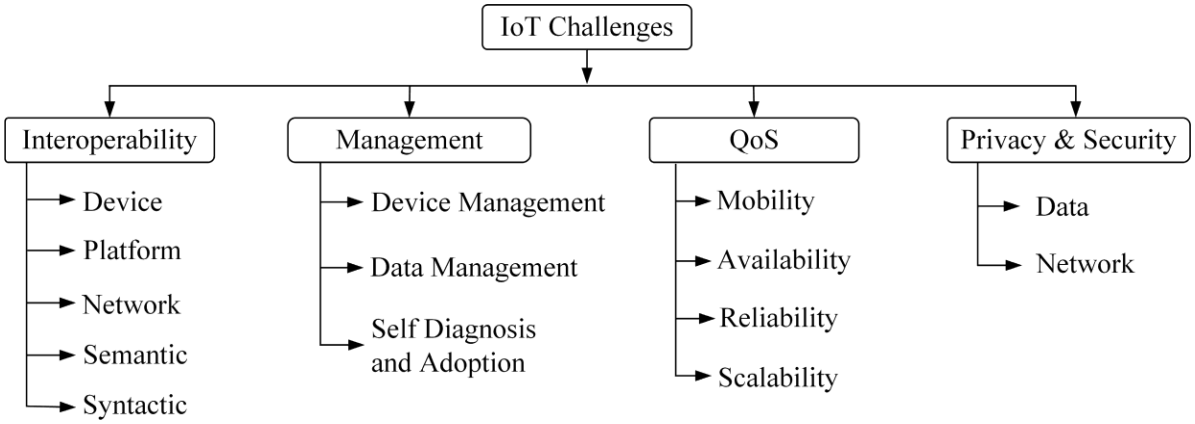


FIGURE 9. IoT challenges.

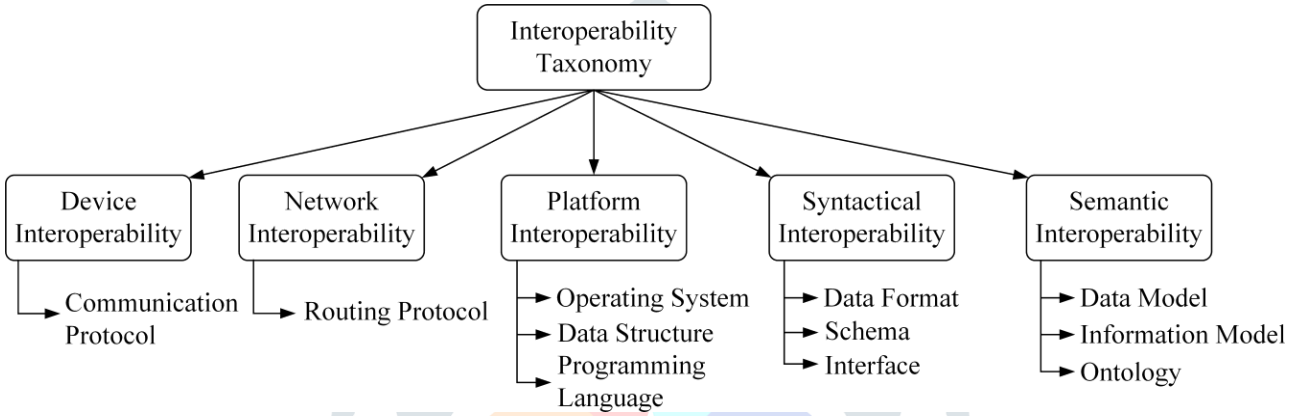


FIGURE 10. IoT interoperability taxonomy.

The IoT devices should employ the necessary communica- tion bandwidth, and data preprocessing and data computation techniques, along with the system design requirements listedin Table 3.

TABLE 3. Requirements of IoT applications.

Type of IoT Application	Requirements
Low Data rate	Heterogeneity, Self-Diagnosis, Privacy & Security
Moderate Data rate	Heterogeneity, Self-Diagnosis, Privacy & Security, Mobility, Device Management, Scalability
High Data rate	Heterogeneity, Self-Diagnosis, Privacy & Security, Mobility, Device & Data Management, Availability, Scalability

IV. IoT APPLICATION REQUIREMENTS AND CHALLENGES The widespread of IoT is because of its benefits like sensing, processing, communicating, and actuating capabilities. The issues like Mobility, Interoperability, Privacy & Security, Storage Management, and Device management are rising due to the use of sensors from different vendors, processing platforms, and various communication standards and proto- cols [109]. To achieve the IoT features such as sensing and actuation, unique identity, ubiquity, communication, intelli- gence, self-configuration and adaption and effective service delivery following challenges need to be considered. Figure 9shows the taxonomy of the IoT challenges.

A. INTEROPERABILITY

Interoperability remains a big challenge due to the use of heterogeneous devices in terms of underlying communica- tion standards and protocols, data formats, and technolo- gies [110]. Interoperability is the ability to exchange and make use of information irrespective of the hardware and thesoftware platforms [4]. This issue should be considered in all the layers of the IoT system, i.e., Device interoperabil- ity needs to be addressed at the perception layer, network interoperability should be handled at the transport layer, andsyntactic and semantic interoperability should be taken care at the application layer of IoT three-layered architecture. Interoperability gains the widespread due to 1. Availability ofthe heterogeneous devices in the market,

2. The rapid devel-opment of IoT applications, and 3. Lack of standards [47]. Approximately 47% of the issues can be solved if an IoT sys- tem attains 100% interoperability, which helps IoT services to meet customer expectations [111]. Figure 10 describesthe taxonomy of IoT interoperability. Open Connectivity Foundation (OFC), ETSI (ETSI TR 103, ETSI SR 003, ETSI TS 103), oneM2M (TS-0013), W3C organizations are making their continuous efforts to nullify the interoperability issue [112].

Bröring *et al.* [113] proposed a cloud-based BIG IoT architectural model to address the platform interoperabil- ity. This architecture offers communication between the different platforms by creating standard APIs. Semantic web technologies were used by the architecture to enable the interoperability between the IoT applications, platforms, and services. The architecture consists of Bosch, Consorzio peril Sistema Informativo (CSI), Siemens, Verkehr Mobili- tat Zukunft, and World sensing platforms. The architecture also contains features like resource registration and discov- ery, authentication, and authorization. Derhamy *et al.* [114] analyzed the existing solutions to achieve protocol interop- erability. The authors have proposed a secure, low latency on-demand transparent multi-protocol translator service for Industrial IoT. The proposed translator architecture was based on the service-oriented architecture, and the translators were located at the cloud layer of IoT. The translator uses infor- mation like interface design, service description, communi- cation profile, and semantic profile to provide services to the producers and consumers. Yang and Wei *et al.* [115] presented a Tabdoc approach, which enables the semantic interoperability between the IoT users and devices for data accessing and controlling. In this approach, the messages exchanged between the IoT devices and the users were con-sidered as the semantic documents. The proposed method uses the Semantic Extraction Algorithm (SEA) to get com- plex semantic document components and Semantic Inference Algorithm (SIA) to implement automatic semantic documen- tation interpretation. SEA provides the mapping between the Tabdoc and non-Tabdoc documents, and SIA performs the automatic cross-context semantic interpretation using thesemantic inference.

B. MANAGEMENT

The application like factory automation/ Industry 4.0 com- prises of numerous devices; the status of these devices needsto be monitored for any failures. Time to time software updates needs to be maintained in the devices, etc. These devices also generate an enormous amount of data with dif- ferent data formats; this needs to be taken care of IoT device and data management. The organizations like ISO and IEC are working jointly to address Big Data issues and formation of standards. These organizations formulated the standard like ISO/ IEC JCT 1/WG 9 for effective data management in IoT [116]. Open Mobile Alliance (OMA) proposed a device management standard to perform device management in the IoT network [117], [118].

1) DEVICE MANAGEMENT

Device management allows the network administrator to per-form activities like Device Monitoring & Diagnostics, Soft-ware Updates & Maintenance, Device, and Configuration &Control. Device management has become challenging due to
1. Rise in the number of devices, 2. Heterogeneous devices,
3. Device mobility, 4. Dynamic topology and 5. Numerous firmware updates.

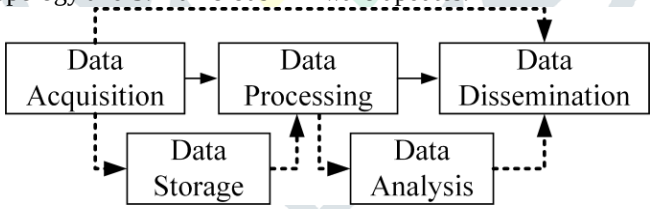


FIGURE 11. Data management life cycle.

Yoon *et al.* [119] proposed an architecture to manage IoTsensors and their associated data using a Fog computing. Devices, Fog node, and Operational & Management Server (OMS) are the elements of this architecture. The devices capture the data from the environment and transfer it to the fog node for processing. The fog node acts as a monitoring node, which performs the activities like data management, connection management, and component monitoring. OMS is responsible for collecting the anomalous values with the timestamp from the fog node and send the reports to the administrator. Haus *et al.* [120] developed an edge-based plat- form iConfig to administrate the IoT devices in smart cities. This platform addresses issues like registration, configuration and maintenance of the IoT devices. iConfig platform focuses on the management of the BLE devices and provides services like device status, device functionality and localization of the devices. The edge module uses the centralized queue to synchronize the user actions and collects the maintenance data to identify the beacon status and broken beacons.

2) DATA MANAGEMENT

Data management [121] plays a leading role in handling the huge volume, variety, and velocity of data generated by IoT applications [122]. Collecting, storing, processing, analyzing,visualizing the data, actuation, and communication are the different steps in the IoT data management life cycle [123]. Figure 11 depicts the data management life cycle in IoT.

a. Data Collection and Storage: Internet, sensors, satellites, wearable devices, and cameras are the major sources of data [124], [125], and these data can be accumu- lated in the local databases or cloud database. The data may be structured, semi-structured, and unstructured. SQL databases are generally used to store structured data. The most common type of data produced by IoT devices is Semi-structured and Un-structured data. Usu- ally, NoSQL databases are an excellent choice to store IoT data. NoSQL databases have a focus on reading and writing rather than organizing the data. MongoDB,HBase, Hive, Redis, Neo4j, and Cassandra are examplesof the NoSQL database [126].

b. Data Processing: Data processing can be done locally (where data is generated), using edge nodes, fog nodes, cloudlets, and cloud platforms [127]. Data fil- tering and aggregation are crucial operations in data processing [128]. Apache Storm, Spark, Rapid Miner, and High-Performance Computing Cluster (HPCC) are the tools used to process the data in real-time scenarios.

c. Data Analysis: Data Analysis is the critical aspect to bring out the hidden patterns from the processed data [128]. This data is beneficial to control and actuate the decisions. IoT systems should offer both offline and online data analysis methods [123]. The different analysis techniques like streaming, spatial, time series and prescriptive analysis are more common in IoT.

Streaming: This technique takes actions by perform-ing the analysis of real-time data using continuous queries.

Spatial Analysis: It is a process of analyzing the attributes and their relationship of an entity based on the location information.

Time Series Analysis: This technique analyses the data captured at different time intervals.

Prescriptive Analysis: It is used to automate the recommendations and make better decisions based on the findings of the analytical models.

3) SELF DIAGNOSIS AND ADAPTION

Self-diagnosis and adaption are the important features of IoT, which identify and minimizes faults in the IoT system. Self-diagnosis is the ability of an IoT device to detect and analyzethe glitches in it or in the IoT system. Self-adaption is the ability of the IoT devices/ system to customize itself to the changes in the hardware, software, and resources as and when required [129], [130]. For example, self-diagnosis and adap-tion play vital role in infrastructure monitoring. A dedicatedIoT node can be used to monitor the status of the sensor nodes using machine learning algorithms. This runs troubleshooting algorithms automatically to solve the issues.

Lee *et al.* [130] proposed a self-diagnosis technique to identify the defective device in the IoT network. The pro- posed technique deals with the identification of memory, processor, and peripheral faults. Monitoring agents and mon- itoring programs are the major blocks of the technique. The monitoring agents collect the information of the IoT devices. It keeps on updating the device information to the monitoring program, which is running in the processing unit. The pro- cessing units (sophisticated servers or cloud platforms) are rich in resources. The monitoring program analyses the call stack to identify whether the device is working correctly or not. Also, the monitoring program analyses exceptions that occurred during the execution of the code and state of the devices (Hanging state, etc.). Huang *et al.* [131] proposed Gaussian Bernoulli restricted Boltzmann Machine (GBRBM)based Deep Neural Network with auto Encoder (DAE). The authors have extracted the features from the IoT networks automatically by using the GBRBM blocks. The learned features are merged into the encoder neural networks to identify the faulty nodes in the IoT networks. To examine the proposed algorithm, the authors have used the datasetof server machines faults collected by Google data center operators. The size of the dataset is of 41 GB with 77, 776 rows and 13 columns/ attributes. The main features used in the self-diagnosis are timestamp, missing info, machine Id, event type, priority, scheduling class, and task index. Theauthors have claimed that the fault detection accuracy of thealgorithms is 89.2 percent. Iftikhar *et al.* [129] proposed a DeltaIoT self-adaption model for IoT networks. The statisticsengine in the model is responsible for gathering and storing the statistics related to the IoT network. The statistics engineprovides information about the performance and energy con-sumption of all IoT devices in the network. The network set-ting engine communicates with the IoT devices via a gatewayto collect and adapt the IoT devices network settings. The authors have tabulated the generic adaption scenarios, whichconsist of the type of uncertainty, type of adaption, and adap-tion goals; these generic adaption scenarios play a major rolein self-adaption. For example, If the IoT network is facing theuncertainty in wireless interference, transmission power and modifying the path to gateway adaptions are required, and the adaption goals are reducing the packet loss and energy consumption in the IoT network.

C. QUALITY OF SERVICE (QoS)

In IoT, QoS manages the network capabilities and resourcesto provide the required services to IoT users. The use ofQoS in IoT results in effective utilization of the services, resources, reduces the network delay and traffic. QoS helps service providers to describe the services based on customerneeds [132].

1) MOBILITY

IoT devices like smartphones, laptops, autonomous cars,robots, and drones are highly mobile in nature [133]. Mobiledevice

management is difficult as compared to static device management in IoT. Mobility is divided into micro and macro mobility. In Micro mobility, devices are roaming from one gateway to another gateway within the network, but in the case of macro mobility, devices are roaming from one net- work to other networks [134]. Protocols like Mobile IPv4, IPv6, Hierarchical IPv6, Fast mobile IPv6, and proxy mobile IPv6 were introduced to handle the mobility issues effec-tively [135]. Unfortunately, it is difficult to incorporate these existing mobility standards into IoT due to the large pro- cessing time and energy consumption. For example, mobil- ity management plays a vital role in applications like V2V communication, Automatic number plate identification, and traffic diversion, etc. in ITS. Since the vehicles change their location over time and share the information with neighboring vehicles, any information loss during mobility of the vehicles may lead to a dangerous situation or accidents. The parame- ters like signaling cost, handover latency, packet loss, end-to- end waiting time, and power depletion need to be considered while designing the mobility management mechanisms [136]. Wu *et al.* [137] presented a software-defined UbiFlow mobility management system for IoT. The system provided features like mobility management, selection of access points, and optimizing the handover mechanism using coordinated distributed controllers. The controllers are scalable, fault- tolerant, and work in resource-constrained environments.

Gia *et al.* [134] proposed an energy-efficient and low latency handover mechanism for IoT to support the mobility of the devices in remote locations. The mechanism uses the RSSI, multilevel thresholds, frame injection, and link con- nection (Bandwidth utilization and the number of nodes connected to each gateway) parameters to attain better han- dover conclusions. The gateways have identical coverage distance and specifications. The authors claimed that the proposed handover mechanism reduces the latency of switch- ing between the gateways by 10% to 50%. Fu *et al.* [138] proposed a group-based mobility management technique in which devices with the same mobility pattern were grouped together, and this was done by using the location database. Location database checks for the mobility pattern of the device when it receives the registration request and deter- mines whether the device can be merged with the existing mobility group or not. For each group, a group leader is selected to performs the mobility management of the devices. The device is moving from one local network to another; the leader node of the group will perform the registration activity.

2) AVAILABILITY

Time-critical applications like smart health, structural health monitoring, disaster management applications use IoT technology. The term availability in IoT is realized in both software and hardware levels. Software availability provides services to users anywhere and anytime. The hardware avail- ability provides the data required by the application services. For example, the Oil leakage monitoring application provides real-time oil leakage detection services; these services make use of the hardware devices and sensors like GPS, infrared sensors, and cameras to detect the leakages. This application completely relies on the data from these sensors. Here, both hardware and software availability play a significant role.

To ensure data availability during the node failure, Qaim and Ozkasap [139] proposed a distributed hop by hop data replication technique (DRAW). This technique selects the best replica node to maintain a copy of the data items. The technique applies a series of conditions on the degree of repli- cation, availability of the memory in the device, a number of hops, common neighbors of the devices, and previous replicas of the data items. DRAW technique was simulated using the NS-3 tool. Wei *et al.* [140] proposed an environmental- based method for modeling the services in IoT. The pro- posed method uses the WSMO-Lite that characterizes WSDL based web services and RESTful APIs. WSMO-lite is used to model the dynamic context and availability of the services. At first, the method is used to capture the features of the context, and based on this context, the availability of the services is represented in three levels: 1. Two-dimensional properties (Location and Time), 2. Resource and Mobility, and 3. Dynamic behavior of the services.

3) RELIABILITY

Reliability is the ability of the components (Software, Hard- ware, and Network) in the system to perform its essential functions at different conditions and specific intervals of time [141]. Reliability requirements need to be discussed in each layer of the IoT architecture. For example, in fac- tory automation/ Industry 4.0, the devices are working in cooperation to achieve the manufacturing milestone. Here, communication plays a significant role, and loss of connec- tivity between the devices creates a severe problem in the manufacturing process.

Al-Kadhim *et al.* [142] proposed four scenarios to min- imize the energy depletion and enhance the reliability at the network layer of the cloud-based IoT network using the Mixed Integer Linear programming (MILP) model. In the route selection process, the failure nodes are identified and replaced with the standby nodes to increase the data deliv- ery rate. The desired reliability level scenario ensured that the selected route would be 99% reliable. The reliability- based sub-channel scenario reduces the overhead on the reli- able routes by employing multiple transmission channels. Reliability-based data compression scenario uses the Sequen-tial Lossless Entropy Compression (S-LEC) technique to minimize the size of the data, which leads to a reduction in transmission power. All these scenarios use the routing path between the IoT device and cloud. Sinche *et al.* [143] designed a structural redundancy-based reliability model, and this model is designed to enhance data reliability and com- munication reliability. Generally, sensors are connected to the gateway using the communication links. Gateways are responsible for connecting the sensors to the central servers, and it is the target of the hackers. In the first

scenario, each gateway is linked to the backup gateway (Gateway and the backup gateway uses the Master-Slave Configuration). In the second scenario, the backup link is created for each main link connecting each gateway. Finally, in the third scenario, reliability can be achieved using reliable commu- nication links and gateway redundancy. Ansre *et al.* [144] proposed a two-way Dynamic Collaborative Spectrum Sens-ing Algorithm (DCSSA) to enhance the energy efficiency of data transmission in licensed channels. In this, both source and destination use Channel State Information (CSI) to improve the reliable message transmission at higher data rates. The use of significant CSI results in the sensing of chan- nels at low SNR regimes and multiple secondary searches for the availability of the licensed channels collaboratively to reduce energy consumption. The secondary users (SU's) finds the availability of the spectrum channel by sensing the busy tone among the primary users and listening for an idle channel. To ensure data transmission reliability, the SU's share the information among them and maintains records of the packets sent by them and its time.

4) SCALABILITY

Growth in the number of IoT devices and their associ- ated data results in scalability issues [145]. IoT scalabil- ity is characterized as a system's ability to manage the increasing number of devices and their information. The technologies like cloud, fog, and edge computing can be used to tackle scalability issues in IoT. For example, net- work devices and related data are increasing in the appli- cation like smart cities. IoT infrastructure should accept, process, and respond to the requests from the devices and users without much delay (Average waiting time should be reduced).

Guo *et al.* [146] projected the transparent computing- based IoT architecture to develop scalable and manageable IoT applications. This architecture enables the centralized supervision of resources like Operating Systems (OS), ser- vices, application data, and on-demand services to run on the devices. This architecture consists of end-users, edge network, centralized network, service & storage, and man- agement layers. An end-user layer is a collection of IoT devices. The edge network contains the high-performance routers and small-scale servers to perform several tasks. The centralized network layer bridges the edge, service, and storage layer using different communication standards. The service & storage layer is accountable for data storage and providing the application services to the customers based on the necessities. The management layer is liable for managing all the application services and assigning the jobs to the con- trol server. Canedo and Skjellum [147] added the scalability feature to the gateway by using parallel computing. In this architecture, the devices that exist in the perception layer are connected to the gateway in the transport layer. To create scalable architecture, the authors presented the concept of the scalable gateway to handle data from an enormous number of devices. The Jetson TX1 is used as the gateway, and twelve Arduino Uno devices are connected to it. The gateway uses parallel computing techniques to handle the data coming from diverse devices effectively. The authors claimed that the use of high-performance gateways with parallel comput- ing techniques increases the performance (data collection, processing, and analysis) and allows us to connect a greater number of devices. Tiurlikova *et al.* [148] proposed a tech- nique to boost the scalability of the LoRaWAN by selecting the efficient Spreading Factor (SF). SF is used to specify the chirp rate, and LoRa SF can choose from the SF7 to SF12. Lower the SF higher will be the data rate. In this approach, the selection of SF is based on the PER 0.01 criterion and PDR. PDR is the ratio of the number of packets received by the gateway and the number of packets sent by the devices. This approach enables the use of a greater number of devices by selecting the appropriate SF.

D. SECURITY AND PRIVACY

IoT applications are more common in society and most likely subjected to diverse types of security attacks [149], [150]. Minor changes to IoT data may lead to serious situations. For example, the patient data in smart healthcare comprises of personal and highly sensitive health data; privacy and security of the IoT data need to be maintained. Section IX provides a detailed description of privacy and security issues.

IoT ARCHITECTURE

Several architectures have been introduced to understand the concept of IoT more precisely; among them most promi- nently used architectures are three and five-layered archi- tectures. Three and five-layered architectures are depicted in Figure 12 [151], [152].

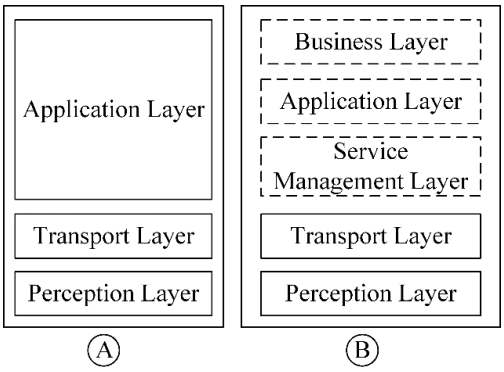


FIGURE 12. IoT layered architecture three layered & B. five layeredarchitecture).

1) PERCEPTION LAYER

The perception layer is a set of objects. Objects serve asan intermediate between the environment and the digital world by using sensors [153]. The primary motive of this layer is to capture data from the environment using various sensors such as temperature & humidity, gas, lights, GPS, camera, etc. depending upon the application requirements. The researcher's focus is on unique object identification, object management, and security in this layer.

2) TRANSPORT LAYER

This layer aims at connecting the objects and sharing infor- mation among the connected objects in a secure way. Wired or wireless communication standards like Ethernet, WiFi, Wi-MAX, ZigBee, and BLE are adapted to share the infor- mation. The reduction of energy consumption in the network,enabling quality of service (QoS), adaptation to dynamic topologies are some of the issues need to be addressed at thislayer [152].

3) SERVICE MANAGEMENT LAYER

This layer is also termed as a middleware layer, which facil-itates the use of heterogeneous devices in IoT applications. This layer also processes the raw data recorded by the objectsin the perception layer. The typical characteristics of the datacaptured are of enormous volume and varied [152].

4) APPLICATION LAYER

This layer is responsible for providing application-specific services to end-users. For example, Smart Homes Application offers services such as home automation, intelligent garden-ing, surveillance, monitoring of older people, and others.

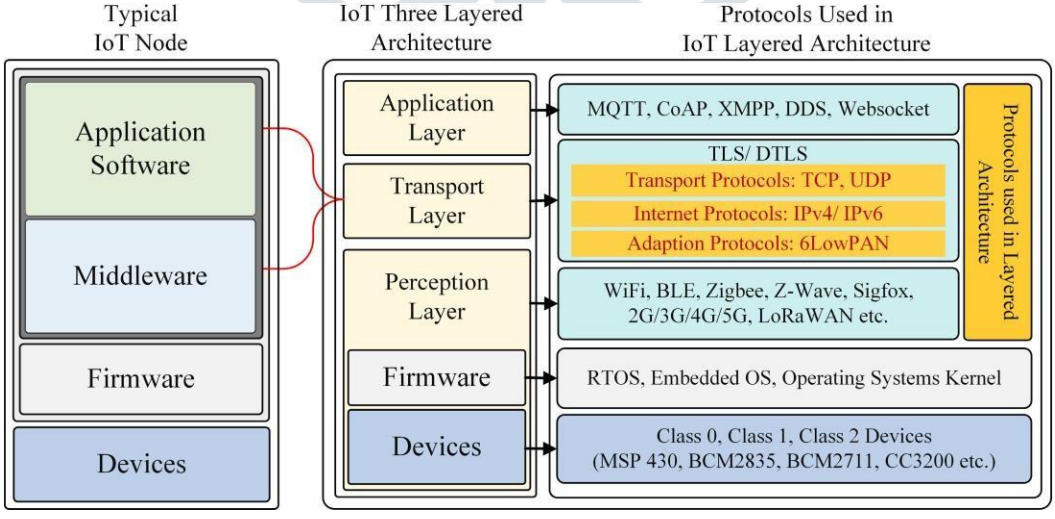


FIGURE 13. Relationship between device, operating system and IoT protocol stack.

5) BUSINESS LAYER

This layer supervises the IoT system's operations and its var-ious services; this creates business models, flow charts, and graphs based on the raw data acquired from other layers. This layer is responsible for analyzing, monitoring, and evaluating the IoT system and its related elements. Decision-making is one of the business layer's primary activity [7], [11].

Figure 13 illustrates the mapping of IoT node (left side) to IoT three-layered architecture (center) and corresponding protocols used in communication. Any IoT node comprises of devices (SoCs, Processors, etc.), firmware, middleware, and application software.

a: DEVICES

Devices are SoCs, Processors with inbuilt interfaces, which can capture, process, send, and receive the data. Also, the devices execute the commands with the help of human intervention or automatic. The size of the deployed IoT devices may vary from a small cuff link sized button to the big industrial machines.

b: FIRMWARE

It is a fixed set of instructions on the devices, which facilitates the users/ programmers to access the lower-level hardware by locating the operating system's kernel. A sig- nificant number of real-time, embedded, and other operating systems are available. RIoT, TinyOS, LiteOS, ContikiOS, Nut/OS, and μC/OS are the embedded OS's. The OS's like OpenTag, FreeRTOS, VxWorks, μ-velocity, and ErikaEn- terprise are the few real-time OS's. The OS's like Tizen, Ubuntu core, Android Things, Raspbian, etc. come under the category of other OS's, which are specifically designed for IoT applications. Section X provides the meticulousanalysis of operating systems for IoT and their protocol stack.

c: MIDDLEWARE

It is a software, usually comes with the operating system, which provides specific services in the form of a library. Sometimes these libraries can be modified according to the functional requirements and added as new external libraries. It is used to connect two or more applications. Generally, applications use communication stack to establish a commu-nication, which is a part of middlewares in the operating sys-tem either as an inbuilt function or additional library function.In IoT, the perception layer uses different standards like IEEE 802.11, IEEE 802.15.4, RFID/ NFC, BLE4.0/ BLE5.0, Zig- bee/ Zigbee Pro/ Zigbee 3.0/ Zigbee NAN, LTE-A, Z-Wave, EPCglobal, MiWi, WirelessHART, Thread, Ant+, Wi-SUN, LiFi, Ingenu, Insteon, Telensa, Wireless M-Bus, NB-Fi, GSM, NB-IoT, Sigfox, LoRa, DASH7, Weightless, etc. Inter-net protocols like IPv4 or IPv6 are used in the internet layer; these protocols are responsible for assigning the unique identity called IP address to the devices and transmission of packets over the internet. 6LowPAN is also called as IPv6 for low power wireless devices like BLE, Zigbee, and Thread. 6LowPAN performs encapsulation and header compression in low power wireless networks. The transport layer in IoT uses protocols like TCP and UDP. The security protocols like TLS and DTLS are used by the TCP and UDP, respectively, to protect the communication channels. The application layer protocols like MQTT, Secure-MQTT, CoAP, XMPP, DDS, Websocket, AMQP, Lightweight M2M, Simple Text Ori- ented Protocol, Simple Media Control Protocol, Simple Sen- sor Interface Protocol, RESTful, and HTTP are used in the application layer.

d: APPLICATION SOFTWARE

It is a set of programs working in cooperation to meet the user needs. The objectives of the application software is to collect the data from the sensors, preprocess the data, and extract the value. This value may either be used to control the actuatorsor input to any other service.

Vogel and Gkouskos [154] proposed an Open IoT architecture to address the sustainable necessities of IoT. The architecture supports flexibility, extensibility, and customizability. Flexibility allows users to perform various settings and operations with a shorter delay. Robustness, eas-iness, and cost-effectiveness are the properties of flexibility.Extensibility offers the developers to enhance the architec- ture with minimum cost upgrades. Modularity, Adaptabil- ity, and compatibility are the properties of the extensibility. Customizability allows the user to modify the features of the system. Customizability has properties like easiness and cost-effectiveness. Cirani *et al.* [155] proposed a scalable and self-configurable architecture for IoT to implement automated services and resource discovery features. In this architec- ture, the Zeroconf mechanism (No prior configuration infor-mation is needed to discover new services and resources) is used to discover the services and resources within the local network. Service discover protocol is used to find out the services and resources based on the Universal Resource Identifier (URI) and the number of nodes between the client and the server. Hu *et al.* [156] proposed a Software-Defined Device-based IoT System architecture. In this architecture, a centralized controller is used to manage the devices in the IoT network. This architecture aims at providing features like device discovery services, sharing, and reusing of the device resources. To enable these features, the authors introduced the Software Defined Device layer between the transport layer and application layer. Xu and Helal [157] proposed a scalable Cloud-Edge-Beneath based architecture for IoT. This architecture consists of beneath, edge, and cloud layer. The beneath layer comprises of the physical and sensor plat- form layer. The physical layer is a collection of devices and descriptions about themselves. The authors have used XML based device description language to define the

specificationof the devices. The sensor platform in this layer is responsiblefor device configuration, booting, and control. Also, this layer performs data acquisition and caching. The edge layer provides the service discovery and configuration mechanism using Open Services Gateway initiative (OSGi) framework. The cloud layer is liable for the deployment and running of the application-specific services. Sarkar *et al.* [158] proposeda scalable distributed architecture for IoT. This architecture proposed an approach to handle issues like scalability, het- erogeneity, interoperability, and security. The architecture consists of Service Layer (SL), Virtual Object Layer (VOL),and Composite Virtual Object Layer (CVOL). SL is respon-sible for the creation and management of IoT services. The virtualization of the physical objects is carried out in this layer, and this contains the characteristics and capabilitiesof the physical objects. Virtual nodes in the CVOL actsas a coordinator and also performs the smart scheduling of the tasks. A cross layer security management module is defined by the architecture to address privacy and security issues.

COMMUNICATION TECHNOLOGIES FOR IoT

Communication plays a significant role in the sharing of raw and analyzed data between the objects. The role of communication in IoT is to enable the seamless connectivity between the two endpoints anytime, at anyplace [159]. IoT comprises of Edge to Edge, Edge to Gateway, Edge to Cloud,and Backend Data Sharing models [160].

A. COMMUNICATION MODELS

1) EDGE TO EDGE COMMUNICATION MODEL (E2EC)

In this model, devices communicate directly with other devices. This communication model takes advantage of wiredor wireless communication standards. Figure 14 depicts the working of the Edge to Edge Communication model. This model plays a vital role in indoor applications like home automation, industrial automation, and others [160], [161].

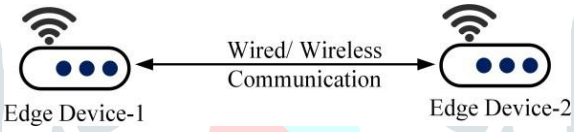


FIGURE 14. E2EC model.

2) EDGE TO GATEWAY COMMUNICATION MODEL (E2GCM)

In this model, the gateways are loaded with the application layer software and serves as an intermediate between the devices and application service providers. Figure 15 illus-trates the working of the Device to Gateway Communication
Model [160]. This mode plays a key role in applications likesmart homes, V2V communications, livestock monitoring, sports and fitness, and others.

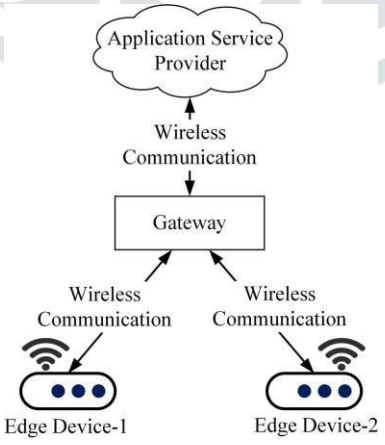


FIGURE 15. E2GC model.

3) EDGE TO CLOUD COMMUNICATION MODEL (E2CC)

In this communication standard, devices are directly con- nected to the cloud. Devices read and write data from and to the cloud, respectively. This communication model is given

in Figure 16 [160]. Applications like Smart Grids, FMS, Connected Labs, etc. use this communication model.

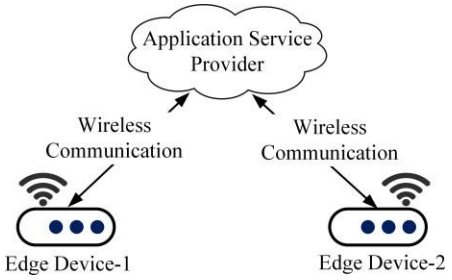


FIGURE 16. E2CC model.

4) BACK END DATA SHARING MODEL (BDS)

This model allows the trusted third parties to obtain the sensory data from the cloud for aggregation and analysis. Theefficient implementation of this model enables the trusted third parties to transfer their data between the multiple IoT services. Figure 17 represents the concept of the Backend Data Sharing model.

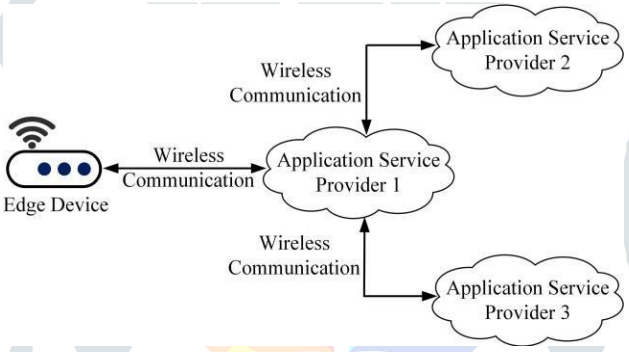


FIGURE 17. BDS model.

Generally, these communication models use communica-tion standards like RFID, WiFi, Bluetooth, ZigBee, Z-wave,LoRa, Sigfox, NB-IoT, 5G, etc.

B. COMMUNICATION STANDARDS

IoT devices can use wired or wireless communication stan- dards. The use of wireless communication standards is increasing day by day in society. Therefore, this section provides the short and long-range wireless communication standards used in IoT. At present, the predominant orga- nizations like IEEE (IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4, IEEE 802.16, etc.) [162], and Internet EngineeringTask Force (IPv6 over Networks of Resource-Constrained Nodes (6lo), Constrained RESTful Environments (CORE), IP Wireless Access in Vehicular Environments (IPWAVE), IPv6 over Low Power Wide Area Networks (IpWAN), Soft-ware Updates for Internet of Things (SUIT), etc.) [163] are providing the communication standards for IoT.

SHORT-RANGE COMMUNICATION STANDARDS

Generally, the perception layer is the collection of sensorsthat aims to collect and transmit the physical parametersto the gateway using short range communication standards. Most of the sensors at the perception layer are battery oper- ated and use short-range, low-power wireless communication standards like RFID, BLE, WiFi, Zigbee, Z-wave, Thread, Light-Fidelity (LiFi),and Wireless HART to transmit the data. The use of these communication standards will increase the lifetime of the sensors.

a: RFID

RFID is a wireless communication technology designed for automatic identification, tracking, and collection of data fromthe objects [164]. RFID is a combination of RFID tags and readers. RFID-tag is a microchip used to store the identifi- cation information; this identification information is used totrack the objects later. RFID-reader fetches the data whenan RFID-tag comes into contact. RFID-reader can be fixed or mobile [165]. The communication range of RFID exists between 3 to 90 meters. This type of communication is bene- ficial in applications like production monitoring and control, supply chain management, etc. [166].

b: WiFi

In WiFi, transmission, and reception of data over a short distance are using 802.11 radio technology. Various classes of WiFi currently exist in the market, such as 802.11a,802.11b, 802.11g, and 802.11n. The WiFi coverage rangeis about 50 meters, and the data rate reaches up to11 Mbps [167]. WiFi is ubiquitous in smart homes, buildings,etc.

c: BLUETOOTH LOW ENERGY (BLE)

It is also known as the Bluetooth Smart, built for short- range communication. Compared to classical Bluetooth tech- nology, BLE consumes less power and provides high data rates for audio and video applications. It uses a 2.4 GHz ISM frequency band and can reach up to 100 meters. BLE provides the data rate up to 1Mbps. Applications like smart homes, smart offices, smart shopping, etc. are benefited fromBLE [168].

d: ZigBee

ZigBee is a communication standard developed by the ZigBee Alliance to carry small chunks of data. ZigBee is of low power and low cost, which consumes 1mW most of the time. The operating range of the ZigBee is up to 150 meters.The bandwidth of the channel is 1 MHz, and its data transfer rate is about 250kbps. Zigbee is very useful in applications like smart homes, industry 4.0, etc. [169].

e: Z-WAVE

This communication standard establishes communicationover a short distance, and Zensys developed it. Z-Wave forms the mesh networks to enable the device to device communica- tion. Z-Wave reduces the latency in transmitting the smallerpackets at the rate of 100kbps. The communication range ofthis protocol is about 30 meters [170].

f: LIGHT-FIDELITY (LiFi)

LiFi is a wireless communication standard, which uses the visible light spectrum (i.e., Light Emitting Diode Bulbs) to transmit the data. Sometimes, this standard is also alluded as Visible Light Communication (VLC). LiFi supports greater mobility and multiuser access. It provides a hundred times faster data rate than WiFi (i.e., 1 Gbps). However, the rangeof communication is very limited. Smart homes and Industry 4.0, Virtual and augmented reality applications get more benefit from this standard [171], [172].

g: WIRELESS HART

Wireless HART is an extended version of wired HART pro- tocol, and it is commonly used in process automation. This standard supports time synchronization, self-organization, and self-healing mesh topology in industrial networks. Wire- less HART operates using a 2.4 GHz ISM band. The data rate of the protocol is about 250 Kbps and covers a radius of 230 meters. This protocol is best suited for the Factory automation/ Industry 4.0 applications [173].

1) LONG-RANGE COMMUNICATION STANDARDS

Long-range communication standards are commonly used by the gateways to share the data across the remote locations. LoRaWAN, NarrowBand-IoT, Sigfox are generally used for long-distance communication, and the use of 5G technology is increasing slowly. These standards provide features like high data rate, less power consumption, and more extensive coverage area (communication range).

a: LoRaWAN

LoRaWAN is a Low Power Wide Area Network (LPWAN)technology designed to connect battery-operated deviceswirelessly [174]. LoRaWAN uses the LoRa spread spec- trum modulation technique at the physical layer to achieves the characteristics of the LPWAN. Chirp Spread Spectrum (As time increases, frequency also increases linearly) mod- ulation technique is used by the LoRa. The maximum admis-sible data rate in LoRaWAN is about 50 kbps with an area coverage of 5 km (urban) to 20 km (rural) [175].

b: NARROW BAND (NB)-IoT

NB-IoT is an LPWAN cellular radio technology developed by the 3GPP to cover a wide area, and it uses the LTE standard [176]. The NB-IoT aims to enable low-power, low-cost, and a wide range of communication in the IoT environment. Per each cell, NB-IoT accommodates up to 50,000 devices, and the data rate is ranging from 230 kbps (Downlink) – 250 Kbps (Uplink) with an average area cov- erage of 35 km [177]. NB-IoT based devices have a better life for up to ten years. This technology is very flexible withthe 2G, 3G, and 4G technologies and eliminates the need forgateway (Directly connected to the BS).

c: NB-Fi

NB-Fi is an LPWAN connectivity solution developedby WAVIoT. This standard uses 433MHz, 868MHz, and 915MHz ISM bands. Single BS can manage up to 2000, 000 smart devices, which fall in its range; this provides the high scalability in NB-Fi networks. Artificial intelligence is used in NB-Fi networks efficiently in order to enable the self-management and optimization network characteristics. The data rate of this standard is about 25 kbps, and it can cover up to 16 km in urban and

50 km in the rural area. NB-Fiplays a vital role in the application like smart agriculture andsmart retail [178], [179].

d: SIGFOX

Sigfox is an LPWAN communication technology used to deploy IoT networks, which operates at 433 MHz in Asia, 868 MHz in Europe, and 915MHz frequency in North America. Sigfox coverage area ranges from 10 km (rural)to 40 km (urban) with a data rate of 100 bps [175]. Sigfox uses the Ultra Narrow Band (UNB) modulation tech- niques [180], which reduces the consumption of energyand increases the receiver's sensitivity [181]. A Sigfox device can send only 140 messages per day with a pay-load length of 12 bytes. A Sigfox device survives up to a decade [182].

e: INGENU

INGENU is a proprietary LPWAN communication standard,and it is also known as the On-Ramp Wireless standard.It uses the 2.4GHz free ISM bands for communication and Random Phase Multiple Access (RPMA) technology; this enables robust and reliable communication. RPMA obeys the specifications of IEEE 802.15.4k. The communication range of INGENU is about 15 km, and the data rate is about20kbps [183].

f: TELENSA

Telensa is an LPWAN standard, which uses an ultra- narrowband transmission mechanism. The PLANet is a central management system used by Telensa to managethe end-to-end operations. PLANet uses an automated fault detection system, which reduces the energy consumption and manages up to 5000 nodes. It operates using 868MHz and 915MHz unlicensed ISM bands. The communication range of this communication standard is of about 2 km in urban and 4 km in rural areas. The estimated lifetime ofa Telensa node is up to 20 years. As a real-time example, Telensa was used to deploy 369,000 street-lights in the city of Georgia. This communication standard is suited for smartcity applications [184].

TABLE 4. Characteristics of communication technologies.

Characteristics	WPAN	WLAN	WMAN	WWAN
Communication Standards	Bluetooth, BLE, Zigbee, Thread Z-Wave, EnOcean, ANT	WiFi, ETSI HiperLAN, Wireless HART	WiMAX, ETSI HiperMAN, Zigbee-NAN, NWave, LoRa, Sigfox, NB-IoT, Wi-SUN	3G/4G/5G, LPWAN, WRAN, NB-IoT, Satellite Communications, Cognitive Radio, INGENU, Telensa
Communication Range	10 – 150 Meters (Short Range)	150-1000 Meters (Short/ Medium Range)	2 – 50 km (Medium Range)	Up to 100 km (Long Range)
Frequency	2.4GHz	2.4 – 5.9 GHz	10 – 66 GHz	900 – 1800 MHz
Maximum Data Rate	1-2 Mbps	11-54 Mbps	Up to 268 Mbps	10 Kbps – 2.4 Mbps
IoT Applications	Home Automation, Smart Agriculture, Healthcare, Structural Health Monitoring	Home Automation, Smart Agriculture, Healthcare	Smart Cities Multimedia, Digital TV Broadcasting	Smart Cities, Intelligent Transportation

g: 5G (FIFTH GENERATION) AND BEYOND WIRELESSCOMMUNICATION

5G is the foundation for realizing the full potential of IoT. It is a fifth-generation cellular network, which provides seamless connectivity between the more significant number of deviceswith high data rates and greater mobility [185]. The standard-ization of 5G technology started in the year 2016 by Interna-tional Telecommunication Union (ITU) and 3GPP [13]. 5G offers high data rates and spectral efficiency by using mil- limeter wave (mmWave) communication technology. In 5G, Massive MIMO enhances the coverage area and user expe- rience. The operating frequency of the 5G technology is between 350 MHz to 100 GHz, and the latency is about 1ms.5G communication technology can reach the data rate up to 20 Gbps and supports different scenarios like indoor, rural, urban, and suburban scenarios, etc. [185], [164].

Table 4 summarizes the characteristics of different com- munication technologies [186].

V. IoT APPLICATION LAYER PROTOCOLS

The application layer is the topmost layer in IoT layered architecture and delivers the application services to the users by initiating actual communication between the IoT devices [187]. This layer is liable for formatting and pre- senting the data to the end-users by using the application layer protocols. MQTT, CoAP, XMPP, DDS, and RESTful/HTTP protocols are the most commonly used in IoT applica- tions [188]. This section provides the detailed analysis of the above listed application layer protocols. OASIS defines the application standards like MQTT, AMQP, and COEL [189]. The oneM2M defines the protocol bindings for CoAP (TS-0008) [190], HTTP (TS-0009) [191] and MQTT (TS-0010) [192], [193].

A. MQTT

MQTT is a lightweight application layer protocol used to transmit messages. Figure 18 depicts the MQTT messaging model. MQTT operates over the transport control (TCP/IP) protocol and uses the Publisher/Subscriber messaging pat- tern [194]. The

publisher, broker, and subscriber are part

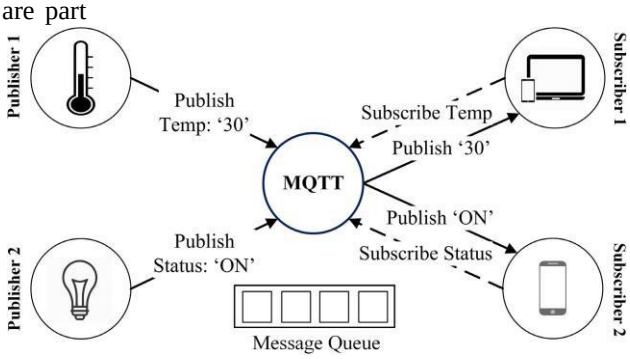


FIGURE 18. MQTT messaging model.

of the MQTT messaging model. The publisher's role is to publish the data to the broker instead of sending messages directly to the subscriber. To receive messages from the bro-ker, the subscriber must register with the broker. The bro- ker administers all the message flow between the publishers and subscribers using the concept of topics. MQTT protocol ensures reliable message delivery in resource-restricted envi- ronments [195].

MQTT offers various QoS [196], [197] to perform better under unstable networking conditions, as listed below.

1. Level 0 (QoS - 0): This makes sure that the messages are sent to the broker almost once, and it does not require any acknowledgment [196], [198].
2. Level 1 (QoS - 1): This makes sure that the messages aresent at least once to the broker, but it could be sent more than once also [196], [198].
3. Level 2 (QoS - 2): This ensures that the messages are sentonly once to the broker. This uses the four-way handshake and which affects the overall performance of the proto- col 196], [198].

The exchange of the data between the MQTT devices is basedon the predefined format depicted in Figure 19 [199]–[201].

1. Control Header: The control header is of 8 bits field,and it is divided into two subfields, such as Packet Type and Flags, respectively. Each subfield is of 4 bits. Table 5 provides the type of packets used by MQTT.
2. **TABLE 5.** Packet types in MQTT.

Packet Type	Value	Flow Direction	Description
RESERVED	0	Forbidden	-
CONNECT	1	Publisher to Broker	Publisher request to connect to the broker
CONNACK	2	Broker to Publisher	Acknowledgment to Publisher from Broker
PUBLISH	3	Publisher to Broker	Publishing the message by Publisher
SUBSCRIBE	8	Subscriber to Broker	Subscriber request to connect to the broker
SUBACK	9	Broker to Subscriber	Acknowledgment to Subscriber from Broker
UNSUBSCRIBE	10	Subscriber to Broker	Unsubscribe request to the broker from the Subscriber
UNSUBACK	11	Broker to Subscriber	Unsubscribe acknowledgment from Broker to the Subscriber

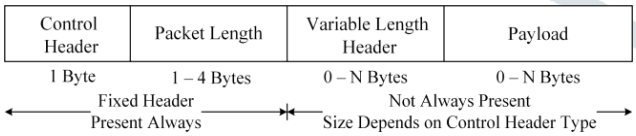


FIGURE 19. MQTT packet structure.

Version 2 Bits	Type 2 Bits	Token Length 4 Bits	Code 8 Bits	Message Id 8 Bits
Token (8 Bits)				
Options (If Any)				
Pay Load (If Any)				

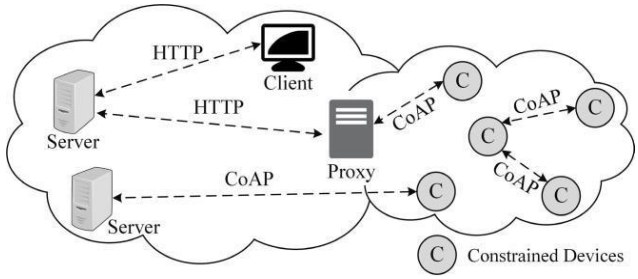


FIGURE 20. CoAP messaging model.

MQTT comprises of Duplicate (DUP), QoS, and Retain flags [91], [199].

3. Packet Length: Signifies the number of bytes left in the existing message, including data in the variable length header and payload [90] [91].
4. Variable Length Header: This field is an optional MQTT message. It is used to carry additional control informa- tion [91], [199].
5. Payload: Payload is the last field of MQTT, which contains the application data [91], [199].

Analysis:

- It is a lightweight protocol suitable for the constrainednetwork of devices.
- It provides the flexibility to choose the level of QoS. It provides excellent services, even if the internet con- nection is
- unreliable.
- The security of the data can be enhanced using dataencryption algorithms.
- Power consumption is more due to the TCP based con-nection.
- Limited support for transmitting high-resolution images,audio, and videos.

B. CoAP

CoAP is an application layer protocol for power and mem- ory restricted IoT devices. CoAP implemented on the top

FIGURE 21. CoAP messaging format.

of the User Datagram Protocol (UDP) [202]. The architec- ture of this protocol is illustrated in Figure 20. CoAP uses the Client-Server communication architecture model. The CoAP uses the RESTful APIs, which reduces communication overhead [202]. It uses HTTP methods like - GET, POST, PUT, and DELETE to empower resource-oriented services in Client-Server applications [203].

Figure 21 shows the CoAP messaging format [204], [205]. The Version field is of about 2 bits, which indicates the version number of the CoAP message. CoAP uses the mes- sage types such as Confirmable (0), Non-Confirmable (1), Acknowledgment (2) and Reset (3) [206].

1. Confirmable (CON): Sender will get the notification fromthe receiver/ server regarding the reception of the mes- sages.
2. Non-Confirmable (NON): Sender will not get any notifi-cation from the receiver/server.
3. Acknowledgement (ACK): This is a replay message to theCON message.
4. Reset: Indicates the confirmable message was received with some missing context. This condition usually occurswhen there is a rebooting and no idea on about state of thereceiving node.

The code represents the message type of the request and response messages. CoAP uses the GET (1), POST (2), PUT (3), and DELETE (4) message types to retrieve, update,create and delete the messages respectively [204]. The mes-sage ID is used to indicate the type of message generatedby the sender of the CoAP node. The token field is usedto differentiate concurrent requests. The token generated by the CoAP node should be unique for the current source and destination pair.

Analysis:

- The communication cycle is faster due to its smallerpacket size. It has lower latency.
- The use of data encryption results in better security ofthe data during transmission.
- Since it uses UDP, there is no guarantee in the messagedelivery.

C. XMPP

The XMPP protocol uses Decentralized Client-Server (DCS)for instant messaging. DCS assists in separating the client andserver developers. The client developer's job is to increase the user experience, and the server developer's job is to provide scalability & reliability [207]. XMPP supports both publisher/subscriber (asynchronous) and request/response (synchronous) messaging patterns. The usage of the mes- saging pattern is up to the IoT application developer. The client, server, and gateway are the components of this architecture [208].

Clients: Clients are the one who initiates the data stream and communicate with the servers.

Servers: It enables the implementation of security fea- tures like authentication and encryption algorithms to provide secure application services.

Authorized client and server connections are also managed by the servers. XMPP uses the XML sections to create a communication link between the client and the server. An XML stream is split into three subsections, such as message, presence, and information/query (iq). Figure 22 depicts the XMPP messaging model [209] and Figure 23 highlights the XML sections [210], [211] used by the XMPP protocol respectively.

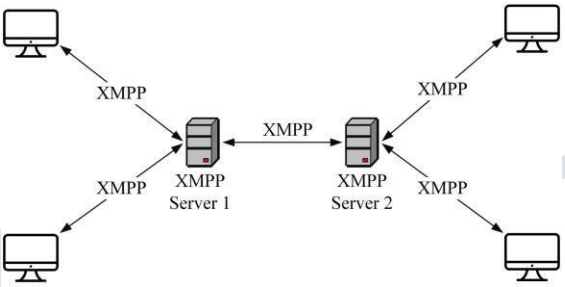


FIGURE 22. XMPP messaging model.

The presence section in XMPP is used to provide client, server, and users status. </show>, </status> and </priority> are the optional elements of the presence section.Non-human-readable characters are part of </show> ele- ment. This section contains information about the avail- able resources. Only one </show> element allowed inside the presence section. </status> element is used along with the </show> element, which includes XML characters.The </show> element provides a detailed definition of the state. The </priority> element contains the non-human- readable XML characters, which indicates the level of resources priority. The priority value must be between −128 to +127. If no priority value is provided, then by default, the server considered the value as zero [95]–[97].

```
<stream>
<presence>
        <show/>
</presence>
<message to = 'x'>
        <body/>
</message>
<iq to = 'y'>
        <query/>
</iq>
</stream>
```

FIGURE 23. XMPP message stanza.

A message section comprised of the title of the message and its content. This section composed of any one of the elements such as </subject>, </body> and </thread>. The </subject> element contains the human-readable XML char-acters, it specifies the message topic. The </body> elementcontains the content of the actual message to be carried.

</thread> element is used as an identifier, which contains the non-human-readable XML characters to track the conver-sation between the entities. Iq section provides the request and response mechanism to communicate [95]–[97].

Analysis:

- Best suited for carrying text messages (Chatting appli-cation).
- The protocol is Decentralized, Extensible, and adaptablenature.
- Excess traffic can be reduced by eliminating the redun-dant data
- Power consumption is more due to the use of XMLmessaging formats.
- Enabling QoS increases the performance of the protocol.

D. DDS

DDS was designed and implemented by the Object Management Group, and it utilizes the publisher/subscriber messaging model [170], [212]. Publishers, Data Writers, Subscribers, Data Readers, and Topics are the components of DDS architecture,

and it is depicted in Figure 24.

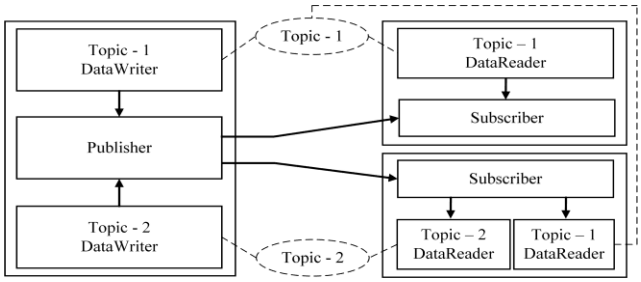


FIGURE 24. DDS messaging model.

The job of the Publisher is to distribute the data. The function of the subscriber is to receive data from publishers

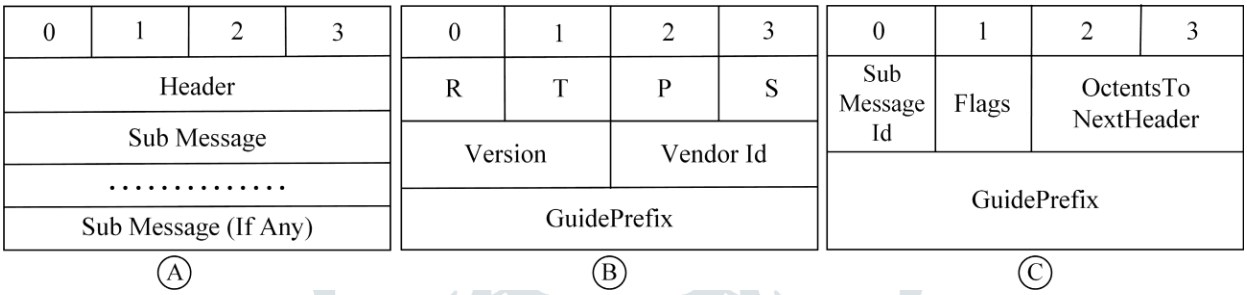


FIGURE 25. Message format of DDS (A. RTPS message format, B. RTPS header format & C. RTPS sub message format).

and makes it available for the application. Publishers and subscribers are attached to the data writers and data readers, respectively. The Data Writer and Reader are attached to theTopic. Since DDS supports QoS and reliability, this protocolis the right choice for M2M applications.

DDS is used by many applications like Rail Network man-agement, Volkswagen Smart Cars for driver assistance, Air Traffic Management System, and many more. The DDS relies on the implementation of the Real-Time Publisher Subscriber Protocol (RTPS) because of its unique features like QoS, plug, and play connectivity, extensibility, and fault tolerance.RTPS is designed to be compatible with both TCP and UDPtransport layer protocol suite [213].

The RTPS message format is shown in Figure 25(A), and it consists of a leading header and a variable number of sub-messages. Figure 25(B) realizes the RTPS message header format, and its length is about 16 bytes with five core sections.Firstly, the protocol field is 4 bytes long; this value must be set to RTPS; The second field is 2 bytes version field, which is used to indicate the RTPS version; The third fieldis two bytes long VendorId field. Each RTPS implementationmust obtain from OMG VendorId; The last field, GuidePre-fix, has a length of eight bytes. The Publisher, Subscriber, DataReader, and DataWriter are GuidePrefix entities with their global identifiers, and these entities are used to guide the source and destination [213].

The sub-message header and payload are part of the sub- message, and it is depicted in Figure 25(C). The Submes- sageId, Flags, and OctecsToNextHeader fields are part of the sub-message. The SubmessageId uniquely identifies the type of the sub-message. The sub-messages use flags. Payload length is indicated by the OctecsToNextHeader field [213].

The sub-message uses 12 message types [213]

1. Data Message: It comprises of an application data objectvalue
2. DataFrag Message: It is used to address the oversized data problem. Here, the data is transmitted in the form of multiple fragments.
3. HeartBeat Message: Checks the readiness of the Readerby the Writer.
4. HeartBeatFrag Message: Indicates the available fragments.
5. Gap Message: Indicates the relevance of the informationto the reader by the writer.

6. AckNack Message: Received by the writer from sender to provide information on data delivery.
7. NackFrag Message: Received by the writer from senderto indicate the missing of the fragment.
8. Info-SRC Message: Used to change the source of the sub-message.
9. Info-DST Message: Used to change the destination of thesub-message.
10. InfoReply Message: It indicates where to send the reply to this sub-message and the sub-messages following this sub-message.
11. InfoTimeStamp Message: Indicates the TimeStamp of the Source.
12. PAD Message: Used for sub-message Padding.

Analysis:

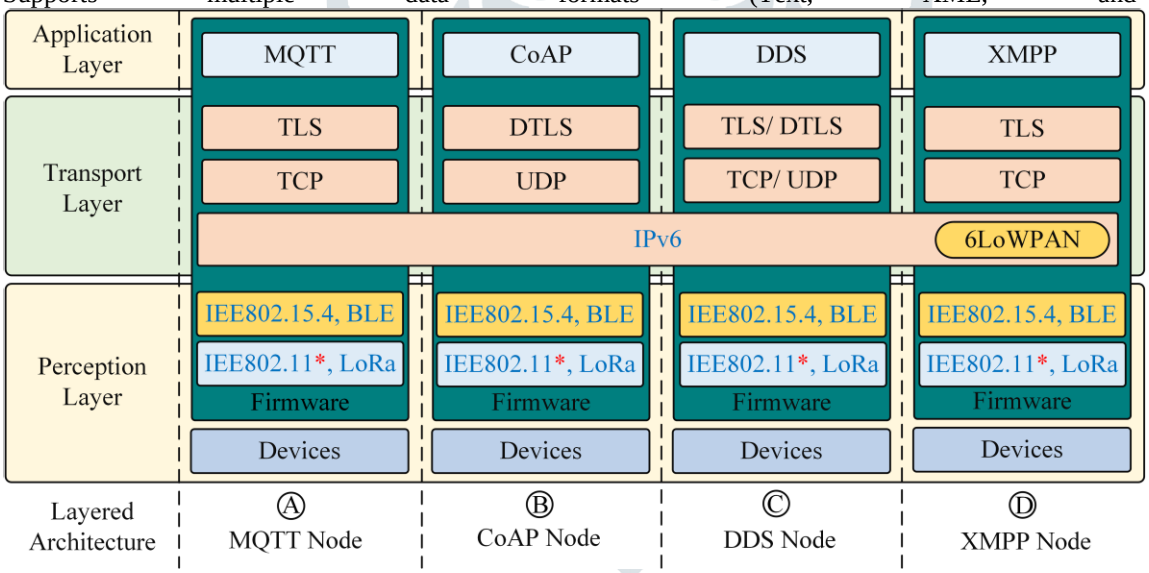
- Point to Point latency is very low compared to otherapplication layers protocols.
- It provides a rich set of QoS options.
- DDS security provides advanced features like authenti-cation and encryption.
- Multicasting is used to enable scalability of the data.It is not suitable for constrained devices.

E. RESTful APIs/HTTP

RESTful API is also termed as a RESTful web service or REST APIs, which are adhered to the constraints of the REST architecture. RESTful APIs use HTTP methods like GET to retrieve, PUT to change or update, POST to create, and DELETE to remove the resources. These methods are defined in RFC2616 protocol. A typical resource is nothing but an image, audio, video, text, shopping order, etc., which are represented using Text, XML, and JSON formats. RESTful API is based on the client-server operation, which is state- less and use URI to identify the resources uniquely. Both clients and servers are independent of each other. RESTful APIs allow the developer to bring multiple services to IoT applications swiftly and make IoT applications distributed and independent. RESTful APIs receive the data from both client and server; this data is used to create new data. The created data is updated among clients and servers; finally, the data is deleted.

Analysis:

- Use of fewer resources
- Supports multiple data formats (Text, XML, and JSON)



NOTE: ** 6LoWPAN can be adapted to work with IEEE 802.11 also (WiFi)*

FIGURE 26. Working models different of IoT nodes.

TABLE 6. Feature analysis of application layer protocols.

Parameters	CoAP	DDS	MQTT	XMPP	HTTP/ HTTPS
Standard	IETF RFC 7252	OMG (Object Management Group)	OASIS Standard	IETF RFC 6120, 6121	RFC2616
Transport Layer Protocol	UDP	TCP/ UDP	TCP	TCP	TCP
Messaging Pattern	Req/Resp	Pub/Sub	Pub/Sub	Pub/Sub & Req/Resp	Req/Resp
Processing Type	Decentralized	Decentralized	Centralized	Decentralized	Decentralized
Security	DTLS	TLS, DTLS & DDS Security	TLS	TLS	TLS
QoS	Yes (2 Levels)	Yes (23 Levels)	Yes (3 Levels)	No	No
Topology	Tree	Bus	Tree	Star	-
RESTful Support	Yes	No	No	No	No
Data Encoding	Defined	Defined	Undefine	Defined	Defined
Resource Locator	URI	Topic Name	Topic Name	Jabber Identity	URL
Default Port Number	5683	Dynamic	1883	5222	8080/ 443
Header Size	4 Bytes	16 Bytes	2 Bytes	-	-
Multicast	Yes	Yes	No	Yes	Yes
Libraries (Python)	CoAPthon, txThings	pyDDS, rticonnextdds	pahoMQTT	SleekXMPP	Requests
Purpose	Applications Based on Constrained Devices	Industrial Automation	Lightweight M2M Communications	Instant Messaging (Chat Applications)	Web Browsing

- Not suitable for handling the massive amount of dataNeed to be incorporate security features
- Offers greater flexibility and Scalability

Figure 26 represents the deployment of the different IoTnodes based on the application layer protocol used (i.e. A. MQTT Node, B. CoAP Node, B. DDS Node, and D. XMPP Node). The devices and firmware and communi-cation standards may varies based on the IoT application developer's interest. However, Zigbee, BLE, and WiFi are the most commonly used for short range communication standards. Similarly, for long-range communication LoRa isused in IoT. IPv6 is the developer's choice for identification in IoT as the number of devices are increasing in the world. The communication standards like Zigbee and BLE use 6LoWPAN, which enables the transmission of data with minimum processing capabilities. WiFi standard has the option of adapting the 6LoWPAN. However, most of the WiFi-based IoT networks do not use 6LowPAN. All thetime, the IoT application layer protocols use either TCP or UDP as the transport layer protocols. TCP and UDP use theTLS and DTLS protocols for security purposes, respectively.Figure 27 depicts the layerwise protocol stack of IoT, and Table 6 provides a feature analysis of IoT application layer protocols.

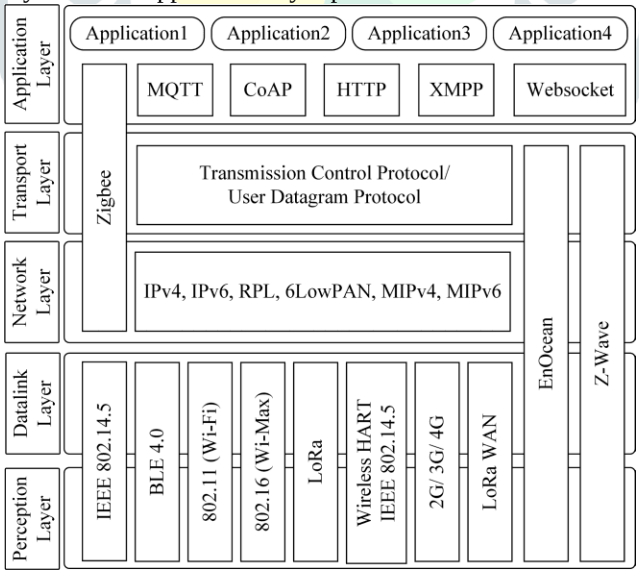


FIGURE 27. Protocol stack of IoT.

Countable IoT application layer protocols were introduced, but still, HTTP and MQTT protocols are the developer's choice to implement IoT applications. IoT appli- cation layer protocols like CoAP, AMQP, and XMPP pro- tocols are infrequent in real-time IoT deployments [214]. Sultana and Wahid [215] provided a detailed analysis of MQTT, AMQP, DDS, HTTP, CoAP, and DDS protocols using IoVT framework. In this framework, cameras were used to capture real-time data. The analysis was performed based on latency, throughput, power consumption, memory, and CPU utilization. MQTT and AMQP have shown better results in terms of latency, throughput, and power consumption. MQTT and HTTP showed less memory usage, and XMPP has shown higher memory utilization. AMQP exhibits higher CPU utilization and XMPP with the lowest CPU utilization. Raspberry Pi based health monitoring system was used by Chen and Kunz [216] to analyses the performance of the MQTT, CoAP, and DDS. MQTT and DDS protocols have shown the zero-packet loss. DDS has shown better perfor- mance compare to CoAP in constrained networks. However, the bandwidth consumption of DDS was much high compare to MQTT and CoAP. Dizdarević *et al.* [217] provided the analysis of IoT application layer protocols (HTTP, MQTT, CoAP, AMQP, DDS, and XMPP) in fog and cloud-based IoT systems. The analysis was based on the characteristics of the protocols and performance metrics (Latency, Power Con- sumption, and Throughput). On the basis of tests conducted by the authors, the MQTT protocol has shown improved per- formance compared to other protocols. Fog and cloud-based IoT systems still use the HTTP protocol. The wide adoption of HTTP is due to the lack of maturity and stability in IoT application layer protocols. Raza *et al.* [218] summarized the features of DTLS and presented a scheme to compress the DTLS headers. The aim of this compression scheme is to reduce energy consumption without compromising end to end security. DTLS header compression is performed using the 6LoWPAN-NHC compression mechanism, and the overall performance of the proposed scheme was analyzed using Contiki OS. Malina *et al.* [219] proposed the security framework for MQTT, which consists of three distinct levels (Level 1, Level 2, and Level 3). Level 1 is used to protect the messages against the tampering and modification attacks. This level is not responsible for message confidentiality. Pub- lishers and subscriber's privacy were protected using Level 2; this layer is also resistant against tampering and modification attacks. Confidentiality and partial anonymity security fea- tures were provided by this level. At this, the authentication is done using the Schnorr digital signatures, and the data is one- time encrypted using the Rabin cryptosystem. Hence, this level is useful to share the messages, which contains personal information. Level 3 enables mutual authentication (using Schnorr digital signatures) between the participants i.e. Pub-lishers, Broker and Subscribers. This protects the data against the tampering, modification and eavesdropping. This enables to create long-term secure communication sessions between the participants. In this, the confidentiality of the data is achieved by AES algorithm. Wang *et al.* [220] proposed lightweight XMPP publisher/ subscriber scheme for IoT devices. This scheme, allows the event driven data exchange between the constrained IoT devices and it is compatible with both TCP and UDP. Based on the interest, subscribers send the service request to the servers. Publishers reacts to the service requests using the subscription rules. Sleep and wake-up modes are used by the proposed scheme to enhance the life of the battery. Publishers provide the requested services to subscribers, when they are in wake-up mode. For this cause, publishers share the sleep and wake-up messages with the server. These messages are used by the server to manage the subscription between the publisher and subscriber effectively.

VI. COMPUTING PARADIGMS

In traditional data models, data was generated by a limited number of companies, and others were consuming the data. However, the days have been changed; everyone is generating and using the data. IoT also contributes significantly to create a large volume of data. Due to the lack of resources and computing facilities, handling of the enormous volume of data in IoT is a tedious task. Computing paradigms like cloud computing, cloudlets, fog computing, and edge computing came into the picture to facilitate the IoT data cycle (Storage, Pre-processing, and Analysis) [221].

A. EDGE COMPUTING

Edge computing is a networking architecture used to per- form certain computations (Analysis), where data is cap- tured [222]. Figure 28 depicts the concept of the General Edge Computing Architecture (GECA). Proximity, Intelligence, and Control are considered as the major components of edge computing [223].

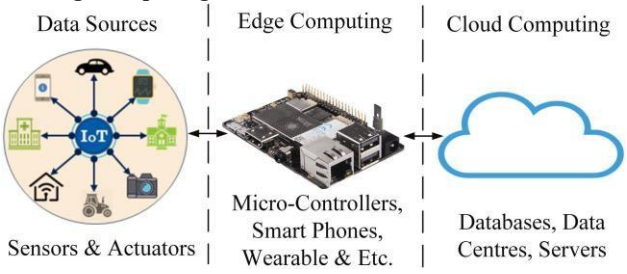


FIGURE 28. Edge computing architecture.

1) CHARACTERISTICS OF EDGE COMPUTING

a. Latency: Centralized cloud computing is not appropriate for all applications, as some applications require imme- diate solutions. Latency is more due to the sharing of resources and distance between objects; the cloud server must serve

multiple objects at a time. The latency prob- lem can be reduced by placing the edge nodes between the objects and the cloud server [224]. Researches have constructed a proof-of-concept platform for running face recognition applications by shifting the computation from cloud to edge and which reduces the response timefrom 900 to 169 milliseconds [225].

b. Security: The objects have insufficient memory to implement conventional security algorithms in IoT, andcloud platforms are the target of hackers most of the time. Once the edge nodes have become the part of IoT networks, data generated by these objects can be secured by implementing the novel lightweight securityalgorithms in it [226].

c. Real-Time Data Analysis: Connected objects areincreasing in number, which results in the generation ofexcessive volume of data. Understanding and extractingthe useful patterns from the excessive amount of hetero-geneous data generated by the objects effectively is a tedious task. The edge computing paradigm enables theprocessing and analysis of data in real-time at the edge of the IoT network.

d. Mobility: Mobile devices are increasing rapidly in day to day life. Mobility of the edge nodes enables the distributed computing in the network. Locator Identity Separation Protocol (LISP) is used by edge computing devices to communicate with mobile devices.

2) EDGE COMPUTING ARCHITECTURES

Currently, there are no architectural standards in edge com- puting, but there are industries and research institutes like Open Edge Computing Community, Open Fog Consortium, and European Telecommunication Standardization group working on edge computing architectures. The three Edge Computing Architectures (A). Edge Server Architectures (B). Edge and Coordinator Device Architecture, and (C). Device Cloud Architecture is shown in Figure 29 [227], and Table 7 gives the differences between the different edge computing architecture.

TABLE 7. Comparison of edge architectures.

Parameters	Edge Architectures		
	ESA	**ECDA**	**DCA**
Bandwidth	High	Moderate	Low
Computational Capacity	High	Moderate	Low
Mobility of the Nodes	Fixed Position	Mobile	Fixed and Mobile
Data Storage	Less Storage	High Storage	No Storage

a. Edge Server Architecture (ESA): This architecture is generic, where all the objects are connected to the Edge Server (ES), and ES is connected to the rest of the network, including the cloud. In this architecture, ES is static with high computational power. This architecture gives a clear difference between the device level, edge level, and the other networks. Nano Data Centers are examples of edgeservers [227].

b. Edge and Coordinator Device Architecture (ECDA): In this, the communication between the objects and the edgeservers is via coordinator objects. The coordinator objects may be smartphones, tablets, or laptops. These objects have more power, bandwidth, and computation facilities compared to edge servers. The coordinator devices can communicate with the other networks via edge servers,or they can communicate directly without the help of edgeservers [227].

c. Device Cloud Architecture (DCA): Both device level and edge level are merged in this architecture. If neces- sary, the devices communicate with the cloud via edge devices. The researcher considers this type of architecture as opportunistic computing or transient clouds (Coopera-tive computing platforms, which enables nearby devices toform a network and offers various services) [227], [228].

Edge computing is seeking attention among the academi- cians and researchers. Most edge computing frameworks usededicated physical devices to coordinate IoT devices. Edge devices in IoT provide temporary storage and data analysis features. Query processing, activity recognition, malicious activity, and malware detection, augmented reality, surveil- lance, and resource management applications are taking the advantages of edge computing technologies. Uddin [229] developed a wearable sensor-based system to predict human activities using Recurrent Neural Networks (RNN) over an edge device. In this work, MHEALTH dataset was used, which comprises of magnetometer, electrocardiogra- phy, accelerometer, and gyroscope sensors data. All the data from the sensor are recorded at a sampling rate of 50 Hz. The system predicts the activities like standing, relaxing, lying down, walking, climbing, jogging, cycling, running, jump- ing, and bending. The authors claimed that they achieveda 99.69% mean prediction rate. A laptop with a Graphical Processing Unit (GPU) was used as an edge device to performthe activity predictions. Zhao *et al.* [230] proposed a data

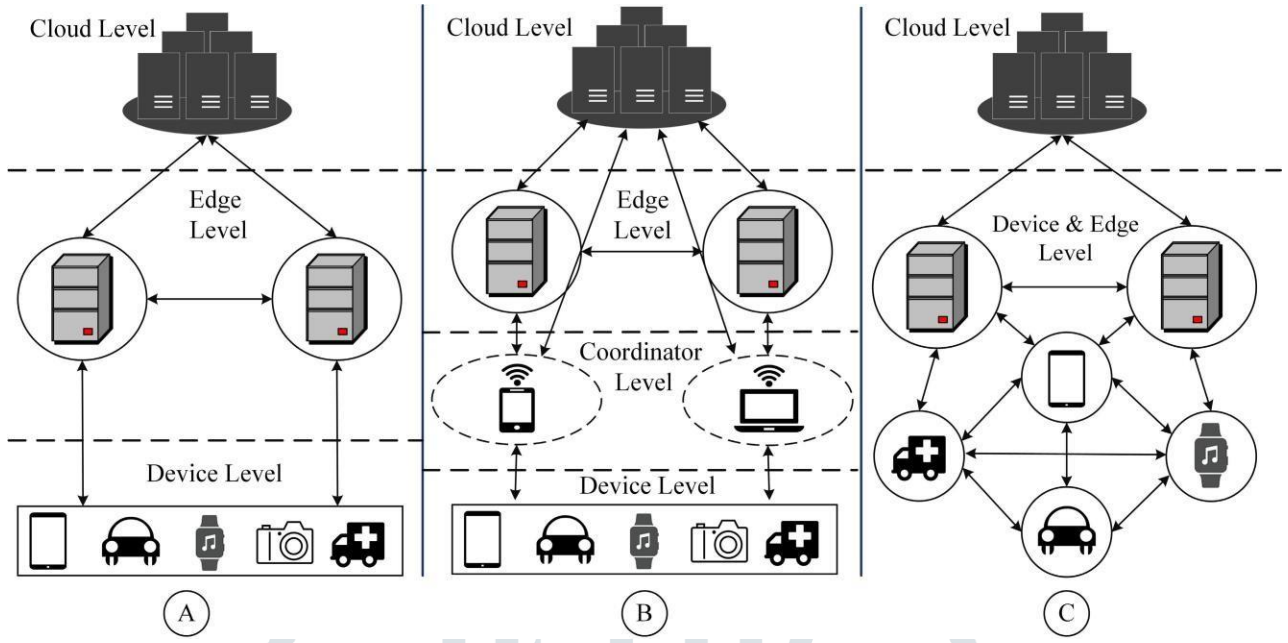


FIGURE 29. Edge architectures (A. Edge server architecture, B. Edge coordinator device architecture & C. Device cloud architecture).

offloading in mobile edge node based on the deep learning technique. A multi-Long Short-Term Memory (LSTM) deeplearning model was used to predict the real-time traffic, whichhelps to perform the data offloading process accurately. Based on the prediction result, the cross-entropy method was used toperform the data offloading, and it is based on the offloadingindicator. In this, the JINHUA dataset is used to approxi- mate real-time traffic. Yuan and Li *et al.* [143] proposed a lightweight and reliable trust mechanism to increase the QoSand mitigate bad-mouthing attacks triggered by malicious feedback providers. This mechanism is based on the calcu- lation of the trust using multi-source feedback (Multi-Source Feedback - feedback gathered from both devices and edge devices) fusion. The feedback trust is divided into 1. Deviceto Device Trust – In this, device trust is formed using the communication history of the devices, 2. Device to Broker Trust – In this, the broker computes the real-time trust of each device after completing the computation task and shares with the other devices, and 3. Overall Trust – is based on the fusion calculation using the trust of multiple devices. The authors have introduced the broker layer (Layer consists of edge devices) between the device and network layer, andit is responsible for monitoring the activities of the devices and feedback trust calculations using objective information entropy theory.

B. FOG COMPUTING

Fog Computing or Fogging is a computing architecture,which was coined by CISCO in 2014 [231]. This comput- ing paradigm minimizes the load on the cloud by provid-ing the computing, communication, storing, and networking facilities [232]. The fog Computing layer acts like a micro data center for IoT applications [233]. Typically, Fog Layerconsists of Servers, Routers, Switches, and Access Points. The objects are connected to the fog nodes to acquire the services provided by them. This computing paradigm plays a key role in the applications where immediate resources and responses are needed. IoT Applications like Disaster Management Systems, Structural Health monitoring, smart health, etc. comprise of critical data, which require immediateprocessing. In IoT, the fog nodes perform the data aggrega- tion, removal of redundant data, filtering and data analysis tasks swiftly [234]. Figure 30 illustrate the architecture of FogComputing.

Rathee *et al.* [235] proposed a security mechanism to mitigate various attacks based on the trust values in fog based IoT networks. A trust manager is used to keep track of all fogand IoT nodes utilizing a look-up table. In this mechanism, the trust values are calculated using Social Impact Theory Optimizer, and values vary between 0 and 1. These values are assigned to the neighbors using the previous interaction history. The trust manager maintains a copy of the fog node's identity, address, trust values, and ratings. Tidal trust algo- rithm is used to calculate the ratings of the nodes and establishthe trust between the fog nodes and users. The fog node is said to be malicious when the trust value of the node is less than the assumed threshold value. Yoon *et al.* [119] proposeda sensor data management method using a fog node, where the fog node acts as a monitoring node, and the authors also discussed the framework of the monitoring node. The mon- itoring node analyzes the sensors incoming data by means of data patterns (Varying of data values of a sensor over a time). The monitoring node reports to the Operational and Management Server (OMS) only when the analyzed data is

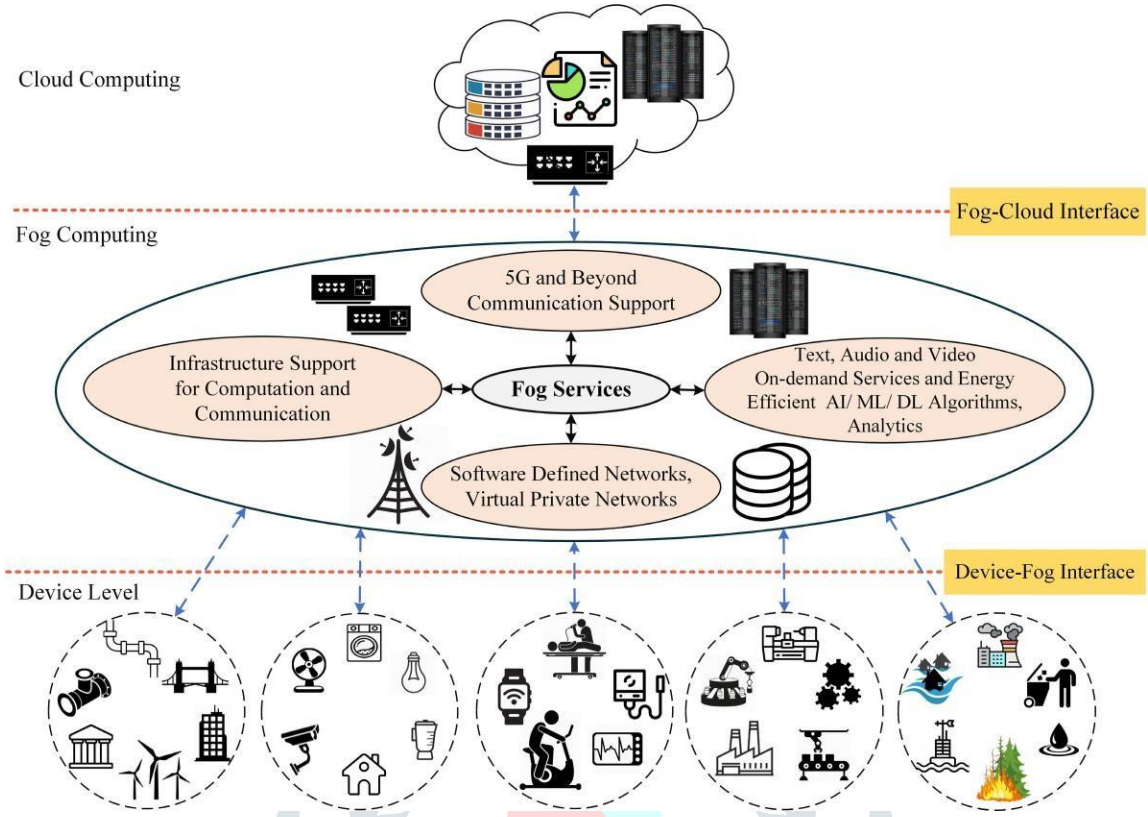


FIGURE 30. Fog computing architecture.

beyond the normal range. The fog node is capable of con- trolling the sensor devices based on the instructions from the OMS. Tuli *et al.* [236] proposed architecture to diagnose heart diseases using the features of IoT and Fog computing. The authors developed the deep learning-based HealthFog system to diagnose or predict heart diseases in the patients, and the system is deployed using the FogBus framework to analyze the performance of the system, such as accuracy, response time, and energy consumption. The proposed method used the Ensemble of deep neural network models for predicting the heart diseases on the Cleveland dataset.

C. CLOUDLETS

A cloudlet is a trusted data center, which swiftly provides cloud computing services to devices like smartphones, wearable, autonomous vehicles, and so on. Typically, cloudlets are situated at the base sta- tions [238]. Cloudlets aims at reducing the end to end latency between the cloud and mobile devices. Also, cloudlets main- tain the privacy of the user's data before offloading them to the cloud [239].

The authors in [240] have summarized the problems asso- ciated with the use of Unmanned Aerial Vehicles (UAVs) to provide the latency-sensitive services using IoT. In this, UAVs are used as cloudlets, which offers the computation offloading ability to resource constraint IoT devices. In this work, servers are mounted on the UAVs. These UAVs fly near to the IoT devices and process the data generated by them. Lai *et al.* [241] proposed a Fairness-oriented computation offloading scheme FairEdge, which enables balanced task distribution in the IoT and cloudlet networks. The FairEdge uses the Balls-and-Bins load balancing theory to offload the IoT tasks into cloudlets. The fairness index of the scheme is calculated using the Jain's fairness index, and it is a part of the task offloading scheme. The authors mentioned that the proposed FairEdge scheme reduces the unbalanced task offloading by 50%. In [242], IoT enabled cloudlet assisted e-health framework is proposed. The aim of this framework is to make it easier to access the real-time data using cloudlets. This framework comprises of IoT, cloudlet, and cloud layers. IoT layer is a collection of healthcare devices to monitor the status of the health in real-time, and these devices are communicating with IoTHub (Gateway) using short-range wireless communication standards. Cloudlets are introduced between the IoTHub and cloud. Usually, data processing tasks were carried out in the cloudlet layer, and cloudlets belong to different geo-locations share the healthcare data among themselves. The cloud layer performs the various queries and analytics operations on the healthcare data.

D. CLOUD COMPUTING

It is a computing standard [243], which enables the users to access a set of services offered by the service providers

TABLE 8. Features of IoT cloud platforms.

Characteristics	Application Layer Protocols	Sensor Data Visualization	Security	Language Support
Amazon IoT Cloud Platform (https://aws.amazon.com/iot/)	MQTT, WebSocket, HTTP	Yes	TLS	C, Node.js, Python, Java
Azure IoT Hub (https://azure.microsoft.com/en-in/services/iot-hub/)	MQTT, AMQP, HTTPS, AMQP	Yes	X.509	C, C++, Python, Java, Node.js, C#
OpenIoT (http://www.openiot.eu/)	MQTT, CoAP, AMQP, XMPP	Yes	TLS	Java
Thing Speak (https://thingspeak.com/)	MQTT, REST API's	Yes	SSL/ TLS	C, Python, MATLAB
IBM Watson IoT Platform (https://www.ibm.com/in-en/internet-of-things)	MQTT, HTTP	Yes	-	Python, Java, Node.js, C#, Embedded C
ThingWorx (https://developer.thingworx.com/platform)	MQTT, WebSocket, HTTP	Yes	-	C, Java, Python
Xively (https://xively.com/)	MQTT, WebSocket, HTTP	Yes	TLS	C, C++, Python, Java
Zetta (https://www.zettajs.org/)	MQTT, CoAP, WebSocket, HTTP	Yes	Yes	Node.js
Temboo (https://temboo.com/iot)	HTTP, MQTT, CoAP			C, Java, Python, Ruby, JavaScript

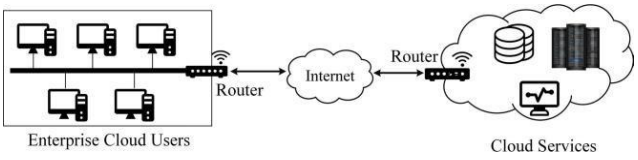


FIGURE 31. Cloud computing architecture.

over the internet. Cloud computing architecture is depicted in Figure 31. Infrastructure, Data storage, resource sharing, and software access are the services offered by service providers [244]. Cloud computing consists of following enti- ties [245]: 1. Cloud Provider – delivers the set of services to the interested parties, 2. Cloud Consumer – utilize the ser- vices offered by the

cloud provider, 3. Cloud Broker – acts as an intermediate between the cloud provider and consumers to manage the service delivery, 4. Cloud Carrier – intermediate that provisions the connectivity and delivery of cloud services and 5. Cloud Auditor – evaluates the cloud infrastructure, services, system operations, performance, and security of the cloud implementations. Cloud Computing helps small enter-prises to create and deliver Information Technology solutions as early as possible to the end-users [246]. E-Governance, Smart Healthcare, Intelligent Transportation Systems, Life Sciences, Smart Grid, Data Analytics, Disaster Management System are some of the applications, which make use of cloud services for different purposes. These applications produce a huge volume, velocity, and variety of data. Cloud Comput- ing is responsible for storing and analyzing the data more swiftly [221].

Private, Public, and Hybrid Cloud are the three categories of the cloud [221], [247]. Always, the private cloud access is provided to the selected people of the organization. Irrespec-tive of affiliation and origin, all the users can pay and access the services provided by the public cloud. The Hybrid cloud possesses the features of both private and public cloud [221]. Data analysis, data management, device management, scal- ability, and security features should be considered while designing and developing cloud-based IoT solutions. Table 8 illustrates the features of the different Cloud platforms for IoT.

Lee *et al.* [248] presented hierarchical cloud computing architecture to manage context-aware services in IoT. The architecture comprises of Cloud Control and User Control layers. The cloud control layer is responsible for resource allocation and scheduling. Similarly, the user control layer is responsible for end to end connection between the devices and the cloud. The proposed architecture consists of 1. Non- uniform Service Binding Model – this model delivers an application-oriented computing environment at the platform level, 2. Real-time Adaptable Service Binding Model – this model comprises of application and transmission bindings, and this uses the meta object-based reflective system and

3. Intelligent Service Management – this model provides intelligent service management using the supervised and reinforcement ML algorithms. Chen *et al.* [249] proposed a service management protocol called IoT-HiTrust for bulk mobile cloud systems. The proposed protocol uses trusted nodes to increase the performance of the applications with respect to security, scalability, and accuracy. The proposed mechanism was based on public relations among the owners of the IoT devices. The trust node's calculations are based on the recommendations received from the IoT devices. The recommendation filtering algorithm applied on these rec- ommendations, which possess the features like familiarity (list of other friends IoT devices), community contact (rep- resents the closeness), and community attention (represents the knowledge on the subject matter). Smart city travel and air pollution detection applications were used to analyze the

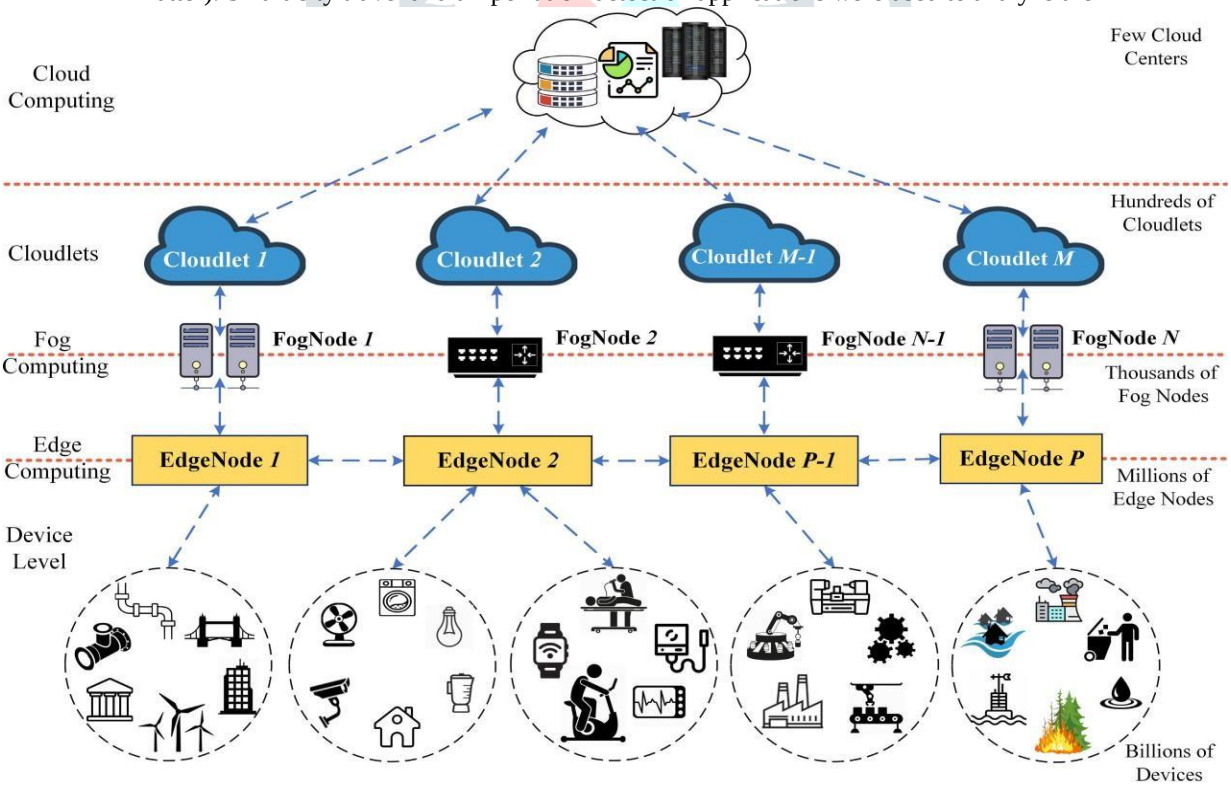


FIGURE 32. IoT computing hierarchy.

feasibility and performance analysis of the proposed proto-col. The authors demonstrated the simulation of the protocol using the NS-3 tool. The proposed mechanism withstands against the attacks like Self-promoting, Bad-mouthing, and Ballot-stuffing attacks. Lee *et al.* [248] proposed a Cloud- Edge-Beneath architecture to enhance the scalability features in IoT networks. The architecture is made up of Beneath, Edge, and Cloud layer. The beneath layer is further comprised of physical sensors and sensor platform layer. This layer is accountable for gathering the raw data using sensors and establishing the communication between sensors and edge nodes using the sensor platform layer (Low power computing and communication

devices). The edge layer is a collection of edge devices used to manage the group of sensors. The formulation of the sensor group is based on their functional-ity, type, and location. The cloud layer is a collection of ser- vices accessed by the devices in the beneath and edge layer. This architecture implemented using Service Oriented Device Architecture (SODA). SODA automates the sensor/ device integration using the Atlas Sensor Platform and middleware. Figure 32 shows the levels of IoT computing architec- ture, which included device, edge, fog, cloudlet, and cloud computing levels. Edge nodes provide the required comput- ing services to the devices and consume the fog services. The fog nodes also provide energy-efficient routing and communication services. Cloudlets bring cloud services to users and devices. Table 9 shows the comparison between thecomputing paradigms used in IoT.

VII. IoT SECURITY

Security is defined as the group of techniques used to prevent, restore, and ensure the safety of the data (Both application and network data) from malicious attacks [250]. Sensitive data captured from the physical world is the target of hackers. The attacker identifies and exploits the vulnerabilities to stealand modify the data [251]. Security and privacy concernsin IoT are unsolved and challenging for the researchers due to the nature of the devices being used and enabling tech- nologies [252], [253]. Further, security threats are increasingdue to the widespread adoption of IoT devices in daily life. The security measure should be addressed from the device design stage to the deployment stage [254]. In the Japanese language, the term Hajime means 'beginning.' It is a mal- ware used to perform a DDoS attack. The devices connected over the internet are the target of this malware. Totally, 297,499 unique devices were affected by this malware. This Malware originated from Vietnam, Taiwan, and Brazil [255]. IoT security concern in different stages of IoT is depicted in Figure 33. The Organization like the National Institute of Standards and Technology (NIST) working continuously to draft the cyber-security specification. Also, NIST drafted some security specifications like NISTIR 8259, 8259A, and

TABLE 9. Comparison between the computing paradigms.

Characteristics	Edge Computing	Fog Computing	Cloudlet	Cloud Computing
Architecture	Distributed	Distributed	Distributed	Centralized
Response Time	Milliseconds	Seconds	Seconds	Minutes
Number of Hops from Source	One Hop	One Hop/ Few Hops	One/ Few Hops	Multiple Hops
Latency	Less	Medium	Medium	High
Data Analysis	Real Time	Near Real Time	Near Real Time	No
Target of Attack	Very Low	Low	Moderate	High
Mobility	Supported	Limited Support	Support	Not Supported
Multitasking	Limited Support	Supports	Support	Supports
Security	Secured	Secured	Less Secure	Less Secure
Scalability	Less	Intermediate	High	Very High
Network Congestion	Very Low	Low	High	Very High
Data Storage	Very Less	Less	High	Very High
Operating Overhead	Less	Less	High	Very High
Energy Consumption	Less	Less	High	Very High
Type of Storage	Temporary	Hours to Days	Permanent	Permanent
Communication Standards	WiFi, Zigbee, Z-wave, BLE	WLAN, WiFi, 3G,4G, 5G ZigBee, Z-wave, BLE	WLAN, WiFi, 3G,4G, 5G and IP Networks	IP Networks
Computing Devices Used	Smart Sensors, Smart Phones, Smart Vehicles, Sometimes Serve	Servers, Switches, Routers, Access Points	Servers,	Servers, Data Stores
Applications	Home Automation, Smart Farmin Structural Health Monitoring, Smart Health, Smart Gardening Connected Cars	Traffic Signaling & Monitoring, Smart Health, Connected Cars	Mobile Games, Smart City, Maps, Intelligent Transportation	Smart Homes, Smart Cities, Smart Grids, Banking, SHM

TABLE 10. Mechanisms used in security services.

Security Level	Security Services	Security Mechanism	Algorithms
Information Level	Confidentiality	Cryptography	Symmetric (AES, DES, 3DES) and Asymmetric Cryptographic (RSA, DSA, IBE, ABE) Algorithms
	Integrity	Hash Functions, Message Signature	SHA-256, MD5, HMAC
	Privacy	Pseudonymity, K-Anonymity, Zero Knowledge Proof (ZKP)	EPID, DAA, Pedersen Commitment, Data Tagging, Data Obfuscation
Access Level	Authentication	Message Authentication, Network Authentication	HMAC, Password Authentication, Challenge Handshake Authentication, Extensible Authentication, Kerberos
	Authorization	Access Control Mechanism	Role Based Access Control, Attribute Based Access Control, Discretionary Access Control
Functional Level	Availability	Intrusion Detection and Prevention Systems, Access Control, Firewalls	Machine Learning and Artificial Intelligence based Intrusion Detection Algorithms

8228 [256]. In 2019, The European Union Agency for Cybersecurity (ENISA) published good practices for IoT security [257], [258]. ETSI TS 118 103 and TS 003 are the standards defined by the oneM2M, which specifies the security solutions for IoT [259].

A. SECURITY SERVICES

Security services are used to protect computer networks, information system, and its associated data from disclo- sure, alteration, disruption, destruction, unauthorized access, and use. Confidentiality, Integrity, Privacy, Authentication, Authorization, and Availability are general security require- ments [250]. The security requirements are grouped into Information, Access, and Functional level, which helps to identify and classify the target of attackers [254]. Figure 34 depicts the classification of the security levels, and Table 10 summarizes the common mechanisms used to enable security at each level.

1) INFORMATION LEVEL

At this level, the utmost concern is to defend the attacks against confidentiality, integrity, and identity of the data.

a. Confidentiality: It ensures that the data is disclosed to authorized users only. Achieving confidentiality is more complicated in the IoT since the attacker can place the sensors and capture the data [260]. Eaves- dropping is the most common attack used to exploit confidentiality [261].
b. Integrity: Integrity ensures that the data reached the end- point without modification. Maintaining integrity is sig- nificant; otherwise, it leads to false alarms/ decisions. For Example, an attacker may change the sensor data sending

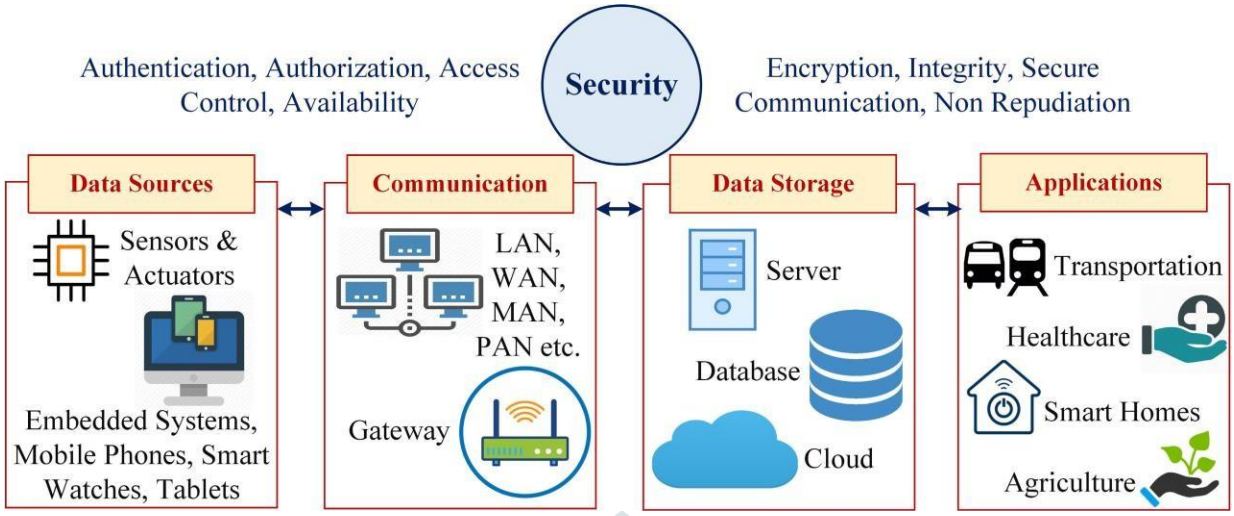


FIGURE 33. Security concern in different stages of IoT

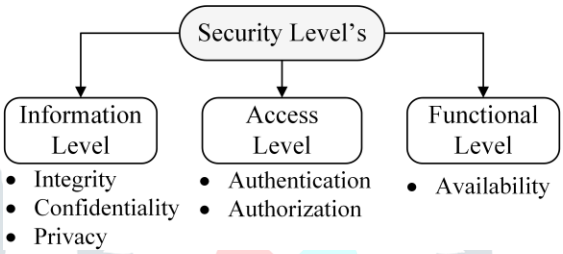


FIGURE 34. IoT security levels.

to the server. The most common integrity attack is false data injection [262].

c. Privacy: The private information of the device and users should not be disclosed during data dissemination to the unauthorized users [263]. Maintaining privacy in IoT net- works is crucial as application data contains the user's personal identity and location information most of the time. Identity theft, traffic analysis, and tracking are the most common attacks on privacy.

2) ACCESS LEVEL

The aim of the security mechanism at this level is to identify, monitor, and grant access to the services and resources to the users or devices.

a. Authentication: Authentication ensures that only legit- imate users or devices are consuming the application services and resources [264]. This is the first operation performed by the server or nodes when the user sends any request. Brute Force attack is most commonly used to exploit authentication.

b. Authorization: Authorization ensures that users or devices are permitted to access the services and resources of the IoT ecosystem. Session fixation and session hijacking are the most commonly used techniques to access the un-authorized services and resources [265], [266].

TABLE 11. Classification of resource constraint devices.

c.

Name	RAM Size	Flash/ ROM	IoT Protocol Stack Support
Class 0 (C0)	<10KiB	<100KiB	Partial
Class 1 (C1)	~10KiB	~100KiB	Extended Support
Class 2 (C2)	~50KiB	~250KiB	Full Support

3) FUNCTIONAL LEVEL

This security mechanism guarantees the provision of services and resources to the users and devices, even in the event of an attack.

a. Availability: This ensures resources like data, network infrastructure, and devices are accessible by authorized users when required. In IoT, the data generated by the devices are real-time in nature, and this data is used by the different services to take

necessary actions. The most com-mon attacks to exploit availability is Denial-of-Service (DoS) and Distributed DoS (DDoS) attack [4], [267].

B. IoT SECURITY IMPLEMENTATION CHALLENGES

1) CONSTRAINED DEVICES

The majority of IoT devices have limited resources(Computing and Storage). The use of traditional security algorithms in IoT results in high time, memory, and power consumption [268]. To secure IoT applications, an energy and memory aware lightweight security algorithms are needed. Table 11 and Table 12 highlight the classification of resource constraint devices [269], [270] and an overview of a few constraint devices used in IoT, respectively.

2) BATTERY LIFE

Most of the IoT devices are functioning in unattended environments, and it is difficult to replace the battery of the devices immediately. Power consumption is directly

TABLE 12. List of resource constrained devices.

Device Name	Controller	Architecture	RAM	ROM	Class
Arduino Ethernet REV3 (https://store.arduino.cc/usa/arduino-ethernet-rev3-without-poe)	8-bit	AVR Enhanced RISC	2KiB	32KiB	C0
PIC32MM0256GPM064 https://www.arrow.com/en/products/pic32mm0256gpm064-ipt/microchip-technology)	32-bit	RISC	16KiB	64KiB	C0
MSP430F5438A (https://www.ti.com/product/MSP430F5438A-EP)	16-bit	RISC	16KiB	256KiB	C1
CC2640 (https://www.ti.com/product/CC2640)	32-bit	ARM Cortex-M3	20KiB	128KiB	C1
Raspberry Pi Zero (https://www.raspberrypi.org/products/raspberry-pi-zero/)	32-bit	RISC	512MiB	Variable Size	C2

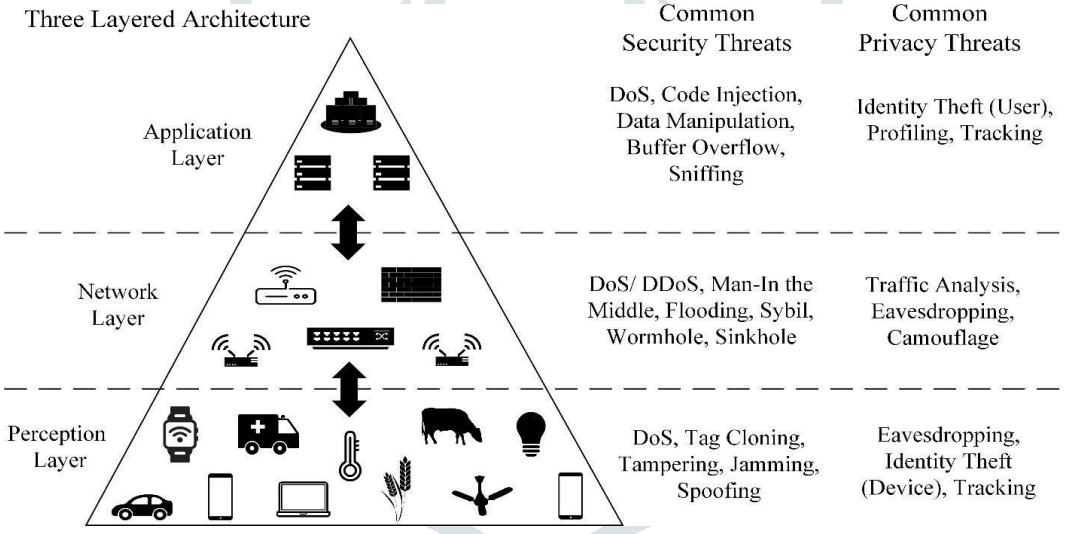


FIGURE 35. Privacy and security concern at different layers of IoT.

proportional to the computation and communication cost. There are different ways to reduce this impact on regular actions; 1. Security mechanisms should consist of fewer calculations, 2. Increase the battery capacity. (This is dif- ficult, as IoT devices are tiny and there is no additional space to accommodate larger batteries) and 3. Use of natu- ral resources such as solar and wind energy. However, this requires upgrades to hardware and special skills.

3) COMPLEX ENVIRONMENT

IoT is not a standalone system; it's an integration of entitieslike heterogeneous devices, interfaces, and people. While designing the security models, researchers should analyze theenvironment of an IoT system and enable the security feature for every entity of it.

C. COMMON SECURITY MECHANISMS FOR IoT

Figure 35 illustrates the common privacy and security threats in the three-layered architecture of IoT. Lightweight cryp- tography, Hardware security, and Intrusion Detection and Prevention Systems are used to mitigate security and privacythreats in IoT networks.

1) LIGHTWEIGHT CRYPTOGRAPHY

Lightweight Cryptography or Encryption is not a new tech- nology; R&D on it was started in 2004 [271]. Lightweight cryptography is a new branch of cryptography [254] designed to run on the constrained devices like sensors, RFID tags, smart cards, healthcare devices and so on [272]. Lightweight cryptographic algorithms are aimed at providing adequate security to the devices, data, network, and application ser- vices with minimum resource requirements in IoT. The lightweight cryptographic algorithms should possess charac- teristics like smaller key size, smaller block size, and should take the minimum number of cycles to generate cipher- text [273]. [274] analyzed different lightweight algorithms and defined features such as 1. The block size of the algorithm should be between 48-bit and 96-bit, 2. The number of rounds is limited to 16 rounds. These lightweight features offer better performance in terms of energy as compared to the traditional cryptographic algorithms.

To transmit smart grid power data securely, Li *et al.* [275] proposed a lightweight quantum encryption algorithm. The scheme consists of key distribution, authentication, and data transmission process. This scheme is based on the photon arrival time. Quantum Random Number Generator (QRNG) is used to generate the session key in the Quantum Key Distribution (QKD) terminal, and both the parties (Sender and Receiver) use the quantum channels to share the session keys during data transmission. Usman *et al.* [276] proposed a lightweight symmetric Secure IoT (SIT) algorithm to encrypt the data. The block size used in the SIT algorithm is of 64-bit (size of the key was 64-bit) and used five rounds to produce ciphertext. The authors implemented the SIT algorithm on AT-mega 328 and found that the total amount of memory occupied by the algorithm is 22 bytes. It is observed that the time taken to encrypt and decrypt the data is 0.188 and 0.187 milliseconds, respectively. Shi *et al.* [277] proposed a lightweight white-box encryption mechanism to protect the data against white-box attacks in wearable devices. The encryption and decryption are completely based on the ran- dom padding of the values. The algorithm comprised of 50-bit plain text and a 30-bit random value. It's been claimed that the amount of memory occupied by the algorithm is 95 KB.

2) HARDWARE SECURITY

IoT devices are deployed in isolated locations and subjected to various attacks like node tampering, jamming, cloning, DoS, and node tracking attacks [278]. The emerging areas like Physical Unclonable Functions (PUFs) can be adopted to enhance the security level of security in IoT [279]. PUFs are functions that generate unique identifiers for the devices by using dedicated secure processors during the manufacturing phase, and often, these identifiers are used to generate secure secret keys and key management [280].

A lightweight Secure Communication protocol for IoT devices using PUFs was developed. The proposed protocol consists of the enrolment and authentication phase. During the enrolment phase, a set of challenges was sent by the server to the devices. On the other hand, devices use the PUF module to generate the corresponding responses to the challenges sent by the server. Devices use the secure channels to send the responses, and the server is responsible for maintaining the repository of the challenge-response pairs of each node. The nodes use the corresponding identities, challenge, and response pair, timestamp, and random number to perform the authentications. The protocols designed for IoT should be efficient in terms of chip area usage, storage, processing, and energy. To address this problem, Muhammad Naveed Aman *et al.* [282] proposed a mutual authentication protocol using PUFs. A newly deployed IoT device exchanges the primary CRP with the server by making use of the time- based One-Time Password (OTP) algorithm, and the server maintains the identity and CRP of the devices in the reposi- tory. For the authentication process, the server generates the encrypted message (M = E {Id, N, R}) using device identity (Id), nonce (N), and response (R), and sends the message, challenge, and Message Authentication Code (MAC) to the device. The IoT device generates the response using its PUF based on the challenges it received from the server. Further, generated and received responses are compared and verified against integrity, source, and freshness of the message

3) INTRUSION DETECTION AND PREVENTION SYSTEM(IDPS)

IDS is a security mechanism used to identify malicious activ- ities by analyzing the data packets in the network layer [283]. Generally, IDS is comprised of following stages 1. Mon- itoring Stage – Monitors the incoming traffic, 2. Analyz-ing Stage – which analyzes and generates the pattern based on the data received from the first stage and 3. Detection Stage – It performs anomaly detection or malicious node detection [284]. The IDS plays a vital role in mitigating the DoS, DDoS, Leakage, and Masquerade attacks in the host and networks [285]. To mitigate DoS, Probe, and Generic attacks in the IoT environment Kumar *et al.* [286] proposed a Unified Intrusion Detection System (IDS) for IoT Environ-ments (UIDs). The authors used the UNSW-NB15 dataset, which contains 175341 records with 47 features. The authors used 13 features to train the different decision trees like C5, CHAID, CART, and QUEST models and generated approximately 359 rules. Based on the threshold confidence factor, UIDS selects the rules generated by each model to perform the analysis on the dataset. This IDS has shown the improved attack detection rate compare to decision tree mod-els. Leu *et al.* [287] used data mining and forensic technique to develop the Internal Intrusion Detection and Protection System (IIDPS). The system creates the personal profile of the users at the System Calls (SC) level to keep track of the usage habits of the users. The major components of the IIDPS are 1. SC Monitor and Filter – gathers the SCs submitted to the kernel and stores, 2. Mining Server – analyzes log data and users usage behavior patterns, 3. Detection Server – compares the user's behavior pattern and SC patterns collected to detect the malicious activity, and 4. Computational Grid – Mining and detection servers are part of the computational grid to enhance the speed of computing. They claimed that the rate of accuracy is about 94.00%, and the response time is approximately 0.50 seconds. Qureshi *et al.* [288] proposed a Heuristic Intrusion Detection System for IoT to detect attacks like DoS, Root-to-Local, probe, and User- to-

root attacks. The IDS is developed using the Random Neural Network (RNN). The authors used the NSL-KDD dataset with 125973 records and for analysis 29 features were used out of 41. The proposed RNN-IDS-based IDS showed the better performance against the different ML algorithms with an accuracy rate of up to 95.2%.

D. COMMON SECURITY AND PRIVACY THREATS

IoT devices and networks suffer from various security and privacy threats. This section describes common secu- rity and privacy threats. Table 13 depicts possible security threats and mitigation techniques. Table 14 shows the proba- ble privacy threats and mitigation techniques.

TABLE 13. Description of common security threats.

Potential Security Threats	Description	Mitigation Techniques
Traffic Analysis	The aim of the attacker is to find IoT devices and monitor activities of them.	Protecting the Communication Channels using Cryptographic Algorithms, Strong Mutual Authentication Protocols
Eavesdropping	The aim of the attacker is to steal the data while exchanging the data.	
Spoofing Attack	This attack purpose is to get the complete access to the application services by using valid device identity.	
Man-in the Middle Attack	In this attack, communication between the two users is captured, modified and forwarded by the unauthorized user.	
Code Injection	Attacker aims at taking the control of the device by introducing the malicious code into an IoT device or application.	Use of Strong Digital Signature Techniques and Cryptographic algorithms to secure Network
Replay Attack	An attacker retransmits an old message to the authorized user and this message accepted as a new message by the IoT devices.	
Data Manipulation	In this attack, the aim of the attacker is to alter the data (compromise the data integrity) generated by the IoT devices.	
DoS/ DDoS	The aim of the attacker is to halt the services provided by the application by flooding the SYN and ACK packets.	Analyzing the Packet Flow information in the Network, Cryptanalysis and Steganography Techniques to prevent Jamming Messages
Jamming	Jamming is a kind of attack, which disrupts the existing communication at receiver side by transmitting the interfering wireless signals.	
Blackhole Attack	Malicious node pretends to be as the shortest path to the destination node and does not forwards the routing packets to its neighboring IoT devices (Drops the routing packets).	
Sybil Attack	An attacker introduces a malicious device into the IoT network and creates the illegitimate identity of it. This malicious device acts as a legitimate device by sharing its identity within the network. The data from the devices within the network should passes through the sybil device, even though multiple paths are available.	Analyzing the network flow information, Artificial Intelligence to identify malicious nodes, Identity based Authentication Schemes
Sinkhole Attack	A malicious device acts as a sink node and tries to capture whole traffic in the IoT network.	
Tag Cloning Attack	Attacker eavesdrop the tag-reader communication to creates the replica of a legitimate tag. The attacker reads the data from the compromised tag and write them into clone tag.	

TABLE 14. Common privacy threats.

Potential Privacy Threats	Description	Mitigation Techniques
Identification	An attacker tries to grab the personal information of the individual and devices such as name, address, etc.	Anonymization, Identity Management, and Local Processing
Localization & Tracking	An attacker target is to find the location of the person and device	Location Anonymization Mechanisms
Profiling	In this attack, an attacker generates the new information by collecting and relating the original data	Perturbing, Obfuscating and Anonymization Techniques
Lifecycle Transitions	An attacker tries to capture the private information when the control handover happening between the different phases of the service	Lifecycle Transitions Detection Algorithms, Private Information Locking Techniques
Inventory Attack	An attacker maintains the repository of persons and device existence and characteristics	Unauthorized Access Methods

1) NODE REPLICATION ATTACKS

a lightweight algorithm to detect the clone nodes in Cognitive Wireless Sensor Networks (CWSN). In this system, for every specified time interval, LEACH protocol elects the Cluster Head (CH), and that acts as a Monitoring Node (MN). The MN node keeps track of its neighbor's information and acts as an intermediate between nodes and base stations. Cognitive Sensor Nodes (CSNs) in the network take the local sensing decisions and sends the details to the MN. Further, MN forwards all this information to the base station with the unique identities of the nodes. To find cloned nodes in the network, the base station applies the Clone Node Detection (CND) algorithm, and it is based on the Cuckoo filtering technique. The cuckoo filter is a data structure, which contains the fingerprint of the original keys in the hash table (look-up table). The CND algorithm applied to the list of nodes present in the base station. If the node details already exist in the look-up-table such nodes are marked as cloned nodes otherwise the nodes information is added to the look-up table. Jamshidi *et al.* [290] used the location information and watchdog nodes to spot the repli- cated nodes. In this method, watchdogs use the buffer, and which comprises of unique node identity, time and current location details of the nodes. To identify the node replica- tion, the watchdog nodes share the available information that exists in the buffer with the neighboring watchdogs in the encrypted form. Each watchdog verifies the shared and its own buffer content to find the replica node. If the different location information of a node at different time intervals is found in multiple buffers, such nodes are marked as the replica of a node. This procedure is based on the Euclidean distance between the nodes and the maximum predefined speed (threshold) of nodes. After identifying the replica node in the network watchdog report to the base station to propa- gate an alarm signal to the network. The implementation of the algorithm is done using the J-SIM simulator (Java Simu-lator), and results have shown that the false detection rate is less than 0.5%. Dimitriou *et al.* [291] proposed a lightweight and decentralized protocol to detect the replicated nodes. The detection scheme is based on random numbers (Nonce). As soon as two nodes in the network meet for the first time, both the nodes generate the random number and exchange them. Further, these nodes meet again, both the nodes request for the random number exchanged in their preceding meeting. If a node reply with an incorrect random number, then the node details are added to the quarantine list maintained by the nodes. Each node maintains the nonce list and claims list. The nonce list contains the values shared by the other nodes in the network, and this list helps in authenticating the nodes each other. The claim list comprises of nodes that are being used by the replicated nodes. The nodes exchange the quarantine list soon after the successful authentication of the nodes. The quarantine list is then added to the claims list, and once the number of claims exceeds the predefined thresholdagainst the particular node, such nodes are considered as the replicated node.

2) SPOOFING ATTACK

A resilient method against spoofing attacks in 6LoWPAN networks. The authors used the time to live and Attack Disruption Window (ADW) parameters to analyze the characteristics of the spoofing attack. The ADW length is based on the frequency with which the address of the node is changed. Therefore, the use of a temporary node address reduces the size of the ADW spoof- ing, which results in resilient against the spoofing attack. IPv6 addressing allows the node to change its address peri- odically. In this method, the authors used the temporary address, which allows the node to register with the routing table whenever the address is changed. At this moment, the MAC address of the node is bound with the new IP address. However, the attacker MAC address is bound with the node's previous IP address. The address of the node spoofed by the attackers is no longer valid to participate in the construction of the routing table. Wang *et al.* [293] proposed a two-step detection scheme to identify spoofing attacks for IoT in Mm- Wave and massive MIMO 5G communication. Firstly, Access Point (AP) extracts the MAC address and corresponding channel information of the devices and analyzes whether the anomaly is present or not by using the Virtual Channel Space (VCS). Further, the AP decides the type of anomaly (Spoofing or Others) using Logistic regression. VCS provides information about the number of virtual channels occupied by each MAC address. Therefore, the distinction between the spoofing traffic and normal traffic is based on the sparsity, overall energy, and path gains of the virtual channel. Hasan and Mohan [294] proposed a Contego-TEE framework to secure the IoT edge devices from the spoofing attacks where an adversary node sends the malicious signals to the controller and the framework developed using the embedded Linux kernel. The framework uses the trusted hardware and real-time characteristics of the system to safe-guard the physical system from intrusions. In this system, the invariant checking mechanism and timing analysis are used to detect the spoofing attack, and this continuously mon- itors the actuation command executed by the tasks. Since the system is deterministic, the number of actuation requests are restricted during the design. Hence, if a task tries to contact the actuator more than predefined actuation requests within a given period, is considered as a threat. At this, the mechanism rejects the subsequent access requests from the tasks and avoids sending the actuation command to the hardware.

3) MAN-IN THE MIDDLE ATTACK (MitMA)

proposed a technique to detect and prevent MitMA in IoT. In this technique, the IoT system consists of Fog nodes and Intrusion Detection and Preven- tion System (IDPs) nodes. IDPS uses lightweight encryp- tion algorithms to prevent MitMA and its variants, such as eavesdropping and packet modification. The primary job of the fog nodes is to provide the services to the IoT devices in lower layers. In this, all the packets except headers are encrypted using the AES algorithm, and the key exchange is using the Diffie-Hellman key exchange algorithm. IDP nodes are used to examine the Fog nodes periodically by sending the encrypted examine request (contains integer value). The IDPs define the rule, the receiver must decrypt the examined request, and the payload should be multiplied by the constant value two, and further, it should encrypt the message and resend to the IDPs. The IDPs declares the node is malicious when the response to examine requests failed to attain the

criteria. IDPs maintains the log of such packets inter-arrival time. Li*et al.* [296] demonstrated the possible threats of MitMA on the OpenFlow control channel in Fog based IoT networks. They have proposed a lightweight mechanism to detect MitMA using bloom filters. In this, the bloom filter maintains a copy of the flow of packets, and this copy is used by the controllers. The controllers collect the packet flow from the bloom filter and checks for modification in the packet flow. MitMA is detected and presented by the controller when it found any modification in the packet flow. Now, this implementation is part of the OpenFlow proto- col. To ensure the confidentiality and integrity of the data in the nuclear power plant. An encryption-based security model on Field Programmable Gate Array (FPGA) to mitigate MitMA. This model uses the AES algorithm with a 128-bit key to encrypt the data. Behavioral analysis of the proposed model analyzed using the CORE9 tool. The FPGA based encryption model has shown better performance compared to the software-based encryption model. EAVESDROPPING

Iellamo *et al.* [298] proposed an intelligent jamming strategy to improve security against eavesdropping. In this, the secu- rity capacity of the channel is determined using Shannon's formula for channel capacity. The concept of the neutralized zone and Artificial Noise (AN) are used in this strategy. The neutralized zone is the region where all the eavesdroppers are deactivated, and AN is a noise generated and broadcasted by the IoT-Gateway to make the neutralized zone of some radius centered at the IoT-Gateway. IoT nodes can examine the surroundings and deactivate the eavesdropper falling inside the neutralized zone. In this strategy, potential eavesdroppers are deactivated using the jamming activities achieved using the In-band Full Duplex IoT gateway and cooperative helpers. Liu *et al.* [299] proposed a SecLight framework to mitigate eavesdropping in IoT. The devices communicate using Visi-ble Light Communication (VLC) and, the required signal is sent through Light Emitting Diode (LED). To increase the security in VLC communication systems, the authors used the multipath redundancy, time reversal, and random choice techniques, which can make transmitted signals automati- cally focus on the authorized users while interference of the eavesdroppers present in the communication channel. The authors analyzed the effect of channel correlation and esti- mation error on the system using the additive noise, leaking factor, and time-varying error. This framework ensures the security in both multi-input single output and single input single output VLC. Wang *et al.* [300] proposed a method to minimize the eavesdropping attack between the sensors and the controllers in the green cyber-physical transportation system. This communication model consists of a jammer, eavesdropper, and a communication channel. In this method, the secrecy of the communication is increased by using the feedback received from the controller, and the sensors are able to adjust its condition using this feedback. This method proved the existence of Stackelberg Equilibrium between the sensors and jammers. To minimize the eavesdropping attack, the model allows the sensors to use dynamic transmission power.

4) DoS/ DDoS ATTACK

Chen *et al.* [301] proposed a method to detect DoS attack in low rate ZigBee networks using the Hilbert-Huang Trans- form (HHT) approach and trust of Intrinsic Mode Func- tion (IMF) components. The trustworthiness of IMF is estimated by combining the correlation co-efficient and Kolmogorov-Smirnov (KS) approaches. Firstly, the HHT approach is applied to the normal traffic (NT) and attack traffic (AT) to obtain the IMF component of the NF and AT, respectively. Secondly, calculate the correlation coefficient and calculate probability using KS of AT with their original traffic. Further, analyze the trust of the IMF component based on the DoS detection rule. Doshi *et al.* [302] demonstrated the DoS detection for consumer IoT devices using the ML techniques. In this work, DoS detection was carried out at the packet level. The features like packet size, inter-packet arrival time, type of transport layer protocol, bandwidth, and destina-tion IP address are considered to detect the DoS attack in the IoT network. The authors used five (KNN, LSVM, Decision Tree, Random Forest, and Neural Networks) different ML algorithms to distinguish the normal and DoS packets. Both the decision tree and KNN has shown the accuracy of 99%. Kajwadkar *et al.* [303] discussed the CoAP protocol and its DTLS security issues. They proposed an algorithm to detect DoS and DDoS attacks in IoT networks, communicating using CoAP. The detection algorithm was implemented at the 6LoWPAN border router gateway. The algorithm monitors the incoming traffic and decides whether the incoming packet is suspicious or not by using the black and graylisted IPs. The border gateway node extracts the incoming packet header and verifies whether the source IP belongs to blacklisted IP or not. The incoming packets were dropped if the packets belong to the blacklisted IP. The payload is considered malicious if the size of the payload is greater than the threshold payload. The gateway receives the packets from the same source with similar characteristics, then the DoS attack is found and the IP address of the device is added to the black list. DDoS attack is found when the gateway receives the packets from the differ- ent sources with the same features, then the different source IP addresses are added to the grey list. Anirudh *et al.* [304] proposed a honeypot model to protect IoT devices against the DoS attack. Honeypot acts like a main IoT device and divert the attention of the attackers. In this, all the incoming packets should pass through the IDS to the Server. If IDS find malicious packets, such packets are directed to the honeypots and the information related to the packets (IP address and MAC address) are stored as logs. Each time, information of the packets is examined against the information present in the logs.

5) TAG CLONING ATTACK

Sufang *et al.* [305] presented a Deterministic Cloned Tag Detection Protocol (DCTDP) to find the cloned tags. This protocol uses the Tree-Based anti-collision method to detect collisions. To detect cloned tags without revealing sensitive information, the protocol uses the pseudonym update method. For distributed IoT, Gope *et al.* [306] presented an RFID based authentication Architecture. This architecture uses the lightweight RFID-based authentication method, which pro- vides secrecy, traceability, and anonymity of RFID-tags and secure localization. Kaur *et al.* [307] proposed an authenti- cation protocol to mitigate replaying, tracking, and cloning attacks. This protocol, developed using the Elliptical Curve Cryptography (ECC) launches the mutual authentication between the tags and the servers.

VIII. PLATFORMS FOR IoT

Generally, IoT is a collection of resource-constrained and resource-aware devices, which are communicating over the internet [308]. The Operating System (OS) is an essential component of any IoT device; it acts as an interface between the physical world and the user applications [309]. In gen- eral, Kernel, System Shell, and Utility Software are part of the OS. The Kernel is the core of the OS, which per- forms resource management. Using system shell, users can access the kernel. Software Utility is a collection of system software such as Assemblers, Debuggers, and Compilers to perform OS level tasks. The operating system like Raspbian, Ubuntu Mate, Snappy Ubuntu and Windows 10 IoT Core are used in the higher level IoT platforms like RaspberryPi, Beagle bone, OrangePi, Samsung Artik, Intel Edison and Galileo, and many more, which requires a large amount of memory.

A. PLATFORMS FOR CONSTRAINED DEVICES

The resource constraint devices make use of the Real-Time Operating System (RTOS), as it supports resource manage- ment, data management, safety, and security. The signifi- cant challenges like connectivity, memory management, and support for heterogeneous devices must be considered whiledesigning the IoT applications on top of the RTOS.

1) RIOT

RIOT is a microkernel-based operating system designed for IoT devices and embedded systems, which is written in C with an open-source license. RIOT uses the modular inter- net stack, and it provides a rich and consistent set of inter- faces to develop IoT applications. Static & dynamic memoryallocation, multi-threading, and tickles scheduling are the main feature of this OS. Tickles scheduler is used to mini- mize power consumption. It does not have a floating-point unit [309], [310].

2) FreeRTOS

This operating system used for real-time processing, which runs on the many of the microcontrollers, including advanced ARM CortexTM - Mx series, and it is written in C. This comes with the various real-time features like - schedul- ing, inter-process communication, timing analysis, and syn- chronization [311], [312]. FreeRTOS-Memory Protection Unit (MPU) is used to secure the Kernel from the invalid execution of the tasks. It supports TCP/IP and a LightweightTCP/IP (LwIP) stack, which is built on IPv6 and less power consuming protocols like 6LoWPAN, CoAP, and manymore.

3) ContikiOS

It was the first operating system developed for sensor networks with features like dynamic memory allocation, multi-threading, multi-tasking, protothreads, TCP/IP proto- col, IPv4, IPv6, simple web server, telnet clients, and many more [313]. Because of these features, ContikiOS is used bymany IoT applications such as smart city monitoring, indus-trial monitoring, alarm systems, construction site monitoring, and many more. This OS also supports the network simulator cooja, which enables developers to create and analyze the network and tests their code. TinyOS

This operating system was developed and managed by the TinyOS alliance [313], [314]. TinyOS written in NesC lan- guage with an open-source license [315], [316]. The size of the OS is around 400 bytes. TinyOS widely used by wireless sensor applications, personal networks, ubiquitous comput- ing [317]. The CPU will be in sleep mode while the scheduler has no tasks to perform, and this facilitates the energy-saving feature. TinyOS consists of components, which eases the application developers to develop diverse applications. The components comprise of three computational entities, such as Commands, Events, and Tasks. A command is a request sent to a component to perform certain tasks. The tasks are consisting of reading and writing the sensor data. An event isan indication of the completion of the service.

4) LiteOS

It is a lightweight OS designed for low power devices and suitable for applications like wearables, connected homes, V2V communications, etc., [318]. This OS was developed by Huawei in 2015, and it claimed that the memory occupiedby this OS is less than 10KB [319]. The main features are zero-configuration, auto-discovery, fast boot, and hierarchi- cal file system. The major components of the LiteOS are 1. LiteShell (Used with the user intervention) – provides the set of commands to perform file and process management and debugging 2. Lite Filesystem (LiteFS) – the sensors are con- sidered as the files, and the file system developed using LiteCand 3. Kernel – which supports features like multithreading, scheduling and provides the callback functions to handle theevents efficiently [320].

5) VxWorks

This RTOS supports Intel, ARM, and POWER architectures of 32-bit and 64-bit microcontrollers. It supports IPv4/ IPv6 network stack and adds security features like secure booting, kernel hardening, and limited access control to the RTOS. In VxWorks, real-world problems can be solved using the programming languages like C and C++, along with this modern programming language like Python3 is also sup- ported by the RTOS. It also assists the multimedia libraries such as OpenVG, OpenGL, OpenCV, and audio-video han- dling libraries [321], [322].

6) ERIKA ENTERPRISE

Erika Enterprise is a hard-real-time open-source operating system suitable for low power and memory computers. It is written in C. ISO 17356 API (OSEK/VDX API) has been adopted by the Erika Enterprise, and it supports for 8-bit, 16-bit, and 32-bit microcontrollers. One of the essential fea-tures of this RTOS is the sharing of the stack between differ-ent tasks; this increases the efficient utilization of the memoryby eradicating the use of dedicated stack for individual tasks.The use of RT-Druit with the eclipse plugin allows program-mers to develop the graphical user interfaces [323].

TABLE 15. Comparison of RTOS's.

Characteristics	ContikiOS	FreeRTOS	RIOT	TinyOS	LiteOS	VxWorks	Erika	Nut/OS
Version	V3.0, 2015	V10.2.1, 2019	V2.1	V2.1.2, 2012	V2.1, 2018	V7.0	V3.0	V5.2.4
Architecture	Modular	Microkernel RTOS	Microkernel RTOS	Monolithic	Modular	Modular	Modular	Modular
Hardware Support	AVR, TI CC2538, MSP430, MSP430x	ARM Cortex – A5, M3, M4, M7, MSP432, MSP430, Xilinx,	AVR, ARM, Cortex-M, X86, MSP430	Atmel AVR, ATmega1281, TI – MSP430, CC2420	Mica Z and IRIS	Intel X86, Fujitsu FR-V, MIPS	Atmel AVR, ARM Cortex M4, MSP 430, Mica Z	8-bit AVR, AVR32, ARM7, ARM9, ARM Cortex M3
Memory Management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Power Management	Yes	Yes	Yes	Yes	Yes	-	-	-
Scheduling	Cooperative	Preemptive	Preemptive	Non-Preemptive	On Task Completion	Preemptive	Preemptive & Non-Preemptive	Non-Preemptive
Scheduling Algorithms	Priority (Round Robin)	Priority (Round Robin)	Priority (Round Robin)	FIFO	Priority (Round Robin)	Priority/ Adaptive	Dynamic (Fixed Priority, Earliest Deadline First)	-
Programming Model	Multi-Threading	Multi-Threading	Multi-Threading	Event Driven	Multi-Threading	Multi-Processing	Multi-Threading	Multi-Threading
Concurrency	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hardware Requirements in KiB	Min-RAM: 2 Min-ROM: 30	Min-RAM: 10 Min-ROM: 12	Min-RAM: 1.5 Min-ROM: 5	Min-RAM: 1 Min-ROM: 4	Min-RAM: 4 Min-ROM: 128	Min-RAM: 2MiB Min-ROM: 512	Min-RAM: 2 Min-ROM:12	Min-RAM: 4 Min-ROM 128
Targeted Device Class	C0 and C1	C1 and C2	C1 and C2	C0 and C1	C0 and C1	-	C0 and C1	C0 and C1
Simulator	Cooja, Netsim	-	IoT-Lab	TOSSIM, PowerTossim	AVRORA	-	-	-
Database	Coffee, Raima	ITTIA DB, Raima	Raima	TinyDB, Raima	-	Raima	-	-
IoT Protocol Stack	Yes	Yes	Yes	Yes	Yes	Yes	Partial	Partial
Security	TLS	TLS (WolfSSL)	-	Cipher Block Chaining	-	SSL/ TLS, Cryptography, SecureBOOT	-	TLS (WolfSSL)
Network Stack	uIP, RIME	-	OpenWSN	BLIP	-	Yes	-	Nut/Net (TCP/IP)
Applications	Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, Industry 4.0, Smart City	Wearables, Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, V2V Communication, Smart Water Grid, Smart Fisheries	Aerospace, Defense, Industry 4.0, Smart Health, Consumer Electronics	Smart Homes, Wearables	-

TABLE 16. RTOS IoT protocol stack.

Operating System	IoT Protocol Stack							
	802.14.5	6LowPAN	RPL	MQTT	CoAP	DDS	XMPP	Websocket
ContikiOS	✓	✓	✓	✓	✓	×	×	×
FreeRTOS	✓	✓	✓	✓	✓	×	×	×
RIOT	✓	✓	✓	✓	✓	×	×	×
TinyOS	✓	✓	✓	✓	✓	×	×	×
LiteOS	✓	✓	✓	✓	✓	×	×	×
VxWorks	✓	-	-	✓	✓	×	×	×

Note: *Real-time operating systems like Erika Enterprise and Nut/OS provides the Networking stack (USB Stack, IPv6, Wireless LAN, and Security). However, these OS's need to provide IoT protocols stack*

7) NUT/OS

Nut/OS is an operating system for memory-constrained devices. The programming language like C and Lua scriptinglanguages can be executed interactively. It supports TCP/ IP protocol stack along with the prioritized event handling. The transport and network layer protocols like TCP, UDP, IP, andICMP are part of the TCP/IP protocol suite. Additionally, the operating system facilitates application layer protocols like FTP, HTTP, DNS, and DHCP. Nut/OS supports the trans- port layer security by using wolfSSL for embedded security.Currently, networking features like IPv6 and WLAN are notavailable in this operating system [324].

Table 15 depicts the comparison between the RTOS's andTable 16 shows the IoT protocol stack supported by the different RTOS's [318], [319], [325].

B. OTHER IoT PLATFORMS FOR IoT DEVICES

These operating systems are designed for resource efficient IoT devices like smartphones, Televisions (TVs), smart-watches, and high-end IoT development boards. These typesof OS's provide rich GUI and other libraries to carry out specific tasks.

1) TIZEN OS

Tizen is an operating system built from scratch to meetthe requirements of the connected devices. Tizen operating system is

used in the in-vehicle infotainment system, Smart TVs, smartwatches, mobiles, etc. Tizen IoT is headed or headless binaries support for the embedded devices. Tizen IoT supports developers to implement IoT applications. Tizen IoT supports RaspberryPi3 and Samsung Artik family devel-opment boards [326].

2) ANDROID THINGS

Android Things is an android based operating system devel-oped by Google, which uses Linux kernel; this helps to android developers to build IoT projects easily and swiftly. The OS uses the 32MB of RAM. Android Things supports a variety of hardware, such as RaspberryPi3, NXP i.MX7D, Qualcomm SDA624, and MediaTek MT8516. The weave is an application layer protocol, which comes with the OS; this protocol can be implemented over IPv6 using BLE, WiFi, Cellular, and thread communication standards [327].

3) UBUNTU CORE

Ubuntu Core is the Linux operating system for embed-ded devices. It facilitates optimized and reliable updates to develop IoT applications with advanced features. It is easy to deploy with the minimum hardware requirement (256MB of RAM and 512MB of flash memory). It is tamper-resistant and hardened against memory corruption. By default, this OS does not provide any GUI. However, Wayland or Mir options can be used. Ubuntu Core supports Artik, Intel, Qual- comm, Beaglebone, Snickerdoodle, and RaspberryPi fam- ily devices. Ubuntu Core establishes communication using Ethernet, WiFi, and BLE [328].

Klingensmith and Banerjee [329] proposed a lightweight hypervisor called Hermes to accomplish the runtime necessi-ties of the embedded software. Always, RTOS schedulers do not provide swift responses to the I/O events. Hermes archi-tecture offers the solution for these kinds of problems by plac-ing the abstract layer between the I/O devices and software services. The authors have used the ARM Cortex M7 CPU to measure the Interrupt Service Routine (ISR) latency (Time between starting of ISR execution and starting of user-space data processing) and measuring of the ISR latency cap- tured in Free RTOS. This architecture has the advantages like enabling and disabling of the interrupt sources and allows the switching between the network port and the serial port. Raymundo Belleza and Pignaton de Freitas [330] ana- lyzed the performance of the RTOS like FreeRTOS, RIOT, Zephyr, and μC/OS-II and μC/OS-III. The analysis was per- formed based on the parameters like task switching time, semaphore passing time, time taken to exchange messages between the tasks, time to get and release memory region, and time is taken to activate a task within the interrupt service routine. After the successful analysis, the authors pointed out the following observations 1. μC/OS-II and μC/OS-III have well suited for safety-critical applications like avion- ics, industry 4.0 and healthcare devices in smart health,

2. Privacy and security are the primary concern, then the open source RTOS's like FreeRTOS, RIOT and Zephyr are the best choices for the IoT devices, and 3. Zephyr is a good choice for the application, which needs an immediate response. **OPEN ISSUES AND FUTURE DIRECTIONS**

In earlier sections, a comprehensive study on system-level aspects of IoT, such as IoT architectures, communication standards, application layer protocols, computing paradigms, security, and real-time operating systems, are discussed. Attributed to the above study, current research issues and future directions have been addressed in this section.

A. IoT ARCHITECTURE

Due to the lack of standards in IoT, many IoT architec-tures have been developed like Computing, Secure, IoT-SDN, SOA, Middleware Architectures, etc. An in-depth examina- tion of many IoT architectures is unexplored. Several Com- puting architectures/ platforms have been introduced, but they are not interoperable, and this becomes a big issue when the multiple applications are integrating. Integration becomes difficult due to the use of abundant services from various platforms, protocols, and file formats. Creating seamless con-nectivity between different applications is of the researcher's interest. IoT devices are increasing exponentially, and these devices are mobile in nature. Developing a scalable and mobility-supported architecture for IoT is still a challenging issue. IoT devices may perish due to power constraints and go offline due to infrastructure constraints (Bandwidth con- straints). Design and development of self-healing and self- configurable architectures are much needed. Recent days, IoT middleware architectures have gained the researcher's inter- est. IoT middleware architectures should provide real-time and secure services to the mission control applications like health monitoring, disaster management systems, etc.

B. COMMUNICATION

Signaling plays a vital role in the exchanging of information between the devices. The data need to be exchanged between node A and B swiftly and reliably in IoT networks. The com-munication standards like WiFi and Thread uses the 2.4GHz radio, due to this signal may interfere. This increases the level of noise and weakens the signal strength. IoT devices require a persistent internet connection to share the information at the global level, which consists of Open Ports. Unauthorized access to Open Ports leads to attacks like malicious code execution, DoS, DDoS, and code injection. Maintaining the bandwidth in IoT networks is essential to provide the QoS to the end-users. The efficient use of available bandwidth in IoT networks is also essential. In the future, the requirement for the bandwidth may increase as the number of devices and data increases. Currently, WiFi is used widely in IoT networks to share the information, which does not use any power management techniques. The low range communica- tion

technologies like BLE, Zigbee, and Z-Wave consumes less power as compared to WiFi communication standard. Power management is significant since the majority of the IoT devices are operating in remote areas. Due to the power and bandwidth constraints, sometimes devices may go offline. Continuous monitoring of the IoT devices is still challenging. High signaling overhead in 5G networks increases the energy consumption in IoT devices. The small cells, such as pico- cells, microcells, and femtocells, bring the QoS capabilities into 5G IoT networks. Also, the use of small cells in 5G IoT networks provides the extended battery life of IoT devices; this is due to the low latency in the novel 5G IoT wireless network environment.

C. APPLICATION LAYER PROTOCOLS

Study of IoT application layer protocols at different environ-ments (resource-aware and resource-constrained) with dif- ferent load and diversity of data (Text, Audio, Images, and Videos) is needed. Generally, IoT application layer proto- cols use Transport Layer Security (TLS and DTLS), which increases the latency, and consumes more resources due to the expensive handshaking procedure; TLS and DTLS are easily susceptible to the Man in the Middle attacks. Developing a strong and lightweight authentication mechanism is still challenging. The usage of protocols like QUIC and HTTP 2.0 is increasing; adopting these protocols into the IoT networks is a researcher's concern. Moreover, the computational operations are moving from Cloud to the Edge of the network. This transformation raises challenges like data management and security issues. A huge number of lightweight encryption algorithms are reaching the IoT community. However, testing and validating the performance of these algorithms by incorporating them into the IoT application layer protocols is also a major research concern. Privacy issues like anonymous communication still require major attention. Most of the application layer protocols are compatible with traditional communication standards like WiFi and local area networks, and these protocols are needed to be compatible with all the communication standards. XMPP does not support QoS and enabling the QoS feature in it is a researcher's concern. IoT application layer protocols use First in First out data delivery strategies, but priority-based data delivery strategies need to be incorporated to address the real-time IoT applications. Due to the in-band binary data transfer limitation, XMPP does not support for sharing of the high-resolution images and videos.

D. COMPUTING PARADIGMS

Cloud and fog computing architectures suffer from server downtime, limited control on the services, and security. Mak-ing edge node work in cooperation with the cloud and fog computing architectures is a challenging task. Data analytics, immediate response, intelligence, and security are the signif- icant characteristics of edge computing architecture. All the time, edge computing devices are not sophisticated servers. Recent days, ML, DL, and AI algorithms are common in IoT applications. Using these algorithms at the edge level turns the edge node into an intelligent edge node. Implementing energy-efficient ML, DL, and AI algorithms in emerging applications like Autonomous Vehicles, V2V communica- tion, Intelligent robots, Outlier detection, AR & AR, and others to make effective decisions is also a challenging task. Edge computing should support for the various QoS features like scalability, mobility, scheduling, and resource utilization. Improved energy-efficient device management and monitor-ing techniques that need to be implemented at the edge level are beneficial. Privacy & security is one of the serious issues in IoT, and researchers should focus on developing trust management techniques in edge devices, which reduces the latency in massive IoT networks. IoT users are increasing, and edge computing should take care of user and resource allo- cation algorithms to provide reliable and scalable services. 5G technical standards are still evolving, and edge computing should create an environment to develop 5G applications.

E. SECURITY

Numerous security mechanisms have been proposed to pro- tect the ecosystem of IoT. In the real world, it is a tedious task to reduce latency and increase the computation speed of the security algorithms in IoT applications. Most of the IoT applications are controlled using mobile applications, and securing such mobile applications is also a major con- cern. Many Lightweight cryptographic algorithms have been introduced to protect data from the attackers. The use of hybrid encryption algorithms, simplifying the key generation mechanisms, provides a high level of security and enhances the computational speed. Different middleware technologies were introduced to reduce the data processing and communi- cation overhead in the IoT networks. Authorization, privacy, encryption, securing the web interfaces, and adequate soft- ware protection are the major security concerns in the IoT middleware. IoT is a self-organizing network; any device can become a part of the network at any point in time. Design and development of novel privacy-preserving lightweight device authentication schemes are essential to enhance the qual- ity of the services to legitimate devices. Now the trend is moving toward the use of biological characteristics (finger- prints, iris, and facial data) to authenticate the users, which poses new challenges. Employing of ML and DL algorithms are increasing in the area of security to predict DoS and DDoS attacks. These algorithms require authentic IoT traf- fic datasets to build efficient security solutions. One of the major issues with the IoT deployment is radio jamming since most of the communication is based on wireless standards. An attacker may disrupt the data transmission by propagating the powerful radio within the proximity of an IoT device. BLE devices in IoT networks may be susceptible to various attacks; this is due to the use of predictable smaller secret keys. Blockchain technology has an impact on applications like supply chain, Industry 4.0, and smart grids. However, the decentralized architecture of the blockchain results in increasing in high energy depletion, storage, and computation overhead. Energy-efficient blockchain solutions need to be developed for resource-constrained IoT devices. Future IoT communication is 5G, which poses various security chal- lenges due to its network architecture and heterogeneous access. Design and development of authentication, identity management, trust models, and privacy protection algorithms increase the security features in 5G IoT networks. *IoT PLATFORMS*

Most of the RTOS's use the preemptive scheduling algorithm and have the drawbacks 1. The instructions of the high pri- ority

tasks may not be executed in the given time, and 2. Sometimes the low priority tasks may wait for an indefi- nite time to execute. The scheduling algorithms developed for traditional OS's are not suitable for real-time applica- tions. Designing and developing context-aware scheduling algorithms is significant. The recent networking technolo- gies like IPv6, RPL, 6LoWPAN, CoAP, and MQTT arepart of most of the OS. Adding new IoT protocols like AMQP, XMPP, DDS, WebSocket, and others to the RTOS is still the researcher's concern. The interoperability issue arises due to the availability of wide range OS's (RIOT, FreeRTOS, ContikiOS, TinyOS, ERIKA, LiteOS, MantisOS, Ubuntu, Android, Microsoft Windows, MACOS and etc.), Data Structures, programming languages, protocols, and IoTarchitecture. Solving interoperability at the different levels of platforms is also a researcher's concern. The services pro- vided by the schedulers to the I/O events are not deterministic. Enhancing the quality of the services to the I/O events is a researcher's motive. RTOS supports programming languages like C and C++. However, the most widely used program- ming languages are Java, Python, and JavaScript. Developing the RTOS's compatible with these languages is also one of theprime concerns; this brings the new application features to theIoT applications.

IX. CONCLUSION

In this survey, a basic understanding of IoT, along with the current research trends in IoT technologies as of 2020, is presented. Next, the characteristics and requirements of theemerging IoT applications are provided. Then, the current challenges in the IoT system design are analyzed. Later, the present research trends in IoT architectures are articu- lated and also described the issues that are necessary to be addressed in the future. Subsequently, the commonly used communication standards in IoT, along with future research directions, are presented. Further, investigation on the char- acteristics of IoT application layer protocols and furtherrequirements of the IoT application layer protocols are pro- vided. Next, the necessity of the computing paradigms in IoT, along with detailed analysis of Cloud, Cloudlets Fog, and Edge computing architecture in detail, are deliberated. Then, the security features of IoT are emphasized and presented a detailed study on security issues, current research, and future scope in IoT security. Further, an analysis of RTOS and the available protocol stack for IoT is provided. At last, a detailed discussion on open research challenges and future directions is provided. From the analysis, it has been noticed that the foremost requirements like reliability, scalability, adaption, context-awareness, interoperability, embedded intelligence, privacy & security issues need to be addressed more pre- cisely. Currently, researcher's focusing on solving the inter-operability at the cloud and cloudlets. However, solving the interoperability issues at the edge level brings lots of benefits to the IoT applications. Our goal is to design and develop an energy-efficient, interoperable, and secure reconfigurable edge node for IoT applications.

ACKNOWLEDGMENT

The authors would like to thank the P. C. Panchariya,Director,
CSIR-CEERI, Pilani, for his encouragement and constantsupport.

REFERENCES

[1] N. Bălău and S. Utz, ''Information sharing as strategicbehaviour: The role of information display, social motivation and time pressure,''Behav. Inf. Technol., vol. 36, no. 6, pp. 589–605, Dec. 2017.

[2] H. Liu, D. Han, and D. Li, ''Fabric-IoT: A blockchain-based access control system in IoT,'' IEEE Access, vol. 8, pp. 18207–18218, 2020.

[3] P. Sethi and S. R. Sarangi, ''Internet of Things: Architectures, Protocols, and Applications,'' J. Electr. Comput. Eng., vol. 2017, pp. 1–25, Jan. 2017.

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ''A survey on Internet of Things: Architecture, enablingtechnologies, security and privacy, and applications,'' IEEE Internet Things J., vol.4, no. 5, pp. 1125–1142, 2017.

[5] V. Miori and D. Russo, ''Improving life quality for theelderly through the social Internet of Things (SIoT),'' in Proc. Global Internetof Things Summit (GIoTS), Geneva, Switzerland, 2017, pp. 1–6.