https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

IOT future in Edge Computing

Vandana C.P[1], Dr. Ajeet A. Chikkamannur[2]

[1]Assistant Professor, ISE Department, New Horizon College of Engineering, India)
[2]Professor and Head of Department of Computer Science, R. L. Jalappa Institute of Technolog, India

Abstract— *With the advent of Internet of Things (IoT) and data convergence using rich cloud services, data computing has been pushed to new horizons. However, much of the data generated at the edge of the network leading to the requirement of high response time. A new computing paradigm, edge computing, processing the data at the edge of the network is the need of the time. In this paper, we discuss the IoT architecture, predominant application protocols, definition of edge computing and its research opportunities.*

Keyword— IoT, Cloud, Fog computing, edge computing, COAP.

I. INTRODUCTION

Internet of Things (IoT) is the convergence of connecting people, things, data and processes which will transform human life, business and everything revolving around it. Cisco predicted a $19 trillion profit market for IoT, and estimates that there will be 50 billion smart objects connected to the Internet by 2020. These are motivating reasons for companies to put their label on this coming IT tsunami.

CLOUD computing paradigm is a major evolution in the way we work, analyze the data since 2005. Applications like Google Apps, Facebook, Twitter and Flicker based on cloud, have been widely used in our daily life. Efficient data processing engines and scalable infrastructures Google File System [2], MapReduce [3], Apache Hadoop [4], Apache Spark [5], supports cloud services.

IoT applications may insist for very short response time, involve confidential data, hugh bulk steaming of data creating heavy loads on the networks. Cloud computing paradigm may be inadequate to tackle such application requirements. There is a data push from cloud services and pull from IoT, creating a need for paradigm shift to handle these issues leading to edge or fog computing.

Fog computing or "edge computing," means that rather than hosting working from a centralized cloud, fog systems operate on network ends. Fog Computing extends the cloud computing paradigm to the edge of the network. It resolves the issues related to applications that require very low latency, geographically distributed applications, fast mobile applications, large-scale distributed control systems (smart grid, connected rail, smart traffic light systems).

Defining characteristics of the Fog are: low latency and location awareness; wide-spread geographical distribution; mobility; very large number of nodes, predominant role of wireless access, strong presence of streaming and real time applications, heterogeneity. Edge computing has the potential to address the concerns of response time requirement, battery life constraint, bandwidth cost saving, as well as data safety and privacy.

The remaining parts of this paper are organized as: Section II presents the IoT architecture model, Section III discuses about the various application protocols in IoT, Section IV talks about the needs for Edge computing and its definition, Section V presents the research direction in the field of edge computing, followed by Section VI with the conclusion.

II. IoT ARCHITECTURE

Due to the need to interconnect millions of physical devices (sensors and actuators) into the IoT ,the requirement for a comprehensive, flexible layered architecture for IoT is a must. Till now there is no standardization done in the field of IoT architecture. The basic model consists of 5 layer model [17] as shown in the fig 1.

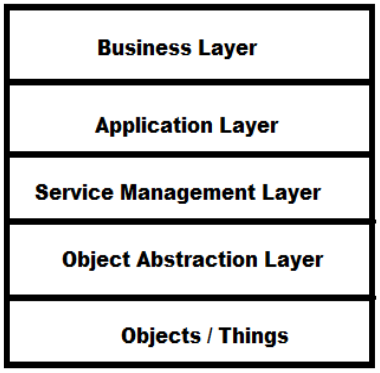

Fig.1: Five –layer IoT architecture

Objects/Things: The represent the physical sensors and actuators in IoT to collect and process the data. Various functionalities like querying location, temperature, light, humidity are performed by these sensors. This layer digitalizes and transfers the data to the higher layer.

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

Object abstraction Layer: It will abstract the fine details of the data and transfer the data produced by various objects to the service management layer using various technologies like RFID, GSM, WiFi, Bluetooth, ZigBee, infrared etc. Cloud computing and data management functionalities are performed by this abstraction layer.

Service Management: Middleware (pairing) layer pairs a service with its requester based on addresses and names. This layer enables the IoT application programmers to work with heterogeneous objects without consideration to a specific hardware platform. Also, this layer processes received data, makes decisions, and delivers the required services over the network
wire protocols.

Application Layer*:* The application layer provides the services requested by customers. For instance, the application layer can provide temperature and air humidity measurements to the customer who asks for that data. The importance of this layer is that it has the ability to provide high-quality smart services to meet customers' needs. The application layer covers numerous vertical markets such as smart home, smart building, transportation, industrial automation and smart healthcare.

Business Layer: The business (management) layer manages the overall IoT system activities and services. The responsibilities of this layer are to build a business model, graphs, flowcharts, etc. based on the received data from the Application layer. It is also supposed to design, analyze, implement, evaluate, monitor, and develop IoT system related elements. The Business Layer makes it possible to support decision-making processes based on Big Data analysis. In addition, monitoring and management of the underlying four layers is achieved at this layer.

The fig 2 which represents IoT reference model [7] details the architecture layers to 7 broad layers. Edge computing layer plays a significant role in this architecture model. In this layer, data elements present at the edge of the network act as both data producers and consumers and the relevant data is processed at the edge instead of the centralized cloud. Remaining layers enumerates same data collection, storage, abstraction and processing as represented in fig1.

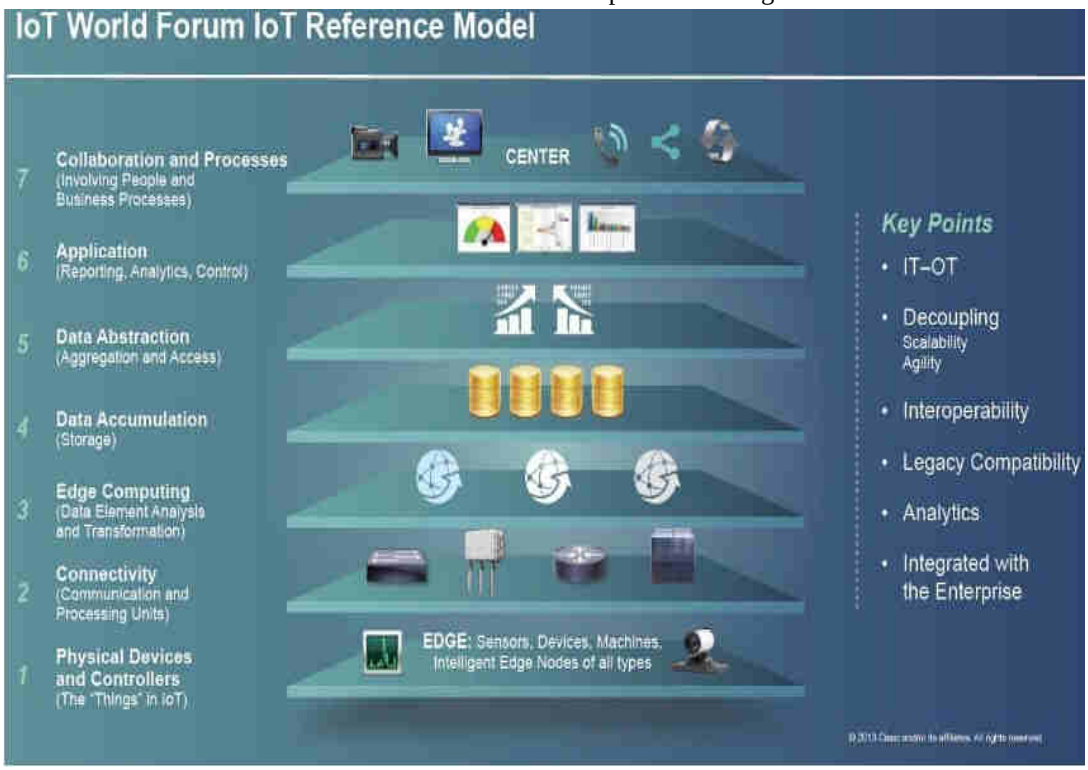

Fig.2: IoT Reference Model

Fig 3 represents the major IoT elements starting from identification, sensing, communicating to the sink, computing and providing services through semantic analyze. Identification process in IoT requires unique identifier and addresses for the objects. Unique identifiers will follow a naming convention, like for a temperature sensor T1, but the address will uniquely map these objects within the communication network.

Addressing techniques in IoT include IPv4 and IPv6 conventions. For low power wireless networks, 6LOPWPAN [5] provides a compressed technique over IPv6 headers. Resource discovery Services, Information Aggregation Services, Ubiquitous Services are some of

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

the main services identified in any IoT applications. Ubiquitous services obtain the information from aggregation services for decision making leading to ubiquity (anyone anytime anywhere).Semantic analysis is the extraction of the correct meaning from the data to build the knowledge which helps in providing services for making proper decision making. It includes discovering the resources and modelling the information from the raw data. This data is analyzed to make the right decision to provide the most accurate service. Thus, semantic element forms the brain of the IoT .This need is supported by Semantic Web technologies such as the Resource Description Framework (RDF) and the Web Ontology Language (OWL), *Efficient XML Interchange* (EXI) formats.

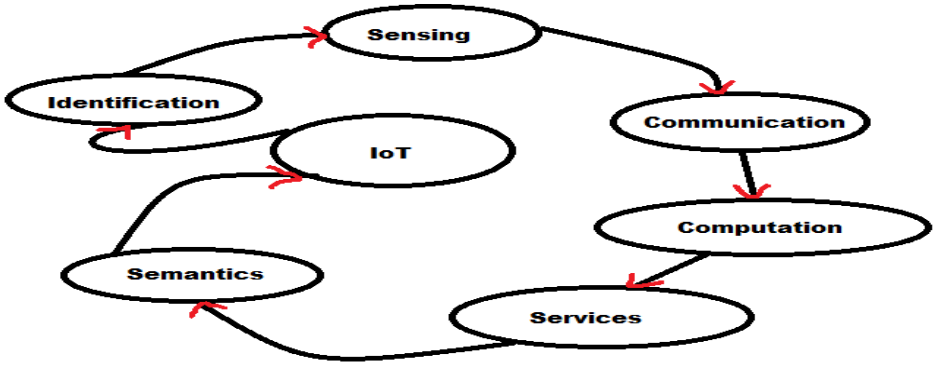

Fig.3: IoT components

III. IoT ENABLING APPLICATION PROTOCOLS

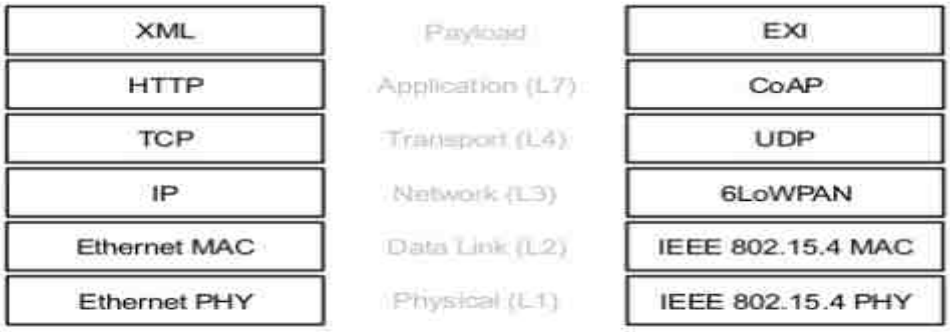

Fig.4: IoT reference layers

Constrained Application Protocol (CoAP): COAP is an application layer protocol in IOT framework. It is a specialized web transfer protocol for resource constrained nodes and networks in IoT. The web transfer protocol defined by CAOP is based on *REpresentational State Transfer* (REST) on top of HTTP functionalities. It is designed for machine-to-machine (M2M) applications like home automation, smart energy etc. Its main features include are open IETF standard, compact 4 byte header, UDP,SMS support, strong DTLS security, asynchronous subscription and built in discovery.

REST enables stateless exchange of data between clients and servers over HTTP.HTTP *get*, *post*, *put*, and *delete* methods are used in REST. REST allows the clients and servers to expose and consume web services, Simple Object Access Protocol (SOAP) but in an easier way using Uniform Resource Identifiers (URIs) as nouns and HTTP *get*, *post*, *put*, and *delete* methods as verbs, using standard Internet Media Types to transfer data. REST does not require XML for message exchanges. CoAP offers a number of features that HTTP lacks, such as built-in resource discovery, IP multicast support, native push model, and asynchronous message exchange. There are many implementations of CoAP in various languages, such as libcoap1 (an open-source C-implementation) and Sensinode's NanoService2.

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

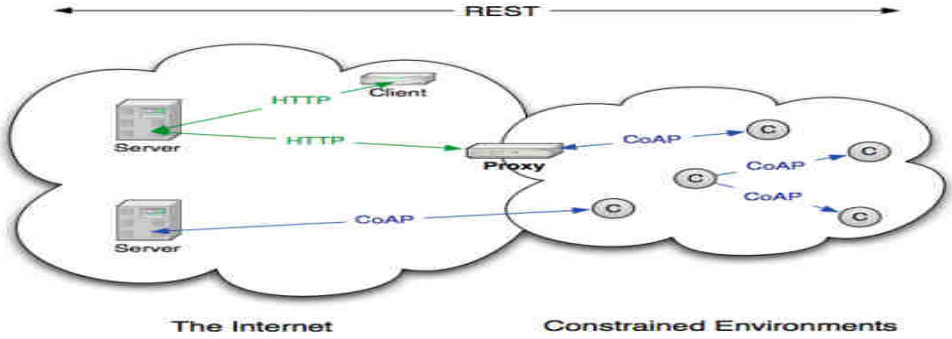

Fig.5: COAP protocol

Message Queue Telemetry Transport (MQTT)*:* MQTT, a messaging protocol standardized in 2013. MQTT aims as an optimal connection protocol for IOT nodes and M2M for low power, low memory devices and low bandwidth networks. It employs one to one, one to many and many to many routing mechanisms. It is based on publish/subscribe design pattern. MQTT is based on TCP. However MQTT-SN is defined on UDP with broker support for indexing topic names. MQTT has main components, subscriber, publisher, and broker. An IOT node registers as a subscriber for a particular topics and would be informed by the broker about the publishers publishing the topics of interest. The publisher acts as a generator of interesting data. Furthermore, the broker achieves security by checking authorization of the publishers and the subscribers. Fig 6 shows the data flow in MQTT.

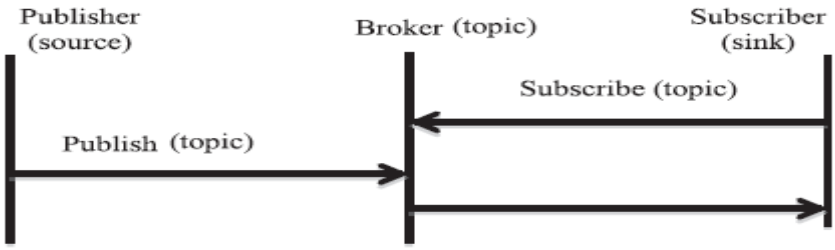

Fig.6: Data flow in MQTT

Extensible Messaging and Presence Protocol (XMPP): XMPP is instant messaging (IM) application following a decentralized fashion. It enables sending of instant messages on the Internet supporting heterogeneous operating systems making it suitable for IOT communication as well. A client communicates to a server using XMPP using a stream of XML stanzas.XML stanza has 3 main components: message, presence, and iq (info/query). Message part includes the source (destination addresses, types, and IDs of XMPP entities that utilize a push method to retrieve data.

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

Fig.7: XML stanzas

A message part includes fills the message title and contents. The presence part notifies customers as authorized. The *iq* pairs message senders and receivers. Compression is used to reduce the overhead of XML payload in XMPP.

Advanced Message Queuing Protocol (AMQP): AMQP is based on message oriented architecture based on reliable TCP. It is an open source application protocol allowing point-to –point and publish/subscribe communication. It has 2 main components: exchanges and message queues. Exchanges are employed to route the messages to appropriate queues. Queues store the messages and sent to receivers. Communication between exchanges and message queues is performed by pre-defined rules and conditions. Messages can be stored in message queues and then be sent to receivers. AMQP supports two types of messages: bare massages that are sent by the sender and annotated messages that are received at the receiver.

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

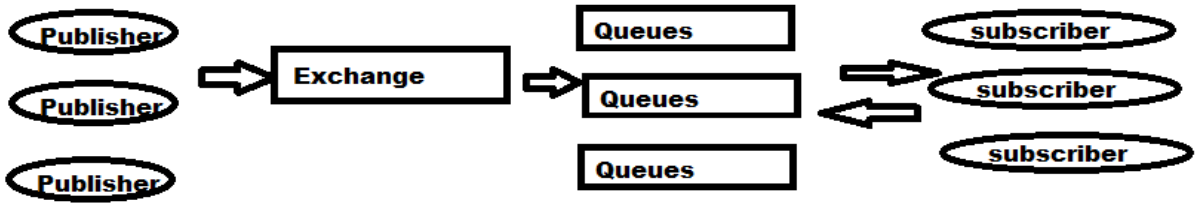

Fig.8: AMQP

Data Distribution Service (DDS): DDS developed by Object Management Group (OMG) is a publish-subscribe protocol for M2M communications. DDS is based on a broker-less architecture and uses multicasting for QoS, high reliability and real time conformance for IOT. DDS comprises of two layers: Data-Centric Publish-Subscribe (DCPS) and Data-Local Reconstruction Layer (DLRL). DCPS delivers the information to the subscribers. DLRL, an optional layer serves as the interface to the DCPS functionalities. DLRL facilitates the sharing of distributed data among distributed objects. Publisher disseminates the data, Data Writer interacts with the publisher about the values and changes of data specific to a given type. Publisher and Data Writer association provides the data in contextual form. Subscriber receives published data and delivers it to the application; Data Reader, employed by the Subscriber to access to the received data; Topic is identified by a data type and a name and relates Data Writers to Data Readers. Data transmission is performed within a DDS. Domain which is a virtual environment for connected publishing and subscribing applications.

Table.1: Comparison of application protocols in IoT

Application Protocol	**COAP**	**MQTT**	**XMPP**	**AMQP**	**DDS**	**HTTP**
REST	Yes	No	No	No	No	Yes
Transport layer Protocol	UDP	TCP	TCP	TCP	UDP	TCP
Publish/subscribe	Yes	Yes	Yes	Yes	Yes	No

The most dominant protocols for service discovery in IOT are multicast DNS (mDNS) and DNS Service Discovery (DNS-SD). DNS-SD and mDNS present a solution where no additional infrastructure, in addition to the current DNS servers, is needed, and merely requires that resources be enabled with an IP-based addressing.

*1) **Multicast DNS (mDNS):*** mDNS is an extension of Internet DNS protocol. However infrastructure requirement is eliminated and requires that resources be IP-based addressable. mDNS queries the names by sending an IP multicast message to all the IoT nodes in the local domain. When the target node receives its name, it multicasts a response message which contains its IP addresses. All nodes in the network that obtain the response message update their local cache using the given name and IP address.

*2) **DNS Service Discovery (DNS-SD):*** *A* client can discover a set of interested services in a specific network by employing standard DNS messages using DNS-SD protocol. DNS-SD utilizes mDNS to send DNS packets to specific multicast addresses through UDP. First it finds the host names of required services such as printers and pairs the IP addresses with their host names using mDNS. Pairing function multicasts network attachments details like IP, and port number to each related host.

The main drawback of these protocols is the need for caching DNS entries especially when it comes to resource-constrained devices. However, timing the cache for a specific interval and depleting it can solve this issue. Bonjour and Avahi are two well-known implementations covering both mDNS and DNS-SD.

EPC (Electronic Product Code): EPC is a unique identification number which is stored on an RFID tag. EPCglobal organization is responsible for the development of EPC, manages EPC and RFID technology and standards. The RFID system has a radio signal transponder (tag) and tag reader. The tag has a chip to store the unique identity of the object and an antenna for communication between the chip and teh tag reader using radio waves. The reader then passes that number to an *Object-Naming Services* (ONS). An ONS looks up the tag's details from a database such as when and where it was manufactured. EPCglobal Network consist of EPC, ID system, EPC Middleware, Discovery Services, and EPC Information Services. EPC assigns a unique number to objects. The ID system links the EPC identities to a database using an EPC reader through the middleware. The discovery service is a mechanism of EPCglobal to find the required data by the tags using the ONS.

Realizing web of things

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

There is a need for self-configuration protocols to enable the integration of an increasingly large number of IoT devices. These constrained devices needs to be automatically discovered, automatically assign DNS names and needs to be directly accessible over IPv6 via HTTP or CoAP. Alternatively, devices can add their resources to a publicly accessible directory service. This realizes the Web of Things (WoT). These extensions enrich the capabilities of the basic CoAP protocol and contribute to the realization of the WoT. The main issue in the inclusion of existing web service composition models to take into account the limitations of constrained devices. Cloud technology is used for collecting, storing and processing the large amount of sensor data.

IV. EDGE COMPUTING

Edge computing is the computing paradigm which enables the computation to be carried out at the edge of the network, download the data (on behalf) of cloud service and upload the data on behalf of IoT services. Edge terminology refers to the computing network resources along the path between data sources and cloud data centers. Eg: A gateway is an edge in a smart home automation, between home appliances and the cloud services. Computing should happen at the close proximity of the data producers (sources). Edge performs data offloading, storage, data caching and processing, distribute request and delivery service from cloud to user. Requirements of edge computing include: Shorter response time from cloud service providers, Shorter Response time for IoT based devices: with the proliferation of cheaper sensor devices, the data generated from IoT physical layer will be tremendous. Uploading to the cloud and getting consumed will incure larger response time. The consumers would also be present at the edge of the network. So the traditional cloud computing paradigm may be at a disadvantage, Privacy requirements at the edge of the network.

Research area in the field of edge computing includes the following

1. Application Softwares: In cloud computing, infrastructure is transparent to the user. Programs written in one language are compiled for different platforms but runs on cloud. But in edge computing, heterogeneous nodes and platform makes it difficult to develop application software to run on heterogeneous platforms. It can be solved by employing software defined networking.
2. Communication Technologies: Due to the mobility of things, adhoc network topology, privacy and security protection, scalability requirements makes the communication technologies at the edge of the network to be more efficient. Traditional naming mechanisms like DNS and IP mapping may be heavy at the edge network. So new communication technologies needs to be implemented.
3. Security: Due to the resource constrained and dynamic nature of edge network, security tools optimized in resource usage at the edge network needs to be explored. Securing the data at the edge of the network is a better option than providing the security while transmitting the data from edge to the clouds.
4. Energy efficiency: Since the edge devices are now acting as both providers and consumers, the energy utilization by these devices should be optimized for increasing the lifespan of the framework. More research needs to be done in exploring energy efficient consumption and transmission of data by the edge network, so that the advantages of the edge computing in terms of bandwidth, latency, cost can be ripped to the maximum.

V. CONCLUSION

In this paper we reviewed the various application technologies enabling IoT in the current timeframe. We came up with the need for edge computing and the paradigm shift where edge devices role will change from data consumers to data producers/consumers. Edge computing will ensure shorter response time, better bandwidth utilization, assured reliability where data will be handled at the edge rather than uploading to the cloud. We have explored the challenges and opportunities in the field of edge computing and the same will be explored in our future work.

REFERENCES

[1] D. Evans, “The Internet of things: How the next evolution of the Internet is changing everything,”CISCO, San Jose,CA, USA,White Paper, 2011.

[2] J. Gubbi, R. Buyya, S.Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[3] Z. Sheng *et al.*, “A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities,” *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013, pp. 65–76.

[4] Z. Yang *et al.*, “Study and application on the architecture and key technologies for IOT,” in *Proc. ICMT*, 2011, pp. 747–751.

https://dx.doi.org/10.22161/ijaers/3.12.29 *ISSN: 2349-6495(P) | 2456-1908(O)*

[5] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, "Research on the architecture of Internet of Things," in *Proc. 3rd ICACTE*, 2010, pp. V5-484–V5-487.

[6] L. Tan and N. Wang, "Future Internet: The Internet of Things," in *Proc. 3rd ICACTE*, 2010, pp. V5-376–V5-380.

[7] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, assumptions, problem statement, and goals," Internet Eng. Task Force (IETF), Fremont, CA, USA, RFC4919, vol. 10, Aug. 2007.

[8] K. Ashton, "That Internet of Things thing," *RFiD J.*, vol. 22, no. 7, pp. 97–114, 2009.

[9] D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," CISCO White Paper, vol. 1, pp. 1–11,2011.

[10] Z. Yang *et al.*, "Study and application on the architecture and key

[11] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: Early progress and back to the future," *Proc. IJSWIS*, vol. 8, no. 1, pp. 1–21, Jan. 2012.

[12] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, Fourth Quarter 2015,2347

[13] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu,"Edge Computing: Vision and Challenges", IEEE Internet Of Things Journal, Vol. 3, No. 5, October 2016, 637