International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217394

# Internet of things - A Survey

**Pranjal Upadhyay*[1], Prof. Deepak Upadhyay[2]**
*[1]M.E (Cyber Security), GTU, Graduate school of Engineering and Technology, Ahmedabad, Gujarat, India
[2]Assistant Professor, GTU, Graduate school of Engineering and Technology, Ahmedabad, Gujarat, India

**Article Info**
Volume 7, Issue 3
Page Number: 417-438

**Publication Issue :**
May-June-2021

**Article History**
Accepted : 25 May 2021
Published : 31 May 2021

## ABSTRACT

In the survey paper we defined all the topics related to the Internet of Things. All the components related to the internet of things in Details. You will get detailed knowledge about the Internet of things ecosystem, Internet of things Elements, Internet of things Architecture. Also, we will cover all the internet of things protocols and brief about protocols. In this we will provide the details of attack based on Protocols and at the end we justify why RPL is useful over 6Low-PAN in the internet on things network layer.

Keywords : — IoT, Internet of Things, RPL, IoT Protocols, IoT Architecture, IoT Ecosystem, IoT Attacks, Routing Attack, 6LowPAN, Rank Attack, Sybil Attack, Countermeasures

## I. INTRODUCTION

The Internet of things (IoT) describes the network of physical objects "things" that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. The Internet of Things (IoT) is envisioned to grow rapidly due the proliferation of communication technology, the availability of the devices, and computational systems. Hence, IoT security is an area of concern in order to safeguard the hardware and the networks in the IoT system. However, since the idea of networking appliances is still relatively new, security has not been considered in the production of these appliances.

Some examples of existing IoT systems are self-driving vehicles (SDV) for automated vehicular systems, microgrids for distributed energy resources systems, and Smart City Drones for surveillance systems. A microgrid system represents a good example of a cyber physical system: it links all distributed energy resources (DER) together to provide a comprehensive energy solution for a local geographical region. However, a microgrid IoT system still relies on traditional Supervisory Control and Data Acquisition (SCADA). The integration of the physical and cyber domains actually increases the exposure to attacks: cyber-attacks may target the SCADA supervisory control and paralyse the physical domain or the physical devices may be tampered or compromised, affecting the supervisory control system. On the other hand, the drone market is moving quickly to adopt automation techniques and can be integrated into firefighting, police, smart city surveillance, and emergency response. As municipalities and citizens begin to rely on such a

system, it will become critical to keep the system secure and reliable. Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways. However, the journey is far from over. We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web. We are entering an era of the "Internet of Things" (abbreviated as IoT). This term has been defined by different authors in many different ways. Let us look at two of the most popular definitions. Vermes an et al. Define the Internet of Things as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors.

Defines the Internet of ´ Things as a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object. We use these capabilities to query the state of the object and to change its state if possible. In common parlance, the Internet of Things refers to a new kind of world where almost all the devices and appliances that we use are connected to a network. We can use them collaboratively to achieve complex tasks that require a high degree of intelligence. For this intelligence and interconnection, IoT devices are equipped with embedded sensors, actuators, processors, and transceivers. IoT is not a single technology; rather it is an agglomeration of various technologies that work together in tandem. and actuators. Another definition by Pena-L ˜ opez et al.

Sensors and actuators are devices, which help in interacting with the physical environment. The data collected by the sensors has to be stored and processed intelligently in order to derive useful inferences from it. Note that we broadly define the term sensor; a mobile phone or even a microwave oven can count as a sensor as long as it provides inputs about its current state (internal state + environment). An actuator is a device that is used to effect a change in the environment such as the temperature controller of an air conditioner. The storage and processing of data can be done on the edge of the network itself or in a remote server. If any pre-processing of data is possible, then it is typically done at either the sensor or some other proximate device. The processed data is then typically sent to a remote server. The storage and processing capabilities of an IoT object are also restricted by the resources available, which are often very constrained due to limitations of size, energy, power, and computational capability. As a result the main research challenge is to ensure that we get the right kind of data at the desired level of accuracy. Along with the challenges of data collection, and handling, there are challenges in communication as well. The communication between IoT devices is mainly wireless because they are generally installed at geographically dispersed locations.

## II. Internet of Things Ecosystem

Key Elements

- Device
- Network
- Platform and
- Agent

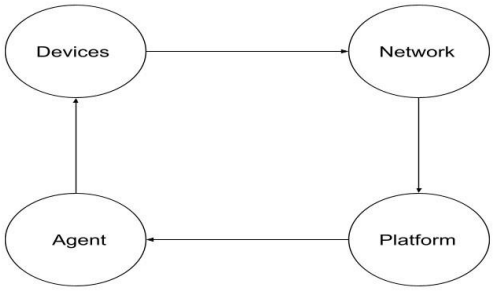


**Figure 1 1 : Key elements**

- IoT devices

As we said earlier, there are many scenarios in which IoT can be employed and they all require different

devices. Here, at the most basic level, we can speak of sensors (i.e., devices that sense things, such as temperature, motion, particles, etc.) and actuators (i.e., devices that act on things, such as switches or rotors).

Rarely, though, will a smart solution make do with just one type of an IoT sensor or an actuator. If you think of a smart surgical robot, for example, it will require hundreds, if not thousands, of components that measure different parameters and act accordingly. But even apparently less complicated solutions aren't truly that easy. Consider running a smart farm – for a plant to grow, it's not just a matter of measuring the humidity of the soil, but also its fertility; it's also a matter of providing proper irrigation based on insolation, and much more. So, you need not just one, but many sensors and actuators that all have to work together.

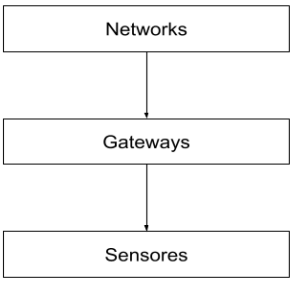


**Figure 2 1 : IoT Devices**

- Networks

Based on what you read before, you may think: "Well, if an automatic door senses my presence and opens itself, is that IoT?" Obviously, it is not, because while that door has sensors and actuators, it is not connected to much else. And, as the name suggests, the Internet of Things requires both things and the Internet (although there are cases of data delivery without the use of the Internet Protocol). Arguably, the real power of this concept lies in the connectivity. Again, based on your deployment needs, there are plenty of different IoT connectivity options, starting with the "classics," such as WIFI or Bluetooth, to more specialized and field-oriented technologies, such as Low-Power Wide Area Networks (LPWAN). They all differ in range and speed of data transfer, making them more or less appropriate for particular deployments. Consider, for example, smart cars that require both high data speed and long range and juxtapose them with the smart farms we've mentioned that don't necessarily need either.

- IoT platform

Whether they are in the cloud or not, IoT platforms are always the binder for any IoT ecosystem. They are the quiet administrators that take care of device lifecycle management, so that you don't have to worry about them. They are also the hub that collects and aggregates the data, allowing you to make sense of it. With the variety of platforms offered on the market and the breadth of claims their providers make, the choice of the "ideal" IoT platform for a deployment is arguably the most significant, yet also the most difficult to make. It shouldn't be taken lightly, as it determines whether the IoT ecosystem will thrive or wither into oblivion.

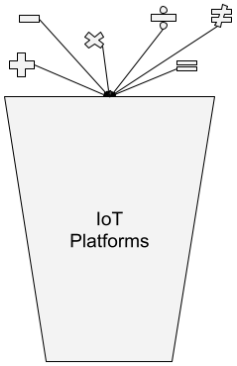


**Figure : 1 IoT Platforms**

The right IoT device management platform should be versatile and adaptable, as the IoT world is very fragmented and constantly shifting and you don't want the core element of your ecosystem to become the stumbling block of your deployment. It should also be scalable, so that your ecosystem can grow

naturally, and secure, so it can do so without any threats.

- Agents:

Agents are all the people whose actions affect the IoT ecosystem. These may be the engineers who devise IoT deployments and design the platforms, it can also be the platform operators. But probably, most importantly, it's the stakeholders, who ultimately reap the results. After all, IoT deployments aren't just art for art's sake. These complex ecosystems are put in place for a reason: to drive efficiency and improve the quality of life. And it is the agents who decide on how to use the devices, networks and platforms to achieve these results. This is where technology and business converge, because it's business goals that very much shape the IoT ecosystem.

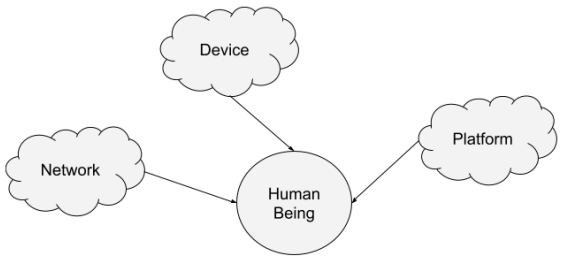


**Figure : 2 IoT Agents**

People are an essential part of this equation. Ecosystems are created by us, managed by us and, ultimately, it is our responsibility to realize their full potential. It is the devices that collect the data, but it is the people that make sense of it and put it to use. Similarly with networks and platforms, which are a necessary component of the ecosystem, but wouldn't be of much value if it weren't for the people who create and perfect them to fit their needs.

## III. Internet of Things Elements

Understanding the IoT building blocks helps to gain a better insight into the real meaning and functionality of the IoT. In the following sections we discuss Few elements needed to deliver the functionality of the IoT as illustrated in Fig. 4. Table II shows the categories of these elements and examples of each category.

- Identification

Identification is crucial for the IoT to name and match services with their demand. Many identification methods are available for the IoT such as electronic product codes (EPC) and ubiquitous codes (uCode). Furthermore, addressing the IoT objects is critical to differentiate between object ID and its address. Object ID refers to its name such as "T1" for a particular temperature sensor and object's address refers to its address within a communications network. In addition, addressing methods of IoT objects include IPv6 and IPv4. 6LoWPAN, provides a compression mechanism over IPv6 headers that makes IPv6 addressing appropriate for low power wireless networks. Distinguishing between object's identification and address is imperative since identification methods are not globally unique, so addressing assists to uniquely identify objects. In addition, objects within the network might use public IPs and not private ones. Identification methods are used to provide a clear identity for each object within the network.

- Sensing

The IoT sensing means gathering data from related objects within the network and sending it back to a data warehouse, database, or cloud. The collected data is analysed to take specific actions based on required services. The IoT sensors can be smart sensors, actuators or wearable sensing devices. For example, companies like WeMo, revolve and SmartThings offer smart hubs and mobile applications that enable people to monitor and control thousands of smart devices and appliances inside buildings using their smartphones. Single Board Computers (SBCs) integrated with sensors and

built-in TCP/IP and security functionalities are typically used to realize IoT products (e.g., Arduino Yun, Raspberry PI, Beagle Bone Black, etc.). Such devices typically connect to a central management portal to provide the required data by customers.

- Communication

The IoT communication technologies connect heterogeneous objects together to deliver specific smart services. Typically, the IoT nodes should operate using low power in the presence of lossy and noisy communication links. Examples of communication protocols used for the IoT are WiFi, Bluetooth, IEEE 802.15.4, Z-wave, and LTE-Advanced. Some specific communication technologies are also in use like RFID, Near Field Communication (NFC) and ultra-wide bandwidth (UWB). RFID is the first technology used to realize the M2M concept (RFID tag and reader). The RFID tag represents a simple chip or label attached to provide the object's identity. The RFID reader transmits a query signal to the tag and receives a reflected signal from the tag, which in turn is passed to the database. The database connects to a processing center to identify objects based on the reflected signals within a (10 cm to 200 m) range. RFID tags can be active, passive or semi-passive/active. Active tags are powered by battery while passive ones do not need battery. Semi-passive/active tags use board power when needed. The NFC protocol works at a high frequency band at 13.56 MHz and supports data rate up to 424 kbps. The applicable range is up to 10 cm where communication between active readers and passive tags or two active readers can occur. The UWB communication technology is designed to support communications within a low range coverage area using low energy and high bandwidth whose applications to connect sensors have been increased recently

**Table 1** : Simulation Platform

| Operating System | Language Support | Event based Programming | Multi-Threading |
|---|---|---|---|
| Tiny Os | nesC | Yes | Partial |
| Contiki | C | Yes | Yes |
| Lite Os | C | Yes | Yes |
| Riot Os | C/C++ | No | Yes |
| Android | Java | Yes | Yes |

- Computation

Processing units (e.g., microcontrollers, microprocessors, SOCs, FPGAs) and software applications represent the “brain” and the computational ability of the IoT. Various hardware platforms were developed to run IoT applications such as Arduino, UDOO, FriendlyARM, Intel Galileo, Raspberry PI, Gadgeteer, BeagleBone, Cubieboard, Z1, WiSense, Mulle, and T-Mote Sky.

Furthermore, many software platforms are utilized to provide IoT functionalities. Among these platforms, Operating Systems are vital since they run for the whole activation time of a device. There are several Real-Time Operating Systems (RTOS) that are good candidates for the development of RTOS-based IoT applications. For instance, the Contiki RTOS has been used widely in IoT scenarios. Contiki has a simulator called Cooja which allows researchers and developers to simulate and emulate IoT and wireless sensor network (WSN) applications. TinyOS, LiteOS and Riot OS also offer lightweight OS designed for IoT environments. Moreover, some auto industry leaders with Google established the Open Auto Alliance (OAA) and are planning to bring new features to the Android platform to accelerate the adoption of the Internet of Vehicles (IoV) paradigm.

Some features of these operating systems are compared in Table I.

Cloud Platforms form another important computational part of the IoT. These platforms provide facilities for smart objects to send their data to the cloud, for big data to be processed in real-time, and eventually for end-users to benefit from the knowledge extracted from the collected big data. There are a lot of free and commercial cloud platforms and frameworks available to host IoT services. Some of these services are introduced in Section VII-B.

## IV. Market Opportunity

The IoT offers a great market opportunity for equipment manufacturers, Internet service providers and application developers. The IoT smart objects are expected to reach 212 billion entities deployed globally by the end of 2020. By 2022, M2M traffic flows are expected to constitute up to 45% of the whole Internet traffic. Beyond these predictions, McKinsey Global Institute reported that the number of connected machines (units) has grown 300% over the last 5 years. Traffic monitoring of a cellular network in the U.S. also showed an increase of 250% for M2M traffic volume in 2011.

Economic growth of IoT-based services is also considerable for businesses. Healthcare and manufacturing applications are projected to form the biggest economic impact. Healthcare applications and related IoT-based services such as mobile health (m-Health) and telecare that enable medical wellness, prevention, diagnosis, treatment and monitoring services to be delivered efficiently through electronic media are expected to create about $1.1–$2.5 trillion in growth annually by the global economy by 2025. The whole annual economic impact caused by the IoT is estimated to be in the range of $2.7 trillion to $6.2 trillion by 2025. Fig. 2 shows the projected market share of dominant IoT applications.

On the other hand, Wikibon predicts that the value created from the industrial Internet to be about $1279 billion in 2020 with Return on Investment (ROI) growing to 149% compared to 13% in 2012. Moreover, Navigant recently reported that the Building Automation Systems (BAS) market is expected to rise from $58.1 billion in 2013 to reach $100.8 billion by 2021; a 60% increase.

All these statistics, however, point to a potentially significant and fast-pace growth of the IoT in the near future, related industries and services. This progression provides a unique opportunity for traditional equipment and appliance manufacturers to transform their products into “smart things.” Spreading the IoT and related services globally requires Internet Service Providers (ISPs) to provision their networks to provide QoS for a mix of M2M, person-to-machine (P2M) and person-to-person (P2P) traffic flows.

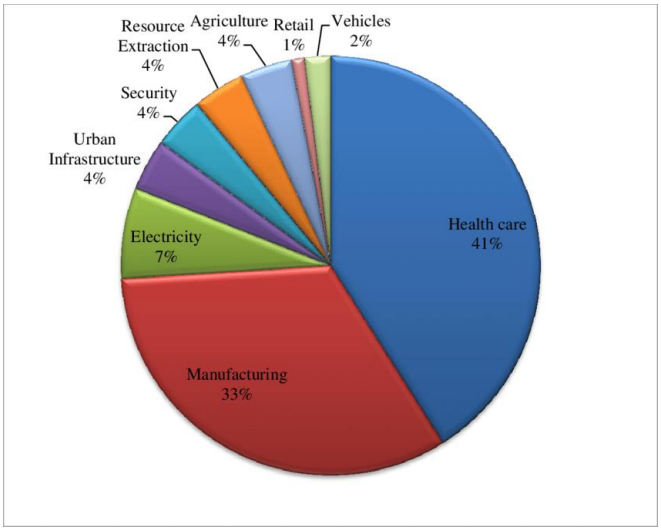


**Figure 5 1 share of dominant IoT applications**

## V. Internet of Things Architecture

There is no single consensus on architecture for IoT, which is agreed universally. Different architectures have been proposed by different researchers.

Three- and Five-Layer Architectures. The most basic architecture is a three-layer architecture as shown in Figure 1. It was introduced in the early stages of research in this area. It has three layers, namely, the perception, network, and application layers.

The perception layer is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment.

The network layer is responsible for connecting to other smart things, network devices, and servers. Its features are also used for transmitting and processing sensor data.

The application layer is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed, for example, smart homes, smart cities, and smart health.

The three-layer architecture defines the main idea of the Internet of Things, but it is not sufficient for research on IoT because research often focuses on finer aspects of the Internet of Things. That is why, we have many more layered architectures proposed in the literature. One is the five layer architecture, which additionally includes the processing and business layers. The five layers are perception, transport, processing, application, and business layers (see Figure 1). The role of the perception and application layers is the same as the architecture with three layers. We outline the function of the remaining three layers.

The transport layer transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN, Bluetooth, RFID, and NFC.

The processing layer is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules.

The business layer manages the whole IoT system, including applications, business and profit models, and users' privacy. The business layer is out of the scope of this paper. Hence, we do not discuss it further.

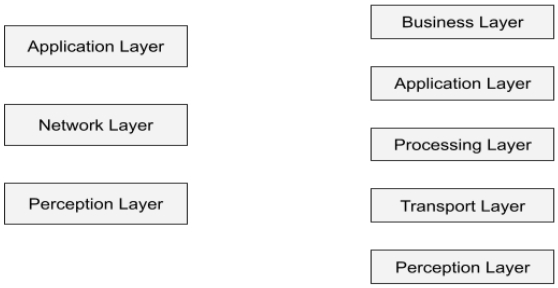


**Figure 6 IoT Architecture**

Another architecture proposed by Ning and Wang is inspired by the layers of processing in the human brain. It is inspired by the intelligence and ability of human beings to think, feel, remember, make decisions, and react to the physical environment. It is composed of three parts. First is the human brain, which is analogous to the processing and data management unit or the data center. Second is the spinal cord, which is analogous to the distributed network of data processing nodes and smart gateways. Third is the network of nerves, which corresponds to the networking components and sensors.

A 7-layer model of the IoT ecosystem. At the bottom layer is the market or application domain, which may be smart grid, connected home, or smart health, etc. The second layer consists of sensors that enable the application. Examples of such sensors are temperature sensors, humidity sensors, electric utility meters, or

cameras. The third layer consists of an interconnection layer that allows the data generated by sensors to be communicated, usually to a computing facility, data center, or a cloud. There the data is aggregated with other known data sets such as geographical data, population data, or economic data. The combined data is then analyzed using machine learning and data mining techniques. To enable such large distributed applications, we also need the latest application-level collaboration and communication software, such as, software defined networking (SDN), services-oriented architecture (SOA), etc. Finally, the top layer consists of services that enable the market and may include energy management, health management, education, transportation etc. In addition to these 7 layers that are built on the top of each other, there are security and management applications that are required for each of the layers and are, therefore, shown on the side.

**Table 2 : 7 Layer of IoT Architecture**

| | |
|---|---|
| People & Process | Layer 7 – Transformational Decision |
| Applications | Layer 6 – Costume Application |
| Data Analysis | Layer 5 – Data mining, Machine learning |
| Data Ingestion | Layer 4 – Big data |
| Global Infrastructure | Layer 3 – Cloud Infrastructure |
| Connectivity / Edge computing | Layer 2 – Protocols, Networks, Machine – 2 – Machine |
| Things | Layer 1 – Devices, Sensor's ETC… |

## VI. Cloud and Fog Based Architectures

In particular, we have been slightly vague about the nature of data generated by IoT devices, and the nature of data processing. In some system architectures the data processing is done in a large centralized fashion by cloud computers. Such a cloud centric architecture keeps the cloud at the centre, applications above it, and the network of smart things below it. Cloud computing is given primacy because it provides great flexibility and scalability. It offers services such as the core infrastructure, platform, software, and storage. Developers can provide their storage tools, software tools, data mining, and machine learning tools, and visualization tools through the cloud.

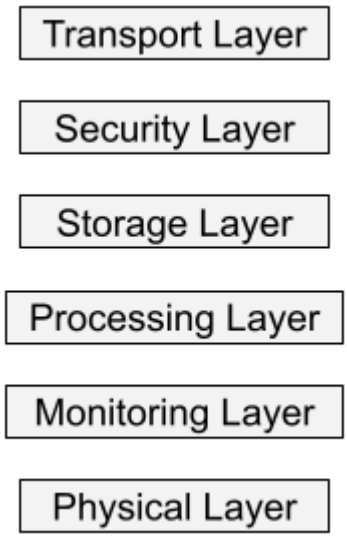


**Figure 7 : Cloud and Fog based Architecture**

Lately, there is a move towards another system architecture, namely, fog computing, where the sensors and network gateways do a part of the data processing and analytics, which inserts monitoring, pre-processing, storage, and security layers between the physical and transport layers. The monitoring layer monitors power, resources, responses, and services. The pre-processing layer performs filtering, processing, and analytics of sensor data. The temporary storage layer provides storage

functionalities such as data replication, distribution, and storage. Finally, the security layer performs encryption/decryption and ensures data integrity and privacy. Monitoring and pre-processing are done on the edge of the network before sending data to the cloud.

Often the terms "fog computing" and "edge computing" are used interchangeably. The latter term predates the former and is construed to be more generic. Fog computing originally termed by Cisco refers to smart gateways and smart sensors, whereas edge computing is slightly more penetrative in nature. This paradigm envisions adding smart data pre-processing capabilities to physical devices such as motors, pumps, or lights. The aim is to do as much pre-processing of data as possible in these devices, which are termed to be at the edge of the network. In terms of the system architecture, as a result, we do not describe edge computing separately.

Finally, the distinction between protocol architectures and system architectures is not very crisp. Often the protocols and the system are codesigned. We shall use the generic 5- layer IoT protocol stack for both the fog and cloud architectures.

## VII. Network layer in brief

In this section, we discuss some standard and non-standard protocols that are used for routing in IoT applications. It should be noted that we have partitioned the network layer in two sublayers: routing layer which handles the transfer of the packets from source to destination, and an encapsulation layer that forms the packets. Encapsulation mechanisms will be discussed in the next section.

❖ RPL

Routing Protocol for Low-Power and Lossy Networks (RPL) is distance-vector protocol that can support a variety of data link protocols, including the ones discussed in the previous section. It builds a Destination Oriented Directed Acyclic Graph (DODAG) that has only one route from each leaf node to the root in which all the traffic from the node will be routed to. At first, each node sends a DODAG Information Object (DIO) advertising itself as the root. This message is propagated in the network and the whole DODAG is gradually built. When communicating, the node sends a Destination Advertisement Object (DAO) to its parents, the DAO is propagated to the root and the root decides where to send it depending on the destination. When a new node wants to join the network, it sends a DODAG Information Solicitation (DIS) request to join the network and the root will reply back with a DAO Acknowledgement (DAO-ACK) confirming the join. RPL nodes can be stateless, which is most common, or stateful. A stateless node keeps tracks of its parents only. Only root has the complete knowledge of the entire DODAG. Hence, all communications go through the root in every case. A stateful node keeps track of its children and parents and hence when communicating inside a sub-tree of the DODAG, it does not have to go through the root.

❖ CORPL

An extension of RPL is CORPL, or cognitive RPL, which is designed for cognitive networks and uses DODAG topology generation but with two new modifications to RPL. CORPL utilizes opportunistic forwarding to forward the packet by choosing multiple forwarders (forwarder set) and coordinates between the nodes to choose the best next hop to forward the packet to. DODAG is built in the same way as RPL. Each node maintains a forwarding set instead of its parent only and updates its neighbour

with its changes using DIO messages. Based on the updated information, each node dynamically updates its neighbour priorities in order to construct the forwarder set.

❖ CARP

Channel-Aware Routing Protocol (CARP) is a distributed routing protocol designed for underwater communication. It can be used for IoT due to its lightweight packets. It considers link quality, which is computed based on historical successful data transmission gathered from neighbouring sensors, to select the forwarding nodes. There are two scenarios: network initialization and data forwarding. In network initialization, a HELLO packet is broadcasted from the sink to all other nodes in the network. In data forwarding, the packet is routed from sensor to sink in a hop- by-hop fashion. Each next hop is determined independently. The main problem with CARP is that it does not support reusability of previously collected data. In other words, if the application requires sensor data only when it changes significantly, then CARP data forwarding is not beneficial to that specific application. An enhancement of CARP was done in E-CARP by allowing the sink node to save previously received sensory data. When new data is needed, E-CARP sends a Ping packet which is replied with the data from the sensor nodes. Thus, E-CARP reduces the communication overhead drastically.

## VIII. Network Layer Encapsulation Protocols

One problem in IoT applications is that IPv6 addresses are too long and cannot fit in most IoT data link frames which are relatively much smaller. Hence, IETF is developing a set of standards to encapsulate IPv6 datagrams in different data link layer frames for use in IoT applications. In this section, we review these mechanisms briefly.

❖ 6LoWPAN

IPv6 over Low power Wireless Personal Area Network (6LoWPAN) is the first and most commonly used standard in this category. It efficiently encapsulates IPv6 long headers in IEEE802.15.4 small packets, which cannot exceed 128 bytes. The specification supports different length addresses, low bandwidth, different topologies including star or mesh, power consumption, low cost, scalable networks, mobility, unreliability and long sleep time. The standard provides header compression to reduce transmission overhead, fragmentation to meet the 128-byte maximum frame length in IEEE802.15.4, and support for multi-hop delivery. Frames in 6LoWPAN use four types of headers: No 6loWPAN header (00), Dispatch header (01), Mesh header (10) and Fragmentation header (11). In No 6loWPAN header case, any frame that does not follow 6loWPAN specifications is discarded. Dispatch header is used for multicasting and IPv6 header compressions. Mesh headers are used for broadcasting; while Fragmentation headers are used to break long IPv6 headers to fit into fragments of maximum 128-byte length.

❖ 6TiSCH

The 6TiSCH working group in IETF is developing standards to allow IPv6 to pass through Time Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4e datalinks. It defines a Channel Distribution usage matrix consisting of available frequencies in columns and time-slots available for network scheduling operations in rows. This matrix is portioned into chunks where each chunk contains time and frequencies and is globally known to all nodes in the network. The nodes within the same interference domain negotiate their scheduling so that each node gets to transmit in a chunk within its interference domain. Scheduling becomes an optimization problem where time slots are assigned to a group of

neighbouring nodes sharing the same application. The standard does not specify how the scheduling can be done and leaves that to be an application specific problem in order to allow for maximum flexibility for different IoT applications. The scheduling can be centralized or distributed depending on application or the topology used in the MAC layer.

❖ 6Lo

IPv6 over Networks of Resource-constrained Nodes (6Lo) working group in IETF is developing a set of standards on transmission of IPv6 frames on various datalinks. Although 6LowPAN and 6TiSCH, which cover IEEE 802.15.4 and IEEE 802.15.4e, were developed by different working groups, it became clear that there are many more datalinks to be covered and so 6Lo working group was formed. At the time of this writing most of the 6Lo specifications have not been finalized and are in various stages of drafts. For example, IPV6 over Bluetooth Low Energy Mesh Networks, IPv6 over IEEE 485 Master-Slave/Token Passing (MS/TP) networks, IPV6 over DECT/ULE, IPV6 over NFC, IPv6 over IEEE 802.11ah, and IPv6 over Wireless Networks for Industrial Automation Process Automation (WIA-PA) drafts are being developed to specify how to transmit IPv6 datagrams over their respective datalinks [6Lo]. Two of these 6Lo specifications "IPv6 over G.9959" and "IPv6 over Bluetooth Low Energy" have been approved as RFC and are described next.

❖ IPv6 over G.9959

RFC 7428 defines the frame format for transmitting IPv6 packets on ITU-T G.9959 networks. G.9959 defines a unique 32-bit home network identifier that is assigned by the controller and an 8- bit host identifier that is allocated for each node. An IPv6 link local address must be constructed by the link layer derived 8-bit host identifier so that it can be compressed in G.9959 frame. Furthermore, the same header compression as in 6lowPAN is used here to fit an IPv6 packet into G.9959 frames. RFC 7428 also provides a level of security by a shared network key that is used for encryption. However, applications with a higher level of security requirements need to handle their end-to-end encryption and authentication using their own higher layer security mechanisms.

❖ IPv6 over Bluetooth Low Energy

Bluetooth Low Energy is also known as Bluetooth Smart and was introduced in Bluetooth V4.0 and enhanced in V4.1. RFC 7668 [RFC7668], which specifies IPv6 over Bluetooth LE, reuses most of the 6LowPAN compression techniques. However, since the Logical Link Control and Adaptation Protocol (L2CAP) sublayer in Bluetooth already provides segmentation and reassembly of larger payloads in to 27-byte L2CAP packets, fragmentation features from 6LowPAN standards are not used. Another significant difference is that Bluetooth Low Energy does not currently support formation of multi-hop networks at the link layer. Instead, a central node acts as a router between lower-powered peripheral nodes.

**Table 3 : IoT Protocol Stack**

| |
|---|
| CoAP |
| TCP,UDP |
| IETF RPL, IETF 6LoWPAN, CORPL, 6Lo |
| IEEE 802.15.4e IEEE 802.11 - WiFi Low Power for WLAN |
| IEEE 802.15.4 |

## IX. Internet of things Protocols

IoT deals with the large amount of information, queries, data analysis paradigms and data mining processes with the help of software architectures that maintain the communication standards such as Hypertext Transfer Protocol (HTTP) and Internet Protocol (IP).

As the IoT objects are battery powered, very low power consumption is required when they are plugged into the Internet. More energy is wasted by the transmission of unnecessary data and protocol overhead. HTTP and Transmission Control Protocol (TCP) are not suitable for very less energy transmission due to the high reliability through acknowledgement of packets at higher layers. IoT has a wide range of devices such as RFID, Wireless Sensor Networks (WSNs) and has the capability to communicate as well as the objects connected to the internet such as things and machines. The primary requirements related to this ability are listed below.

Energy-efficient protocol stack: The devices of the LLN networks are battery powered and frequently installed in the areas with no human intervention that makes the frequent battery replacement impossible. The protocol stack must make use of very less energy.

Internet-facilitated protocol stack: The machines in the Internet network use IP as an universal protocol. Since the LLNs are internet connected to make the IoT a realistic, the LLN devices should have a common communication language.

Highly reliable protocol stack: There may be a loss due to link failure in LLN networks. All the layers of the stack must be guaranteed for reliability.

Low power radio technology IEEE 802.15.4-2006 is the well-known standard for the physical (PHY) layer which would meet the energy efficiency requirements of LLN devices. This standard operates on the worldwide unlicensed frequency band of 2.4-2.485 GHz (ISM Band).

The medium access control (MAC) layer adopts the newly developed IEEE 802.15.4e. The important features of this protocol time-synchronized channel hopping to combat fading and interference. IEEE 802.11 - WiFi Low Power for WLAN is the standard which will also be part of the MAC layer that assures high energy efficiency and integrates the existing infrastructure with integrated IP compatibility.

The network layer holds the 6LoWPAN protocol which has the responsibility of connecting the LLN devices to the Internet.6LoWPAN connects the Internet with the devices in the LLNs through the IPv6 capabilities such as encapsulation and header compression that allows the IPv6 packets to be transmitted over low-power link layer technologies. The routing issues are very challenging in the case of LLN devices. IETF's RPL protocol is capable of building the routes quickly and transmits the routing information among the nodes with minimum overhead. Adapting to the topological changes is an additional property of the RPL protocol so that it is applied in a wide range of IoT networks such as smart home, smart healthcare and smart grids.

The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are adopted by the Transport Layer. The Constrained Application Protocol (CoAP) developed by IETF will be utilized by the application layer that implements the interoperability with HTTP for simple integration.

## X. Routing protocol In IoT

The Internet Engineering Task Force (IETF) created working groups (WGs) which developed various IoT protocols for IoT devices. We discuss below the

routing protocols which have been developed by IETF for the Internet of Things (IoT).

**Table 4 : Layer wise IoT Protocols**

| Layer | Protocols |
|---|---|
| Application Layer | CoAP |
| Transport layer | TCP, UDP |
| Network Layer | IETF RPL, IETF 6LoWPAN |
| MAC Layer | IEEE 802.15.4e IEEE 802.11 - WiFi Low Power for WLAN |
| Physical Layer | IEEE 802.15.4 |

## XI. Routing in IPv6 over low power wireless personal area networks (6LoWPAN)

6LoWPAN is an IETF-standardized IPv6 adaptation layer (data link and cross-layer protocol) that enables IP connectivity over low power and lossy networks. This is observed as the basis for the network build up for the Internet of Things such as smart homes, smart cities and industrial control systems. A large number of applications utilize 6LoWPAN for IP-based communication through an upper layer protocol such as the RPL routing protocol. 6LoWPAN essentially adjusts IPv6 packets into frames of 127 bytes, a frame size requirement that low power sensor devices can utilize among themselves. 6LoWPAN supports the transmission of large-sized IPv6 packets on the data link layer of the IEEE 802.15.4. It further provides fragmentation support at the adaptation layer involving processes such as buffering, forwarding and processing of fragmented packets which are expensive on these already resource constrained devices. Rogue nodes can send stale, overlapping or duplicate fragments to disrupt the network. At this layer there is no authentication, so the receiving nodes are debilitated in differentiating between legitimate and spurious packets during fragment reassembly. Usually the receiving nodes store up the fragments received in order to re-assemble them. If the entire set of frames making up the packet are not received after a certain timeout they are discarded. This system could also be exploited by malicious nodes which could send false fragments to fill up the nodes store, so it does not receive the legitimate fragments for re-assembly. This is indeed a challenging security issue in IoT networks. However, some protocols which have adopted 6LoWPAN (Winter et al., 2012; Hui and Thubert, 2011; Shelby et al., 2012) hinge on the security sublayer of the 802.15.4 to prevent 802.15.4 frames introduced by malicious nodes. Indeed the 802.15.4 security sublayer actively achieves this aim by adding to every frame a Message Integrity Code (MIC) and a frame counter. Once a node has been compromised the attacker could easily inject spurious frames into the network and thus, add other non-authorized nodes into the victim's network. This error and security loophole could be propagated even to the upper layer of protocols since the upper layer protocols rely on the 802.15.4 security sublayer for the security of frames.

## XII. Routing protocol for low-power and lossy networks

RPL was developed by the IETF working group as routing functionalities in 6LoWPAN were very challenging due to the resource constrained nature of the nodes. RPL operates at the network layer making it capable to quickly build up routes and distribute route information among other nodes in an efficient manner. RPL is a Distance Vector IPv6routing protocol for LLNs, thus network path information is organized as a set of Directed Acyclic Graphs (DAGs) and this is further classified as a set of Destination

Oriented Directed Acyclic Graphs (DODAG). A DODAG typically consist of sensor nodes and a sink node which collects data from these nodes as shown in Fig.1.Every DODAG is distinguished by four factors which include: DODAG ID, DODAG version number, RPL instance ID and Rank while every DODAG sink is linked with each other (Winter et al.,2012). Route selection in RPL depends on the DODAG link, cost of information to a node such as workload, throughput, node power, latency or reliability. To produce a route topology, every node selects a set of parents that comprises nodes with equal or better paths towards the sink. The node with the best route link is chosen as the parent. RPL employs three types of control messages in order to form and manage routing of information in the network and these are: i. DODAG Information Object (DIO), used for setting and updating the network topology. ii. DODAG Advertisement Object (DAO) used for broadcasting and advertising destination information upwards during network route updates.iii. DODAG Information Solicitation (DIS) is used when a new node seeks topology information while waiting to join the network.DAO and DIS are involved during a topology change process while the DIO message is broadcast and mainly used for the purpose of starting a topology change process. DIO is commonly used to distribute its routing state to other nodes using its rank (rank specifies the link quality to a sink node) and objective function. Every node computes its rank according to the rank of its selected parent and the objective function. A DIO message is sent to all nodes every time a node updates its rank or preferred parent. To prevent the formation of loops, RPL utilizes the rank rule whereby a node in a parent should always have a lower rank than its children. Also, to limit the amount of broadcast, RPL uses the trickle algorithm for scheduling DIO messages to be sent. It does this by setting a counter which observes the network topology and thereby decides when a node has to send a DIO message. For every DIO message received without comparing it with the previous DIO message this will cause the DIO counter to increase and if the DIO counter reaches a threshold value (redundancy value) the node will reset its DIO counter and double the trickle time. This is done to stabilize the network topology over a period of time and avoid the unnecessary frequent route updates which could consume the limited power and bandwidth available. This further helps to limit the number of DIOs produced so as to preserve scarce network resources. For incoming traffic, the node resets its DIO to zero and reduces its trigger time. This gives the opportunity for quick network route update through a rapid DIO generation.The RPL routing protocol has capacity to incorporate different types of traffic and signaling information swapped among nodes although this depends on the requirements of the considered data flows. RPL supports the Multipoint-to-Point (MP2P), Point-to-Multipoint (P2MP) and Point-to-Point (P2P) traffics.

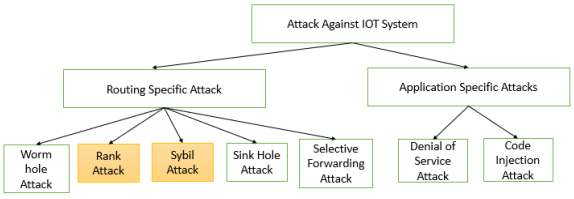


**Figure 8: Attacks on IoT**

## XIII. Attacks on RPL

This one takes into account the goals of the attack and what element of the RPL network is impacted. The taxonomy is depicted in Figure 2 and considers three categories of security attacks. In this paper we have broadly classified the routing attacks in IoT networks in three categories. These are i). Attacks on Network Resources: These include attacks targeting the exhaustion of network resources (energy, memory and power). These attacks are particularly damaging for such constrained networks because

they greatly shorten the lifetime of the devices and thus the lifetime of the RPL network. ii). Attacks on Network Topology: These cover attacks aiming at disrupting the RPL network topology. The attackers herein either aim at sub-optimization of the network topology or isolating a set of RPL nodes from the network. iii). Attacks on Network Traffic: This category corresponds to attacks against the network traffic, such as spoofing attacks or deception attacks.

## XIV. Attacks on Network Resources

Attacks against resources aim at making legitimate nodes perform unnecessary processing in order to exhaust their resources. This eventually intends at consuming node energy, memory or processing. This may impact on the availability of the network by congesting available links and therefore on the lifetime of the network which can be significantly shortened. We further classify it into two subcategories of attacks against resources. The first one is direct attacks where a malicious node will directly generate the overload in order to degrade the network. The second one is indirect attacks where the attackers will make other nodes generate a large amount of traffic. Indirect attacks could be an attack that may create loops in the RPL network which in turn make other nodes produce traffic overhead.

- Direct Attacks

In case of direct attacks, the attacker is directly responsible for resource exhaustion. This can typically be done by performing flooding attacks or by executing overloading attacks with respect to routing tables, when the storing mode is active. Hello Flooding Attacks: For joining the network node broadcast the initial message as HELLO message. Attackers can introduce themselves as a neighbour node to many nodes by broadcasting Hello messages with strong routing metrics and entering the network. In RPL, DIO messages are referred to as Hello messages, which is used to advertise information about DODAG. This attack can be mitigated by using the link-layer metric as a parameter in the selection of the default route. If it fails to receive link-layer acknowledgements then a different route is chosen. Another solution can be by using the geographical distance, nodes should not select the nodes which are beyond their transmission range. This attack cannot exist for a long time in the RPL network, as RPL's Global and Local repair mechanism removes this attack. If this attack combines with the other attacks, then RPL's Global and Local repair mechanism does not remove it.

- Routing Table Overload Attacks in Storing Mode

It is also possible to perform direct attacks against resources by overloading the RPL routing tables. The RPL protocol is a proactive protocol. This means that the RPL router nodes build and maintain routing tables when the storing mode is enabled for those nodes. The principle of routing table overload is to announce fake routes using the DAO messages which saturate the routing table of the targeted node. This saturation prevents the build of new legitimate routes and impacts network functioning. It may also result in a memory overflow. Let us consider the example of the DODAG 2 graph described in Figure 3 and assume that node 12 plays the role of the attacker. Nodes 12 and 13 send a DAO message in order to add the corresponding entries in the routing table of node 11. The attacker, node 12 sends multiple forged DAO messages to node 11 with false destinations. As a consequence, node 11 builds all the corresponding entries in its routing table. Afterwards, when the other nodes including node 13 are sending legitimate DAO messages with respect to new routes, the node 11 is no longer able to record them because its routing table is overloaded. This attack is not specifically mentioned in the literature but it is part of overload attacks more generally.

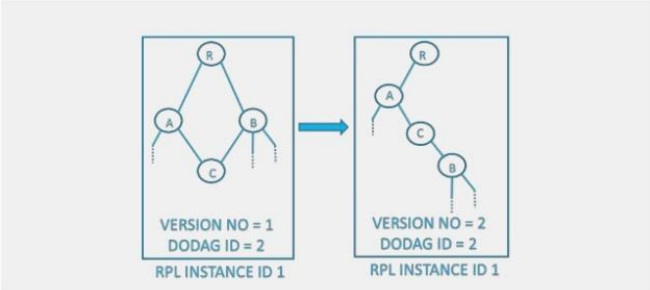


**Figure 9 1: Routing Table overload**

- DIS Attack

DIS (DODAG Information Solicitation) message used by new nodes to get the topology information before joining the RPL network. In this attack, malicious nodes periodically send the DIS messages to its neighbours. When the DIS messages are broadcast by the attacker, the receiver nodes upon receiving DIS messages reset the DIO timer assuming something went wrong with the topology around it. When an attacker unicast the DIS message the receiver node in return sends the DIO message indicating that the sender is willing to join the network. Both ways of sending DIS message adds the consequences in the network as no impact on delivery ratio but DIS multicast attack showed the most increase in end-to-end delay. This attack helps to generate more control overhead and eventually results in energy exhausting.

- Local Repair Attacks

In local repair attacks, attackers without any problem with link quality periodically send the local repair message. This causes the local repair around the nodes which hears the local repair message. Local repair attack creates more impact on delivery ratio than any other kind of attack, generates more control packets and increases the end-to-end delay. Also exhaust the energy of nodes unnecessarily.

- Indirect Attacks

Indirect attacks correspond to attacks where the malicious node makes other nodes generate an overload for the network. It includes: increased rank attacks, DAG inconsistency attacks and version number attacks. Increased Rank Attacks: In RPL rank value increases from root to child node. By changing Rank value, an attacker can attract child nodes for selecting as parents or improve some other metric, and can attract large traffic going toward the root. The variation of rank attack based on the attack existing duration (continuous or discontinuous) and update or no update of DIO information into four types and evaluated in the RPL environment against network QOS parameters. The increased rank attack consists in voluntarily increasing the rank value of a RPL node in order to generate loops in the network. This attack has been studied through ns-2 simulations. The authors showed that their loop avoidance mechanisms cost more than the attack itself. Concretely, in a RPL network, a rank value is associated to each node and corresponds to its position in the graph structure according to the root node. As previously mentioned, the node rank is always increasing in the downward direction in order to preserve the acyclic structure of the DODAG. When a node determines its rank value, this one must be greater than the rank values of its parents. If a node wants to change its rank value, it has to first update its parents list by removing the nodes having a higher rank than its new rank value. Once a node has established the set of parents in a DODAG, it selects its preferred parent from this list in order to optimize the routing cost when transmitting a packet to the root node. A malicious node advertises a higher rank value than the one it is supposed to have. Loops are formed when its new preferred parent is in its prior sub DODAG and only if the attacker does not use loop avoidance mechanisms. In the first scenario, the attacker is node 13 and the new preferred parent (node 24) has already a substitute parent (node 12) to

re-attach to. The node 13 increases its rank value to 3 and chooses node 24 as the new preferred parent. This operation generates a routing loop in the DODAG graph, because the node 24 was in the prior sub-DODAG of node 13. The formed loop is composed of nodes 13 and 24 and is easily repaired because the node 24 can re-attach to node 12 after sending a few control messages. However, this attack becomes more problematic when the node does not have a substitute parent such as node 31 in the second scenario. As depicted in Figure 1, the attacker increases its rank value which requires node 31 to also increase its own in order to find a new parent. Meanwhile nodes 32 and 33 have to connect to a substitute parent (node 22) so node 31 selects node 32 as the new preferred parent. At the end, node 21 increases its rank value to 5 in order to add node 31 as its preferred parent. The count-to-infinity problem is avoided because of the limitation of the maximum rank value advertised for a DODAG. The increased rank attack is more damaging in this second scenario, because more routing loops are built in the neighbourhood. In that case, the loop repair mechanism requires sending many DIO messages (resets of the trickle timer) and requires a longer convergence time. The more the number of affected nodes increases, the longer the convergence time is. We consider this attack as part of the resource consumption attacks because the churn is exhausting node batteries and is congesting the RPL network. To mitigate this attack, the number of times a RPL node is increasing its rank value in the DODAG graph should be monitored to determine if a node can be considered as malicious or misconfigured. It is important to notice that a node can legitimately increase its rank value if it no longer matches the objective function and/or cannot manage the amount of received traffic. However, it must use the loop prevention techniques or it can wait for a new version of the DODAG graph.

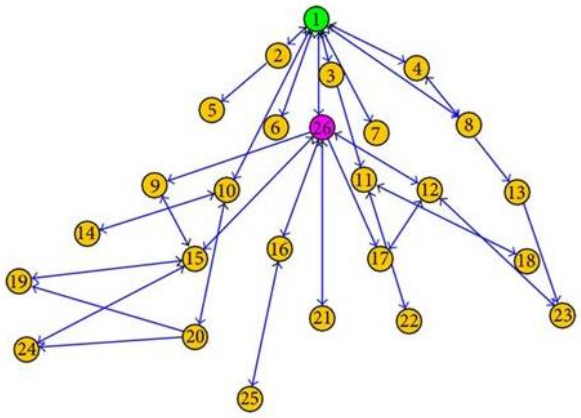


**Figure 10 1: Indirect Attack**

➢ DAG Inconsistency Attacks

A RPL node detects a DAG inconsistency when it receives a packet with a Down 'O' bit set from a node with a higher rank and vice-versa e.g. when the direction of the packet does not match the rank relationship. This can be the result of a loop in the graph. The Rank-Error 'R' bit tag is used to control this problem. When an inconsistency is detected by a node, two scenarios are possible: (i) if the Rank-Error tag is not set, the node sets it and the packet is forwarded. Only one inconsistency along the path is not considered as a critical situation for the RPL network, (ii) if the 'R' bit is already set, the node discards the packet and the timer is reset. As a consequence, control messages are sent more frequently. A malicious node has just to modify the tags or add new tags to the header. The immediate outcome of this attack is to force the reset of the DIO trickle timer of the targeted node. In that case, this node starts to transmit DIO messages more frequently, producing local instability in the RPL network. This also consumes the battery of the nodes and impacts the availability of links. All the neighbourhood of the attacker is concerned by the attack, since it has to process unnecessary traffic. Moreover, by modifying legitimate traffic, all the packets are discarded by the

targeted node. This causes a blackhole and isolates segments of the network. To mitigate the flooding induced by this attack, proposes to limit the rate of trickle timer resets due to an RPL Option to no greater than 20 resets per hour. In, authors have proposed two solutions that take into account network characteristics by using adaptive threshold and node's specific parameters respectively.

- Version Attack

This attack takes place by publishing the higher version number of the DODAG tree. When nodes receive the new higher version number DIO message, they start the formation of a new DODAG tree. This can cause the generation of new unoptimized topology and brings inconsistencies in topology. The loops and rank inconsistencies created by the attack are generally located around the neighbourhood of the attacker. VeRA schema prevents this attack by providing verification to version number using digital signature and MAC. The attack increases control overhead 18 times, impacts energy consumption and channel availability. It also reduces the delivery ratio of packets by up to 30% and nearly doubles the end-to-end delay in a network. An attacker located at a large distance from the root causes the highest increase in overhead, and the higher packet loss. Denial of Service Attack: Denial of service or Distributed denial of service attack is an attempt to make resources unavailable to its intended user. In RPL this attack can be brought using the IPv6 UDP packet flooding. Many malicious nodes by coordinating can bring the Distributed denial of service attack, wherein it is difficult to identify the malicious nodes. However, the IDS system in proposed the framework for detection of DOS attack in 6LoWPAN. The architecture integrates the IDS into the network framework developed within the EU FP7 project ebbits. A security layer of ebbits Dos protection module is added. IDS probe nodes located in the network which sends periodically the traffic in 6LoWPAN through wired connection to the IDS system. The Dos protection manager receives the alerts from the IDS system. It takes the network related information from other modules of the network manager layer to confirm the attack. IDS sends the jamming information of the attack to Dos protection manager. The presence of jamming information at the modules of network manager of ebbits indicated the presence of attack.

## XV. Attacks on Network Topology

Attacks against the RPL protocol can also target network topology. We distinguish two main categories amongst these attacks: sub-optimization and isolation.

- Selective Forwarding Attack

This attack takes place by selectively forwarding packets. With these attacks DoS (Denial of Service) attacks can be launched. The purpose of the attack is to disrupt routing paths and filter any protocol. The RPL attacker could forward all RPL control messages and drop the rest of the traffic. Solution to this attack can be creating a disjoint path or dynamic path between parent and children. Another solution is by using encryption techniques in which the attacker will not be able to identify the traffic flow. Heartbeat protocol is basically used for detection of the disruption in network topology but also can be used as a defense against selective forwarding attack. IDS solution given the End-to-End packet loss adaptation algorithm for detection of selective forwarding attack. Such attacks need to be detected and removed, RPL self-healing does not correct the topology. Routing Table Poisoning Attacks in Storing Mode: In a routing protocol, it is possible to forge or modify routing information to advertise falsified routes to other nodes. This attack can be performed in the RPL network by modifying or forging DAO control messages in order to build fake downward routes.

This can only be done when the storing mode is enabled. For instance, a malicious node advertises routes toward nodes that are not in its sub-DODAG. Targeted nodes have the wrong routes in their routing table causing network sub- optimization. As a result, the path can be longer inducing delay, packet drops or network congestion.

- Sinkhole Attack

In sinkhole attacks attacker node advertises beneficial path to attract many nearby nodes to route traffic through it. This attack does not disrupt the network operation but it can become very powerful when combined with another attacks. The IDS system gives the solution to detect this attack. To defend against sinkhole attack evaluated parent failover and a rank authentication technique. The rank authentication technique relies on one way hash technique. The root begins to generate hash value by picking random value, and broadcast it in DIO message. All nodes calculate the hash value using previous received one and again broadcast it using DIO message. Assumed that malicious node doesn't calculate the hash value, it simply broadcast received DIO message. Each node stores the hash value received by its parent along with number of hops in the path. When root node broadcast random number securely, then node can verify its parent rank using that intermediates hops number. Parent fail-over technique uses UNS (unheard nodes set) field in DIO message indicating that the nodes are in sinkhole compromised path. If the node receives the DIO message containing its ID in UNS then it adds its parent in black list. RPL does not have the self-healing capacity against the sinkhole.

- Wormhole Attack

RPL can undergo the wormhole attack. The main purpose of this attack is Disrupt the network topology and traffic flow. This attack can take place by creating a tunnel between the two attackers and transmitting the selective traffic through it. Wormhole attack can be prevented using the construction of Markle tree authentication. In RPL the tree construction starts from root to leaf nodes and Markle tree construction starts from leaf node to root. It uses the ID of node and public key for calculation of hash. Each parent is identified by its children. Authentication of any node begins with the root node up to the node itself. If any node fails to authenticate, then children nodes avoid the wrong parent selection.

- Decreased Rank Attacks

In a DODAG graph, the lower the rank is, the closer the node is to the root and the more traffic this node has to manage. When a malicious node illegitimately advertises a lower rank value, it over claims its performance. As a result, many legitimate nodes connect to the DODAG graph via the attacker. The malicious node is capable of performing other attacks such as sinkhole and eavesdropping attacks. In the RPL protocol, an attacker can change its rank value through the falsification of DIO messages. The VeRa solution as well as the Rank verification method is able to address this issue. However, authors have shown that VeRa is not sure regarding rank authentication and they proposed improvements to address this issue called TRAIL. They also showed another way to perform this attack by replaying the rank of the attacker's parent which allows it to decrease its rank by one. Since SVELTE can detect sinkhole attacks it can also detect the decreased rank attack. Identity Attacks: Identity attacks gather both spoofing and sybil attacks. In a clone ID attack, an attacker copies the identities of a valid node onto another physical node. This can, for example, be used in order to gain access to a larger part of the network or in order to overcome voting schemes. In a sybil attack, which is similar to a clone ID attack, an attacker uses several logical entities on the same

physical node. Sybil attacks can be used to take control over large parts of a network without deploying physical nodes. By keeping track of the number of instances of each identity it is possible to detect cloned identities. It would also be possible to detect cloned identities by knowing the geographical location of the nodes, as no identity should be able to be at several places at the same time. The location of nodes or similar information could be stored either centralized in the 6BR or distributed throughout the network in a distributed hash table (DHT). In an IP/RPL network cloned identities will cause trouble when packets are heading to one of the cloned identities. Packets will be forwarded to one of the cloned identities based on the routing metrics in the network, and the rest of the cloned identities will be unreachable from certain nodes in the network. This however does not affect the network otherwise, and therefore cloned identities on their own, do not cause harm on a 6LoWPAN network.

## XVI. SUMMARY OF ATTACKS ON IOT

| Attack | Effect on network parameters | Method to counter measure |
|---|---|---|
| Sinkhole | Large traffic flows through attacker node | IDS solution, parent fail-over, rank authentication technique |
| Wormhole | Disrupt the network topology and traffic flow | Markle tree authentication |
| Sybil | Routing traffic unreachable to victim node | No technique evaluated yet |
| Denial Of Service | Make resources unavailable to Intended user | IDS based solution |
| Blackhole | Packet delay and control overhead | No technique evaluated yet |
| Rank | Packet delay, delivery ratio and generation of Un-optimised path and loop | IDS based solutions, VeRA, TRAIL |

**Figure 11 1 : Summary of attack in IoT**

- Countermeasures of RPL

In this section, we investigate possible security solutions for the aforementioned threats.
The ideal solution is the prevention of the possible threats; however, the specific goal is nearly impracticable, but appropriate countermeasures can mitigate the impact of these threats.

- 6LoWPAN Security

Utilizing the IEEE 802.15.4 protocol at the PHY and the MAC sublayers, the Low Power Wireless Personal Area Networks (WPANs) can use only 102 bytes for the transmission of information at next communication layers. However, the value of the Maximum Transmission Unit (MTU) that is needed for the IPv6 requirements is equivalent to 1280 bytes which is considerably higher than the previous number. The purpose of the IPv6 low power WPAN (6LoWPAN) standard is to solve this complication by deploying the interconnection between the IEEE 802.15.4 and IPv6 protocols for WPANs. In particular, it operates as an adaptation layer that utilizes compression, fragmentation and encapsulation mechanisms and transmits the modified IPv6 packets at the MAC sublayer.

Currently, 6LoWPAN standard does not provide any security mechanism, such as IPSec due to the limitations of IoT devices. However, individual research proposals examine possible solutions to address these constraints, designing compressed security headers for the 6LoWPAN adaptation layer which have the same purpose as the existing Encapsulating Security Payload (ESP) and Authentication Header (AH) of IPSec. Also, some studies consider the incorporation of specific mechanisms in the 6LoWPAN against fragmentation attacks. More specifically, the authors discuss the addendum of a timestamp and a nonce field to the 6LoWPAN fragmentation header in order to address such attacks. In addition, proposes the use of mechanisms that can support the pre fragment sender authentication and prevent messages that are considered as suspicious. Finally, a significant security addition to the 6LoWPAN standard is the key management, as the keys must be regularly renewed in order to assure the principles of confidentiality, integrity and authenticity. For instance, the Internet Key Exchange version 2 (IKEv2)

protocol could be adopted, which is appropriate for use in devices with constrained resources. Therefore, as a result, the lack of security mechanisms in the 6LoWPAN standard offer research opportunities for improvements in future versions.

- RPL Security

The RPL protocol was created by the Internet Engineering Task Force (IETF) and is appropriate to route messages in Low Power and Lossy Networks (LLNs). Its operation is based on the creation of a Destination Oriented Directed Acyclic Graph (DODAG) that utilizes an objective function. In more detail, the DODAG consists of a set of nodes, which possess oriented edges in order not to create loops. The creation of a DODAG starts when the root node transmits a DIO message to their neighbours. The neighbouring nodes receive the DIO message and take the decision whether they join in the graph. If a node joins the graph, then the corresponding path to the root node is created. Then, using the objective function, the new node of the graph calculates a value which is called rank. This procedure is repeated for each node in the graph. Finally, it is worth mentioning that the nodes have the ability to transmit a DODAG Information Solicitation (DIS) message in order to discover new DODAGs and as well as they can send DODAG Destination Advertisement Object (DAO) messages to advertise a routing path.

The security in the RPL protocol is based on the existence of secure variations of the RPL packets (DIS, DIO, DAO, DAO-ACK) and also the capability to apply three security modes. These variations provide integrity, replay protection, delay protection and optional confidentiality. Specifically, the cryptographic algorithms and the overall security strategy are identified by the Security field that is analysed further in the following subfields.

## XVII. Why RPL is used Instead of 6LoWPAN

- RPL is a lightweight, rank based routing protocol.
- RPL is the routing protocol developed specifically for low power and lossy networks, in which nodes and routers are expected to be power-constrained.
- So it is made to measure for much of what people have come to believe is (or will be) the Internet of Things.
- RPL runs in power-constrained nodes, it is a reactive protocol. Which means, routes are found when they are needed, rather than routing tables being maintained over time.
- Supporting wifi, 802.15.4, Lora and more in Contiki OS enabled by RPL.
- like signaling overhead, PDR, latency and energy utilization.
- RPL is a well-suited protocol for LLNs.

## XVIII. CONCLUSION

In the survey paper we defined all the topics related to the Internet of Things. All the components related to the internet of things in Details. You will get detailed knowledge about the Internet of things ecosystem, Internet of things Elements, Internet of things Architecture, Protocols, Layer wise protocols, Attacks. Also, we will cover all the internet of things protocols and brief about protocols. In this we will provide the details of attack based on Protocols and at the end we justify why RPL is useful over 6LowPAN in the internet on things network layer.

## XIX. REFERENCES

[1]. Philokypros P. Ioulianou, Vassilios G. Vassilakis, Ioannis D. Moscholios, Michael D. Logothetis “A Signature-based Intrusion Detection System for

the Internet of Things ", White Rose Research online , 2018.

[2]. Abdul Rehman, Meer Muhammad khan, M. Ali Lodhi , Faisal Bashir Hussain "Rank Attack using Objective Function in RPL for Low Power and Lossy Networks", IEEE Internet of Things Journal, May2016.

[3]. T. Winter, P. Thubert, A. Brandt, "RPL: IPv6 Routing Protocol for LowPower and Lossy Networks", Internet Engineering Task Force,2012.

[4]. Ali Alharbi , Mohamed Zohdy , Debatosh Debnath , Richard Olawoyin and George Corser, "Sybil Attacks and Defenses in Internet of Things and Mobile Social Networks", IJCSI, 2018.

[5]. Linus Wallgren, Shahid Raza, and Thiemo Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things", International Journal of Distributed Sensor Networks, 2013.

[6]. P. Levis ,T. Clausen, J. Hui, O. Gnawali, J. Ko, "The Trickle Algorithm", Internet Engineering Task Force (IETF), 2011.

[7]. Usman Shafique, Abid Khan, Abdur Rehman, Faisal Bashir, Masoom Alam, "Detection of rank attack in routing protocol for Low Power and Lossy Networks", Springer, 2018.

[8]. Kuan Zhang, Xiaohui Liang,Rongxing Lu and Xuemin Shen, "Sybil Attacks and Their Defenses in the Internet of Things", IEEE, 2014.

**Cite this article as :**

Pranjal Upadhyay, Prof. Deepak Upadhyay, "Internet of things - A Survey", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 3, pp. 417-438, May-June 2021. Available at doi : https://doi.org/10.32628/CSEIT217394
Journal URL : https://ijsrcseit.com/CSEIT217394