Different Protocols in IoT

1. Infrastructure Protocols

1. **Internet protocol version 4 (IPv4)**
2. **Internet protocol version 6 (IPv6)**
3. LOADng
4. **RPL (Routing Protocol for Low-Power and Lossy Networks)**
5. 6LoWPAN
6. QUIC
7. Micro internet protocol (uIP)
8. Nano internet protocol (nanoIP)
9. Content-centric networking (CCN)

2. Discovery Protocols

1. Physical web
2. Multicast DNS (mDNS)
3. Universal plug and play (UPnP)

3. Data Protocols

1. **MQTT**
2. **MQTT-SN**
3. **CoAP**
4. AMQP
5. **XMPP**
6. SOAP
7. REST
8. WebSocket

4. Identification Protocols

1. EPC
2. uCode
3. URIs

5.Device Management

1. **TR-069**
2. **OMA-DM**

6. Semantic Protocols

1. JSON-LD
2. Web thing model

1. Infrastructure Protocols:

IPv4: The IPv4 header packet shown in Figure has 13 distinct fields, the functions of which are given as follows.

• **VER**: It is 4 bits long and represents the version of IP. It is 4 bits (binary: 0100).

• **HLEN**: It is 4 bits long and denotes the length of the IPv4 packet header.

• **ToS**: It is 8 bits long. The first six most significant bits represent the differentiated services code point (DSCP) to be provided to this packet (by the routers). Explicit congestion notification (ECN), which gives information about the congestion witnessed in the network, is handled by the last 2 bits.

• **TOTAL LENGTH**: It is 16 bits long and identifies the length of the entire IPv4 packet, including the header and the payload.
• **IDENTIFIER**: It is 16 bits long and used to identify the original packets in case of packet fragmentation along the network.
• **FLAGS**: It is a 3-bit field with the most significant bit always set to 0. FLAGS indicates whether a packet can be fragmented or not in case the packet is too big for the network resources.
• **FRAGMENT OFFSET**: It identifies the exact offset or fragment position of the original IP packet and is 13 bits long.
• **TTL**: It is 8 bits long and prevents a packet from looping infinitely in the network. As it completes a link, its value is decremented by one.
• **PROTOCOL**: It is 8 bits long. This field identifies the protocol of the packet as user datagram protocol, UDP (17), transmission control protocol, TCP (6), or Internet control message protocol, ICMP (1). The identification is made at the network layer of the destination host.
• **HEADER CHECKSUM**: It is 16 bits long and used for identifying whether a packet is error-free or not.
• **SOURCE ADDRESS**: It indicates the origin address of the packet and is 32 bits long.
• **DESTINATION ADDRESS**: It indicates the destination address of the packet and is 32 bits long.
• **OPTIONS and PADDING**: It is an optional field, which may carry values for security, time stamps, route records, and others.

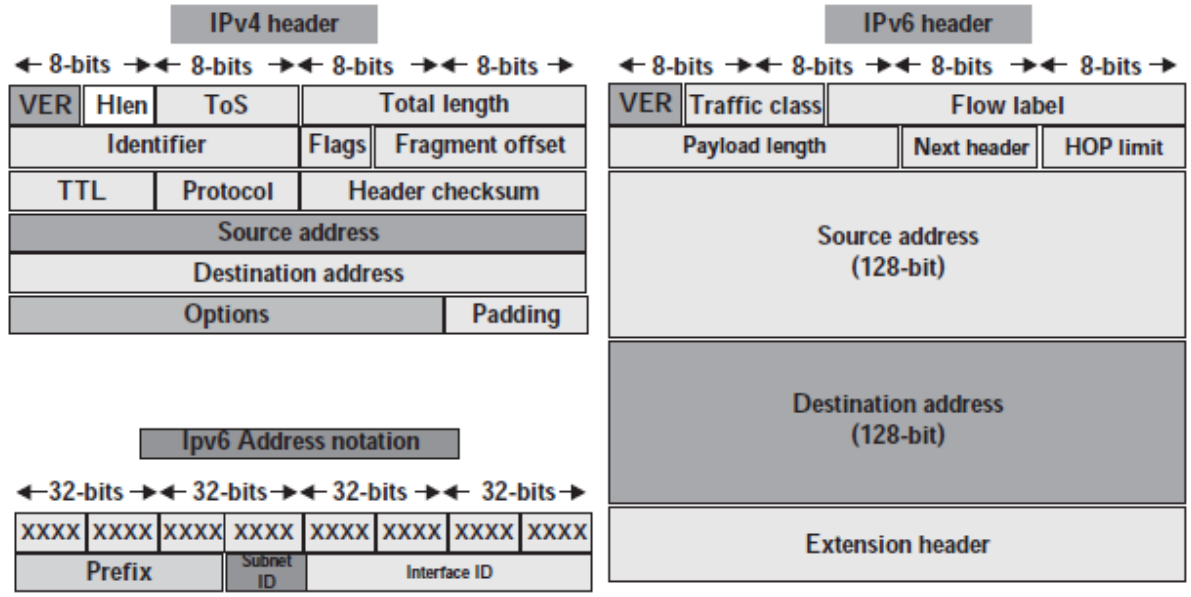
IPV6:

The IPv6 header packet shown in Figure has eight distinct fields, the functions of which are given as follows.
• **VER**: It is 4 bits long and represents the version of IP. It is 6 (binary: 0110).
• **TRAFFIC CLASS**: It is 8 bits long. The first six most significant bits represent the type of service to be provided to this packet (by the routers); explicit congestion notification (ECN) is handled by the last 2 bits.
• **FLOW LABEL**: It is 20 bits long and designed for streaming media or real time data. The FLOW LABEL allows for information flow ordering; it also avoids packet resequencing.
• **PAYLOAD LENGTH**: It is 16 bits long and provides a router with information about a packet's payload length or the amount of data contained in the packet's payload.
• **NEXT HEADER**: It is 8 bits long and informs the router about the type of extension header the packet is carrying. Some of the extension headers and their corresponding values are as follows: Hop-by-hop options header (0), routing header (43), fragment header (44), destination options header (60), authentication header (51), and encapsulating security payload header (50).
In case an extension header is absent, it represents the upper layer protocol data units (PDUs).
• **HOP LIMIT**: It is 8 bits long and prevents a packet from looping infinitely in the network. As it completes a link, the limit's value is decremented by one.
• **SOURCE ADDRESS**: It is 128 bits long and indicates the origin address of the packet.
• **DESTINATION ADDRESS**: It is 128 bits long and indicates the destination address of the packet.

The Internet Protocol Version 6 or IPv6 is a resultant of the developments on and beyond IPv4 due to fast depleting address ranges in IPv4. The IPv4 was not designed to handle the

needs of the future internet systems. The needs of massive scalability and limited resources gave rise to IPv6, which was developed by the IETF (Internet Engineering Task Force. IPv6 works on the OSI layer 3 (network layer). However, in contrast **to IPv4 (which is 32 bits long** and offers around 4,294,967,296 addresses), **IPv6** has a massive logical address range (which is **128 bits long**). Additional features in IPv6 include auto-configuration features, end-to-end connectivity, inbuilt security measures (IPSec), provision for faster routing, support for mobility, and many others.
These features not only make IPv6 practical for use in IoT but also makes it attractive for a majority of the present-day and upcoming IoT-based deployments. IPV6 it cannot be made to support IPv4 applications directly.

Figure shows the differences between IPv4 and IPv6 packet structures.



Important features of IPv6 are as follows:

(i) **Larger Addressing Range:** IPv6 has roughly four times more addressable bits than IPv4.

(ii) **Simplified Header Structure:** IPv6 header format is quite simple than IPv4. IPv6 header's increased size is mainly due to the increased number of bits needed for addressing purposes.

(iii) **End-to-End Connectivity:** Addressing allows packets from a source node to directly reach the destination node without the need for network address translations using IPv6.

(iv) **Auto-configuration:** The configuration of addresses is automatically done in IPv6.

(v) **Faster Packet Forwarding:** Routing decisions by a router are taken much faster by checking only the first few fields of the header.

(vi) **Inbuilt Security:** IPv6 supports inbuilt security mechanisms. IPv6 has security as an optional feature.

(vii) **Anycast Support :** Multiple networking interfaces are assigned the same IPv6 addresses globally; these addresses are known as anycast addresses. This mechanism enables routers to send packets to the nearest available destination during routing.

(viii) **Mobility Support**: The mobility support of IPv6 allows for mobile nodes to retain their IP addresses and remain connected, even while changing geographic areas of operation.

(ix) **Enhanced Priority Support**: The priority support system in IPv6 is entirely simplified as compared to IPv4. The use of traffic classes and flow labels determine the most efficient routing paths of packets for the routers.

(x) **Extensibility of Headers**: The options part of an IPv6 header can be extended by adding more information to it; it is not limited in size. Some applications may require quite a large options field, which may be comparable to the size of the packet itself.

RPL

RPL stands for routing protocol for low-power and lossy networks (LLN) and is designed for IPv6 routing. It follows a distance vector based routing mechanism. The protocol aims to achieve a **destination-oriented directed acyclic graph** (DODAG). The network DODAG is formed based on an objective function and a set of network metrics. The DODAG built by RPL is a **logical routing topology** which is built over a physical network. The logical topology is built using specific criteria set by network administrators. The most optimum path (best path) is calculated from the objective function, a set of metrics, and constraints. The **metrics in RPL** may be **expected transmission values (ETX), path latencies**, and others. Similarly, the **constraints** in RPL include encryption of links, **the presence of battery-operated nodes,** and others. In general, the metrics are either minimized or maximized, whereas the constraints need to be minimized. The **objective function** dictates **the rules for the formation of the DODAG.** Interestingly, in RPL, a single node in the mesh network may have multiple objective functions. The primary reason for this is attributed to the presence of different network traffic path quality requirements that need separate addressal within the same mesh network. Using RPL, a node within a network can simultaneously join more than one RPL instance (graphs). This enables RPL to support QoS-aware and constraint-based routing. An RPL node can also simultaneously take on multiple network roles: leaf node, router, and others. Figure below shows the RPL mechanism with different intra-mesh addressing arising due to different requirements of network and objective functions. The RPL border router, which is also the RPL root (in the illustrated figure), handles the intra-mesh addressing.

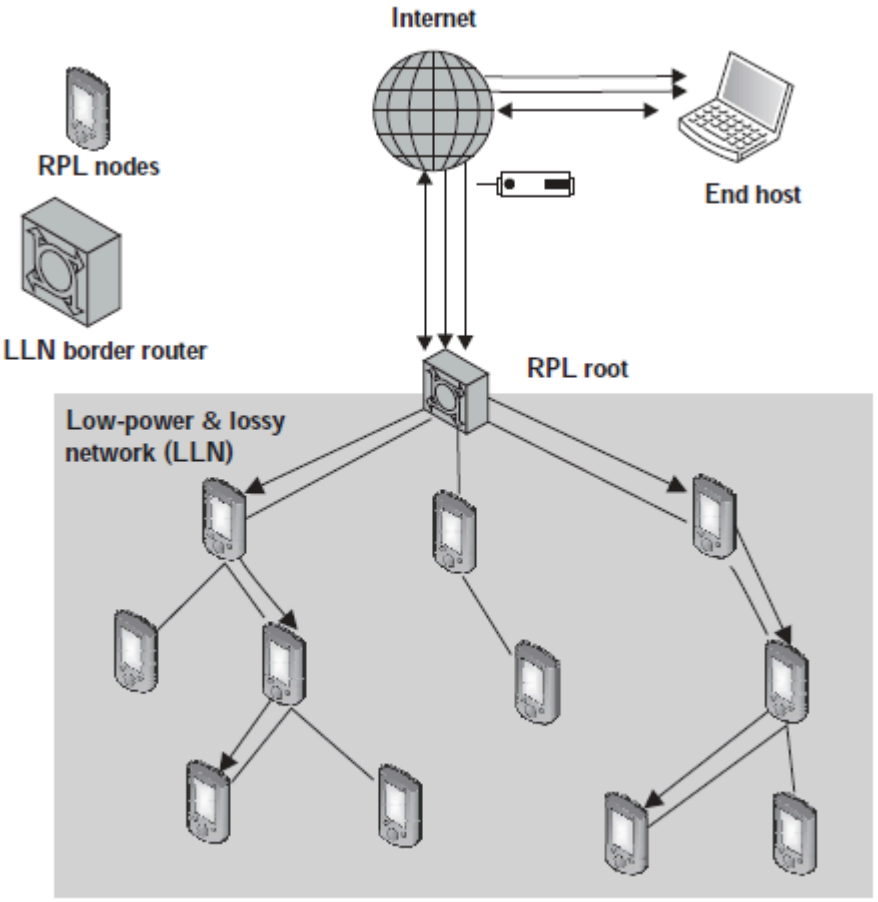


Figure: RPL information ow mechanism with different intra-mesh addressing and paths

RPL Instances

There are two instances associated with RPL: global and local. Global RPL instances are characterized by coordinated behavior and the possibility of the presence of more than one DODAG; they have a long lifetime. Local RPL instances are characterized by single DODAGs. The local RPL DODAG's root is associated directly with the DODAG-ID. The RPL instance ID is collaboratively and unilaterally allocated; it is divided between global and local RPL instances. Even the RPL control and data messages are tagged with their corresponding RPL instances using RPL instance IDs to avoid any ambiguity in operations.