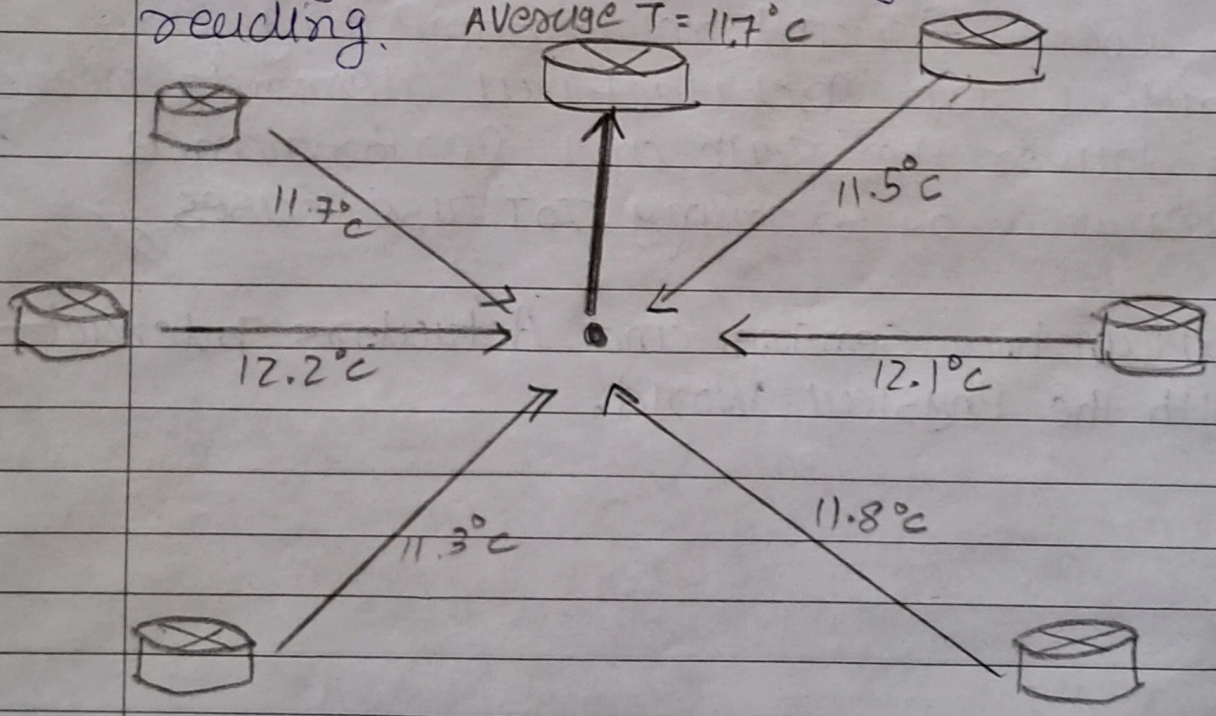
Assignment-2

CLASS ONE
Page No 1
Date

1) Explain the data aggregation in wireless Sensor Network.

Ans Hierarchy Provides the ability to aggregate similar sensor readings from sensor nodes that are in close Proximity to each other

- Data aggregation function in a WSN where temperature readings from a logical grouping of temperature sensors are aggregated as an average temperature reading.



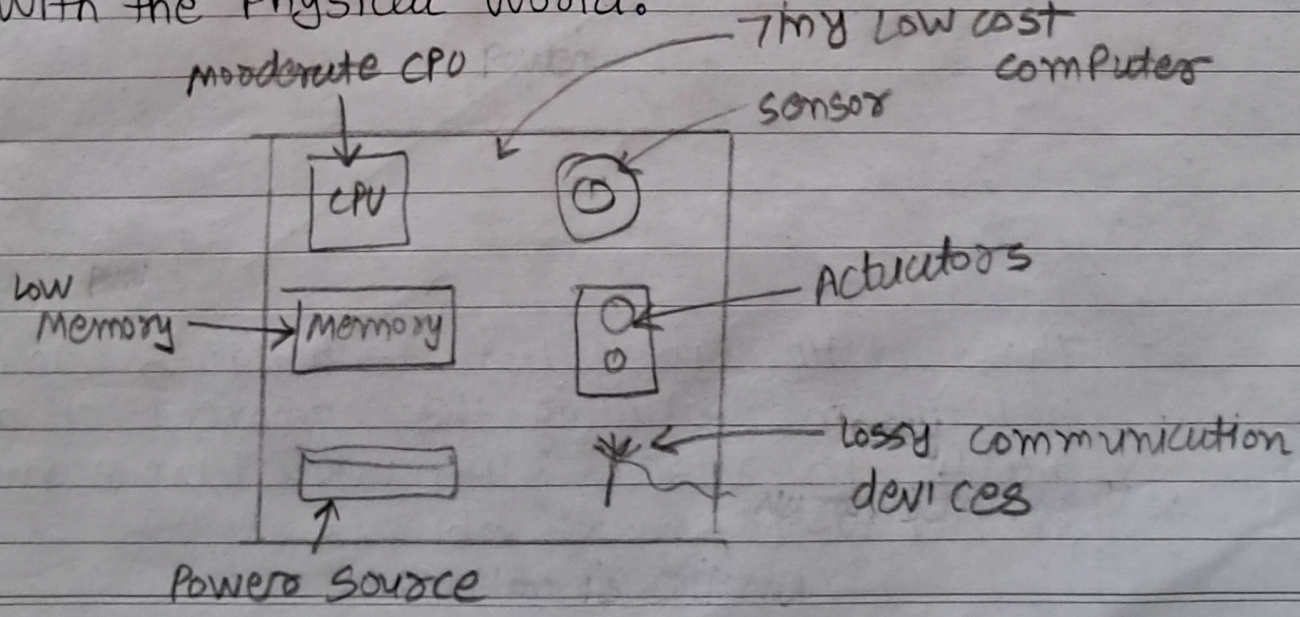
- These data aggregation techniques are helpful in reducing the amount of overall traffic in WSNs with very large numbers of deployed smart objects

- This data aggregation at the network edges is where fog and mist computing are critical IoT Architectural elements needed to delivers the scale and performance required by so many IoT use cases

- These data aggregation techniques are helpful in reducing the amount of overall traffic (and energy) in WSNs with very large numbers of deployed smart objects

- This data aggregation at the network edges is where fog and mist computing are critical IoT Architectural elements needed to deliver the scale and performance required by so many IoT use cases //

2) Explain how sensors and Actuators Interact with the Physical World.



- Actuators are natural complements to sensors

- figure demonstrates the symmetry and complementary nature of these two types of devices

- As discussed in the Previous section, sensors are designed to sense and measure practically and measurable variable in the Physical World

- They convert their measurements (typically analog) into electric signals or digital representations that can be consumed by an intelligent agent (a device or a human).

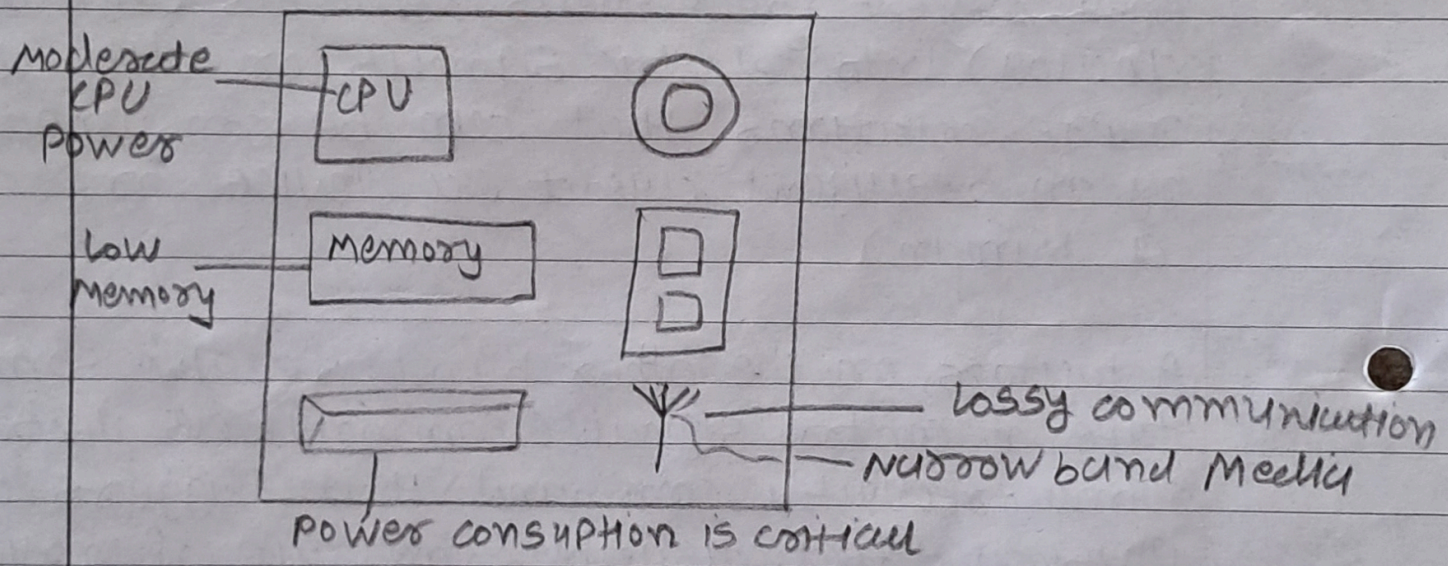
- Actuators, on the other hand, receive some type of control signal (commonly as electric signal or digital command) that triggers a Physical effect, usually some type of motion, force and so on. //

3) What are wireless sensor Networks (WSNs)? Explain.

Ans Wireless sensor networks are made up of wirelessly connected smart objects, which are sometimes referred to as motes

→ Advantages:-
- No infrastructure to consider with WSNs

→ DisAdvantages:-
- variety of design constraints



- The following are some of the most significant limitations of the smart object in WSNs:
- limited Processing Power
- limited memory
- Lossy communication
- limited transmission speeds
- limited Power

-> Smart object with limited Processing, memory, Power and so on are often referred to as constrained nodes

-> Wirelessly connected smart object have one of the following two communication Patterns:

-> Event-driven :-
- Transmission of sensory information is triggered only when a smart object detects a particular event or predetermined threshold

-> Periodic :-
- Transmission of sensory information occurs only at periodic intervals

-> Any communication Protocol must be able to scale to a large number of nodes

-> Sensors often Produce large amounts of sensing and measurement data that needs to be processed

-> This data can be processed locally by the nodes of a WSN or across zero or more hierarchical levels in IoT networks

CLASS ONE

Page No

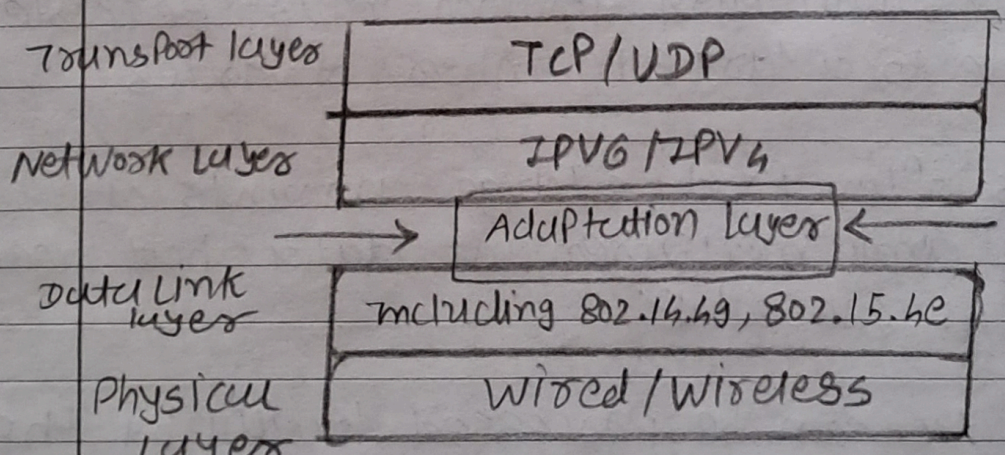
Date

4) Explain optimizing IP for IoT. Explain following term (i) Header (ii) Fragmentation (iii) Mesh Addressing.

Ans While the Internet Protocol is key for a sucessful Internet of things, constrained nodes and constrained networks mandate optimization at various layers and on mutiple protocols of the IP architecture

- The following section introduce some of these optimizations already available from the market or under development by the IETF.

- Figure highlights the TCP/IP layers where optimization is applied



C) Header :-

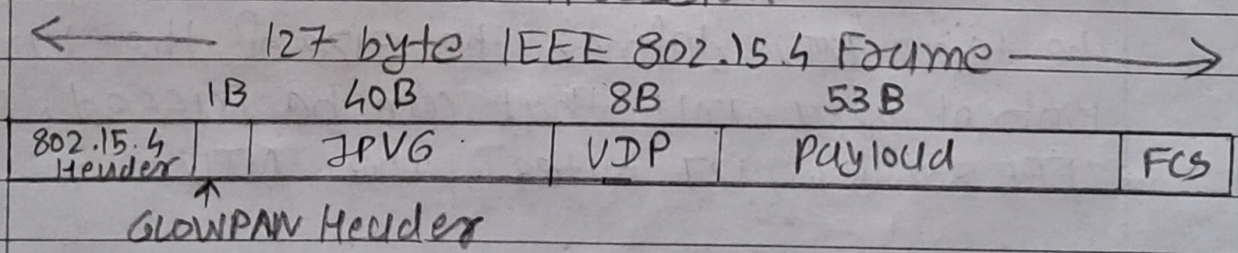
- Note that header compression for 6LOWPAN is only defined for an IPV6 header and not IPV4.

- 6LOWPAN header compression is stateless, and conceptually it is not too complicated.

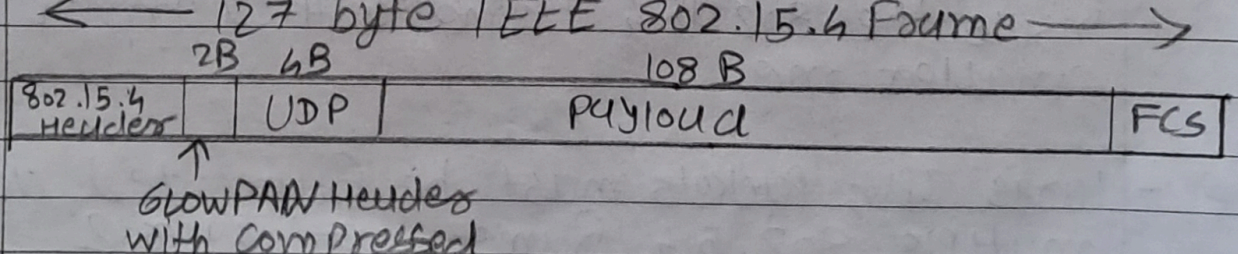
- At a high level, 6LOWPAN works by taking Adv. of shared information known by all nodes from their participation in the local network.

- Figure highlights an example that shows the amount of reduction that is possible with 6LOWPAN header compression

6LOWPAN without Header compression



6LOWPAN with IPV6 and UDP Header compression



- At the top of figure, you see a 6LOWPAN without any header compression enable. The 6LOWPAN header is only a single byte in this case

- The bottom half of figure shows a frame where header compression has been enabled for a best case scenario.

- Most importantly, the header compression has allowed the Payload to more double than double, from 53 bytes to 108 bytes, which is obviously much more efficient

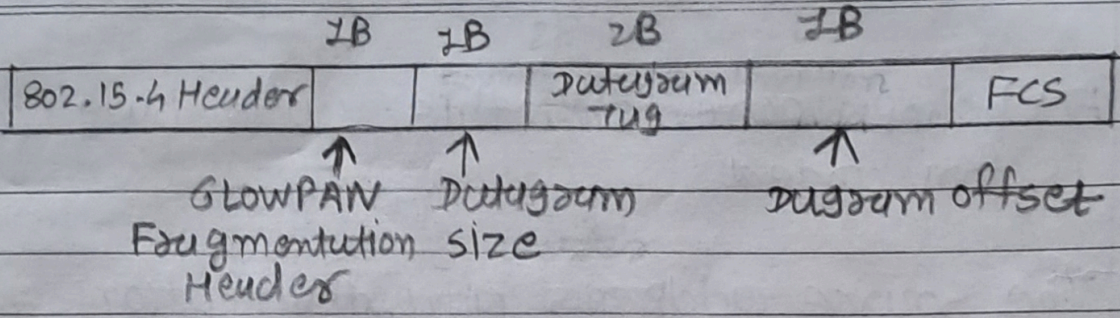
(ii) Fragmentation :-

- the maximum transmission unit (MTU) for an IPV6 network must be at least 1280 bytes

- the term MTU defines the size of the largest Protocol data unit that can be passed. For IEEE 802.15.4, 127 bytes is the MTU

- You can see that this is a Problem because IPV6, with a much larger MTU, is carried inside the 802.15.4 frame with a much smaller one.

- large IPV6 Packets must be fragmented across multiple 802.15.4 frames at layer 2



- In figure, the 6LOWPAN fragmentation header field itself uses a unique bit value to identify that the subsequent field behind it are fragment field as opposed to another capability, such as header compression.

- Also, in the first fragment, the datagram offset is not present because it would simply be set to 0.

(iii) Mesh addressing :-

- The purpose of the 6LOWPAN mesh addressing function is to forward packets over multiple hopes

- Three field are defines for this header:
  - Hop Limit
  - Source Address
  - destination Address

- the hop limit for mesh addressing also provides an upper limit on how many times the frame can be forwarded

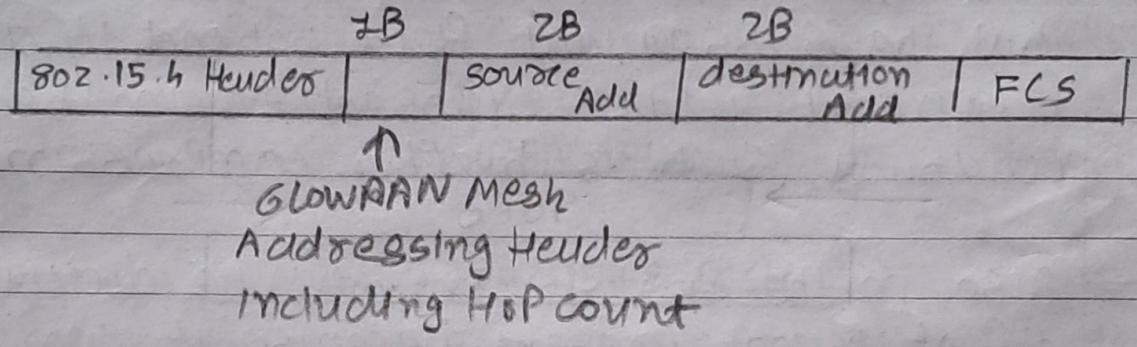
CLASS ONE

Page No

Date

- Each hop decrement this value by 1 as it is forwarded

- The source address and destination address field for mesh addressing are IEEE 802.15.4 addresses indicating the endpoints of an IP hop.



- Note that the mesh addressing header is used in a single IP subnet and is a Layer 2 type of routing known as mesh under.

5/08 Explain 6TISCH Scheduling management mechanism.

Ans Schedules in 6TISCH are broken down into cells

- A cell is simply a single element in the TSCH schedule that can be allocated for unidirectional or bidirectional communication between specific nodes

- Nodes only transmit when the schedule dictates that their cell is open for communication

- The 6TISCH architecture defines four Schedule management mechanism:

⇝ Static scheduling :-
- All nodes in the constructed network share a fixed Schedule

- cells are shared, and nodes contend for slot access in a slotted aloha manner

- Static scheduling is a simple scheduling mechanism that can be used upon initial implementation or as a fallback in the case of network malfunction

⇝ Neighbor - to - Neighbor scheduling :-
- A schedule is established that correlates with the observed number of transmission between nodes

- cells in this schedule ~~are~~ can be added or deleted as traffic requirements and bandwidth needs change

⇝ Remote monitoring and scheduling management :-
- Time slots and other resource allocation are handled by a management entity that can be multiple hops away.

- The scheduling mechanism leverages 6TiP and even CoAP in some scenarios.

- This scheduling mechanism provides quite a bit of flexibility and control in allocating cells for communication between nodes

⇝ Hop-by-Hop scheduling :-

- A node reserves a path to a destination node multiple hops away by requesting the allocation of cells in a schedule at each intermediate node hop in the path

- The protocol that is used by a node to trigger this scheduling mechanism is not defined at this point.

6) Explain RPL in details.

Ans In an RPL networks, each node acts as a router and becomes part of a mesh network. Routing is performed at the IP layer.

- Each node examines every received IPV6 packet and determines the next hop destination based on the information contained in the IPV6 header.

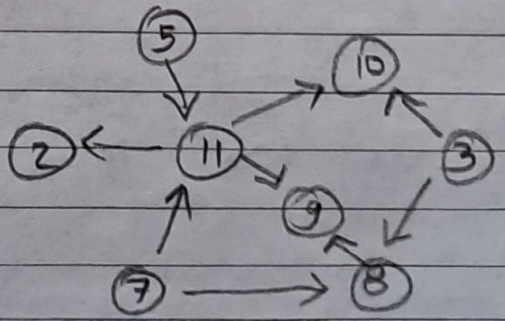
- No information from MAC layer header is needed to perform next hop determination.

- To cope with the constraints of computing and memory that are common characteristics of constrained nodes. the Protocol defines two modes :-

Storing mode :-
- All nodes contain the full routing table of the RPL domain

Non-storing mode :-
- Only the border routes of the RPL domain contain the full routing table.
- ~~Etc~~.
- All other nodes in the domain only maintain their list of Parents and use this as a list of default routes towards the border routes

- RPL is based on the concept of a directed acyclic graph (DAG). A DAG is a directed graph where no cycle exist.



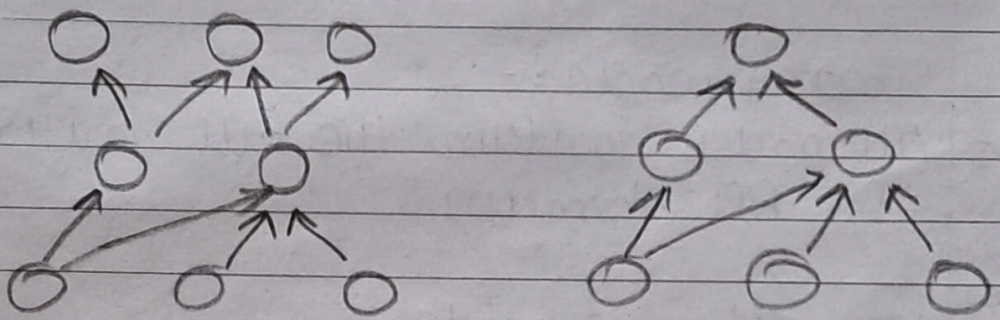
CLASS ONE

Page No

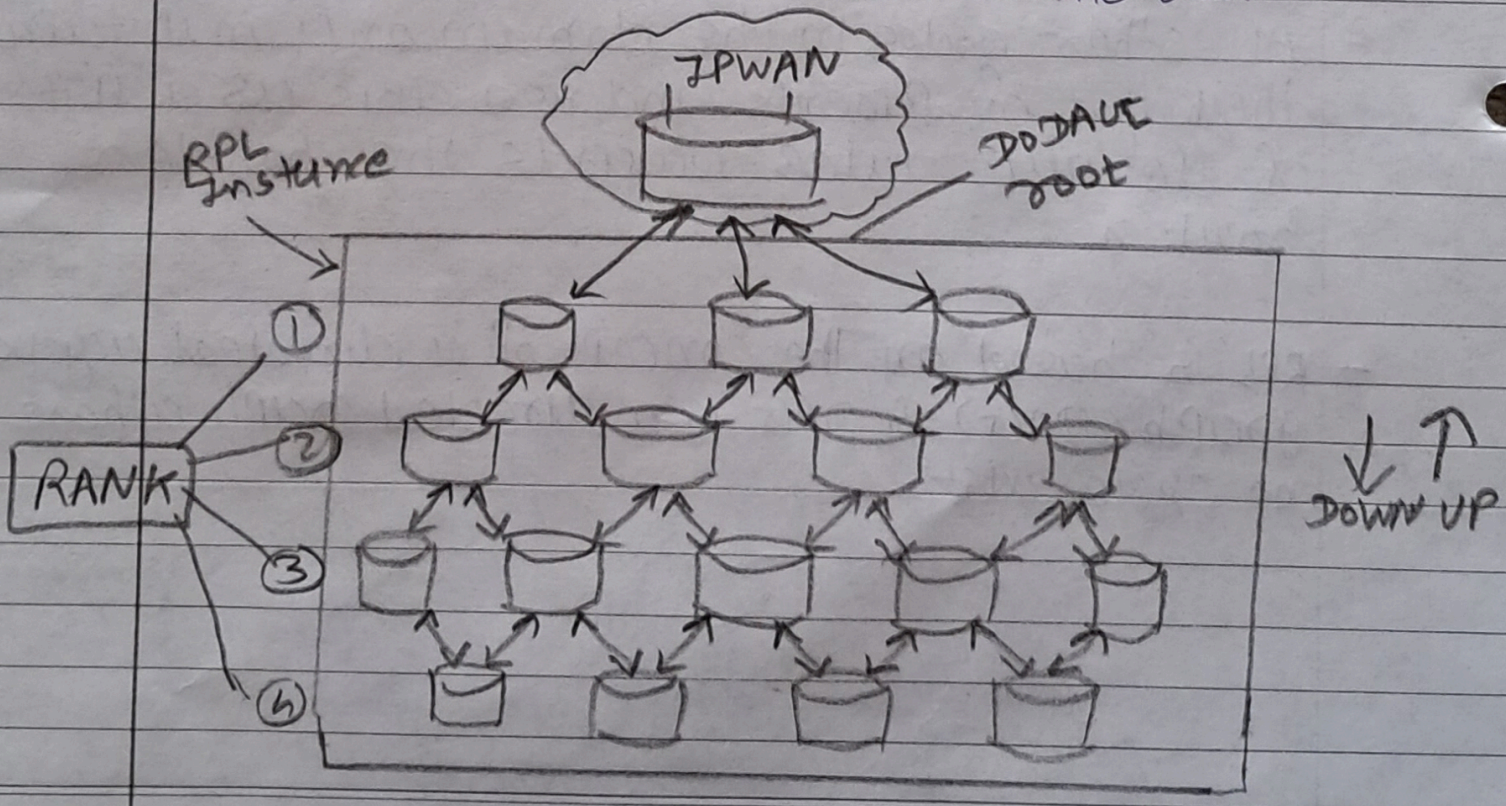
Date

- A basic RPL process involves building a destination oriented acyclic graph (DODAG). A DODAG is a DAG rooted to one destination

- figure compares a DAG and a DODAG. You can see that that a DAG has mutiple roots, whereas the DODAG has just one.



- In a DODAG, each node maintain up to three parents that provides path to the root



- The routing graph created by the set of DoDAG parent across all nodes defines the full set of upward routes.

- Upward routes in RPL are discovered and configured using DAG information object (DIO) messages. Nodes listen to DIOs to handle changes in the topology that can affect routing

- The information in DIO message determines parents and best path to the DoDAG root

- Nodes establish downward routes by advertising their parent set toward the DODAG root using a Destination Advertisement object (DAO) message

- DAO messages allow nodes to inform their parents of their presence and reachability to descendant

- RPL message such as DIO and DAO, run on top of IPV6

- These messages exchange and advertise downstream and upstream routing information between a border router and the nodes under it.

CLASS ONE

Page No

Date

- DAO and DIO messages move both up and down the DODAG, depending on the exact message type. //