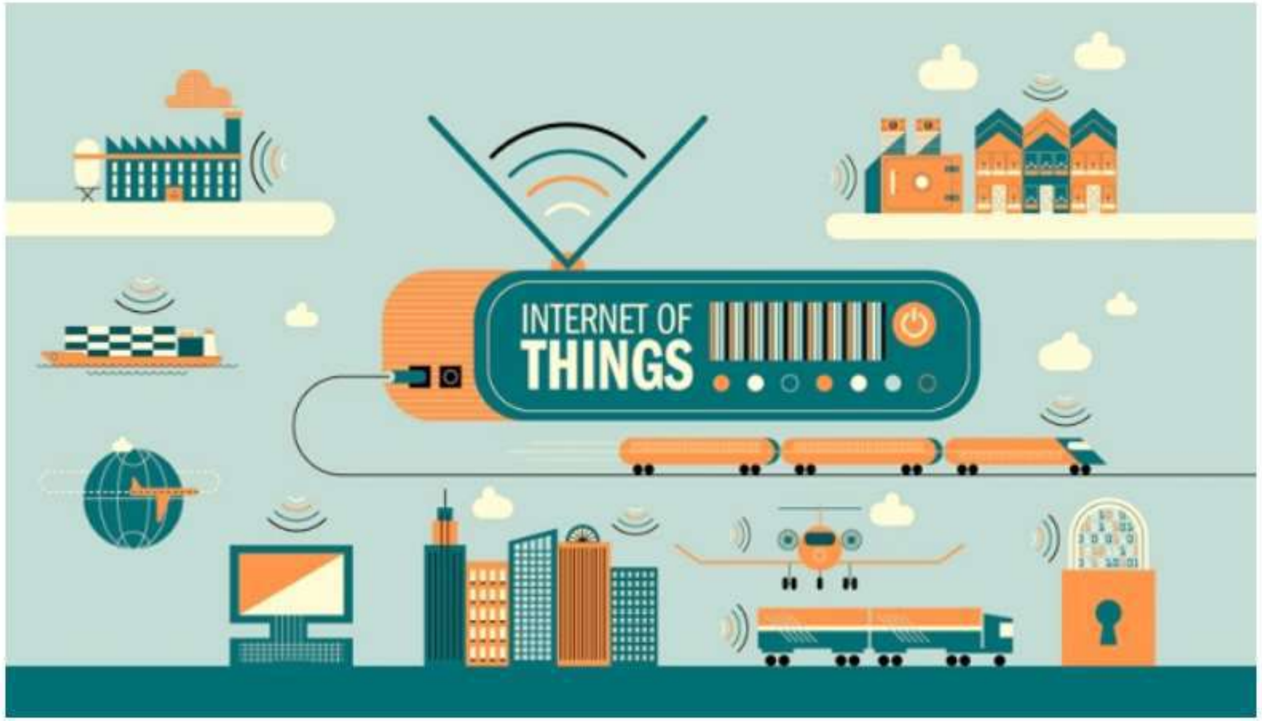
REVA UNIVERSITY



Prof Raghavendran T S
Asst Prof, School of C&IT, Reva University, Bengaluru
Raghavendran.ts@reva.edu.in

www.reva.edu.in

B22EK0601

IP as The IOT Network Layer

This unit talks about following topics as below

1) The Business Case for IP: This section discusses the advantages of IP from an IoT perspective and introduces the concepts of adoption and adaptation.

2) The Need for Optimization: This section dives into the challenges of constrained nodes and devices when deploying IP. This section also looks at the migration from IPv4 to IPv6 and how it affects IoT networks.

3) Optimizing IP for IoT: This section explores the common protocols and technolo gies in IoT networks utilizing IP, including 6LoWPAN, 6TiSCH, and RPL.

4)Profiles and Compliances: This section provides a summary of some of the most significant organizations and standards bodies involved with IP connectivity and IoT

Need for Optimization

Constrained Devices

- RFC 7228 defines three classes for constrained nodes: Class 0, 1, 2

	RAM	Flash Storage	IP stack	Security Scheme	Example
Class 0	< 10 KB	< 100 KB	Not present	No	Push button
Class 1	> 10 KB	> 100 KB	Optimized IP stack	Light	Sensors
Class 2	> 50 KB	> 250 KB	Full IP stack	Yes	Smart meter

Constrained networks are often referred to as low-power and lossy networks (LLNs). Lossy in this context refers to network unreliability that is caused by disruptions in the data flow or packet loss.

Challenges faced by Constrained Nodes in IP Deployment

- Limited Computational resources
- Communication of Bytes of Data Infrequently
- Low Power availability
- Network Congestion & latency
- Security vulnerabilities
- Scalability issues in large networks

Need for Optimization

Constrained Networks

Constrained networks are often referred to as low-power and lossy networks (LLNs). Lossy in this context refers to network unreliability that is caused by disruptions in the data flow or packet loss.

Challenges in Constrained Networks

- **Limited Resources** – Low power, processing, and memory constraints
- **High Latency & Packet Loss** – Weak wireless signals & interference
- **Security Risks** – Vulnerabilities due to lightweight protocols
- **Scalability Issues** – Handling an increasing number of IoT devices
- **Interoperability** – Compatibility issues between different network protocols
- **Energy Efficiency** – Optimizing battery life for long-term operation

Need for Optimization:- IPv6 over IPv4

- Large simple address (2^128 address space)
    - Network ID + Interface ID
    - Plenty of addresses; easy to allocate and manage
- Auto-configuration and Management
    - ICMPv6
- Integrated bootstrap and discovery
    - Neighbors, routers, DHCP
- Global scalability
    - 128 Bit Addressing = 3.4*10^38 unique addresses

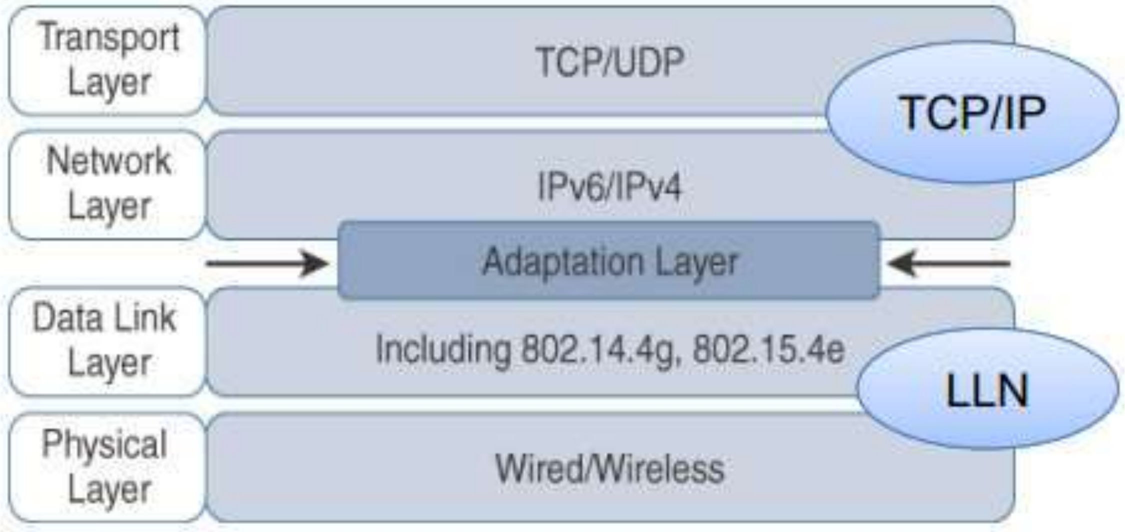
Optimizing IP:- 6LoWPAN

6LoWPAN: IPv6 over Low-power Wireless Personal Area Networks

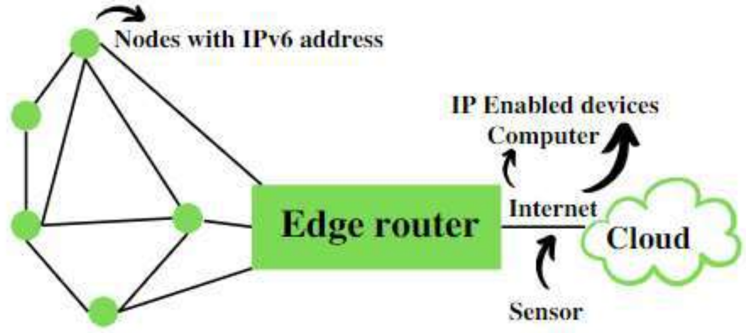
IETF formed 6LoWPAN WG in 2004 to design the Adaptation Layer

Primary goal of 6LoWPAN is even the smallest devices should have access to the IP.

6LoWPAN Applications: General Automation, Home automation, Smart Grid, Industrial monitoring, Smart Agriculture, etc.



6LoWPAN Architecture



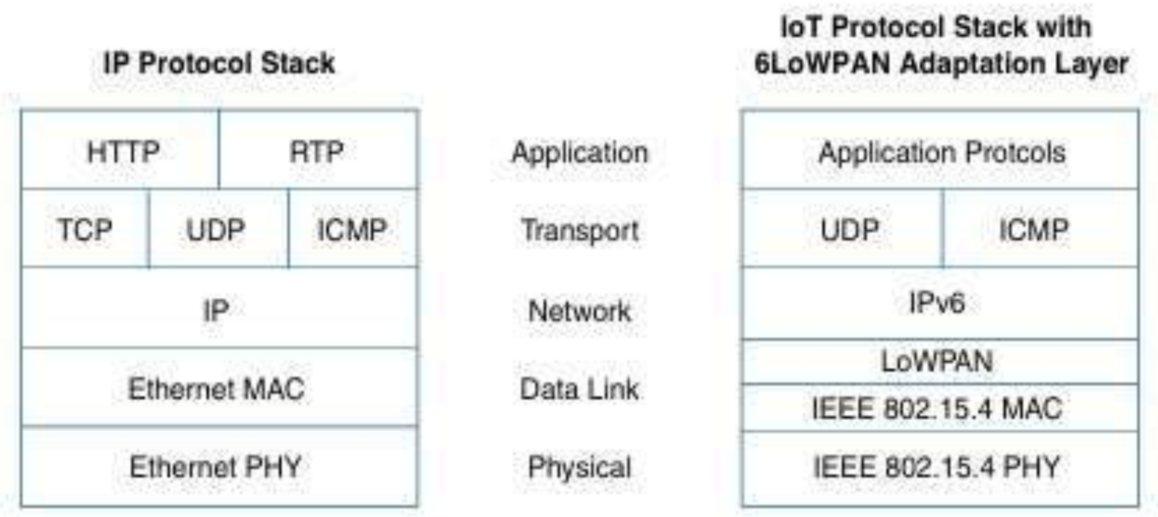
- It is a technology that makes the individual nodes IP-enabled.
- It only allows for the smaller devices with minimal processing ability to establish communication using one of the Internet Protocols.
- It comprises an Edge Router and Sensor Nodes. Even the smallest of the IoT devices can now be part of the network, and the information can be transmitted to the outside world as well. For example, LED Streetlights.

Basic Requirements of 6LoWPAN

- The device should be having sleep mode in order to support the battery saving.
- Minimal memory requirement.
- Routing overhead should be lowered.

Features of 6LoWPAN
It is used with IEEE 802.15,.4 in the 2.4 GHz band.
Outdoor range: ~200 m (maximum)
Data rate: 200kbps (maximum)
Maximum number of nodes: ~100

LOWPAN Protocol Stack



Layer	Protocols Used in LOWPAN Network	Protocols Used in Traditional Network
Application	CoAP, MQTT, HTTP	HTTP, HTTPS, FTP, SMTP
Transport	UDP, TCP (less common)	TCP, UDP
Network	IPv6, RPL (Routing Protocol for Low-Power Networks)	IPv4, IPv6, OSPF, BGP
Adaptation	6LoWPAN	
Link Layer	IEEE 802.15.4	Ethernet (802.3), Wi-Fi (802.11), Cellular (4G, 5G)
Physical	ISM Band (e.g., 2.4 GHz, Sub-GHz)	Fiber Optics, Copper, Wireless

Key Differences in LOPWAN & Traditional Protocol Stack

Feature	IoT Protocol Stack (6LoWPAN)	Conventional IP Protocol Stack
Use Case	Designed for low-power, constrained IoT devices	Designed for high-speed, general-purpose networking
Addressing	IPv6-based (with header compression)	IPv4/IPv6
Routing	RPL (Optimized for low-power networks)	OSPF, BGP, RIP (More resource-intensive)
Transport Layer	UDP (lightweight), TCP (less common)	TCP (reliable), UDP (low-latency)
Security	DTLS for CoAP, IEEE 802.15.4 security features	TLS/SSL for HTTP, IPsec for secure networking
Power Consumption	Optimized for low power	Higher power consumption
Data Rate	Typically < 250 kbps (IEEE 802.15.4)	Varies (10 Mbps – 100 Gbps)
Network Size	Optimized for small to medium-scale IoT networks	Scalable for large-scale global networks
Interoperability	Supports IoT-specific standards (CoAP, MQTT)	Standard internet protocols

Adaptation Layer

The adaptation layer in IoT is a network layer that handles compression and fragmentation of packets. It's used in resource-constrained environments

- ✓ **Header Compression** -> Compresses 40B IPv6 and 8B UDP headers
- ✓ **Fragmentation & Reassembly** -> when MTU of 802.15.4 and IPv6 does not match.
- ✓ **Stateless Autoconfiguration** -> Devices inside 6LoWPAN generate their own IPv6

6LoWPAN Advantages

Open IP Standard

- Use open standard such as TCP, UDP, HTTP, CoAP, MQTT, WebSocket
- End-to-End IP addressable nodes
- No gateway needed.
  - A router connects the 6LoWPAN network to IP

Mesh Routing

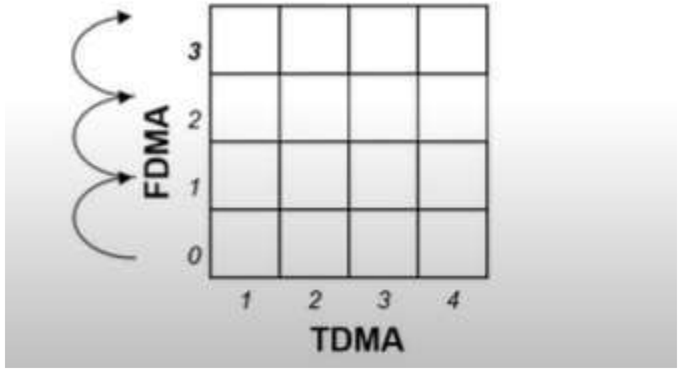
- One-to-many / many-to-one routing
- Robust and Scalable
- Self healing
- Flexible

Multiple PHY Support

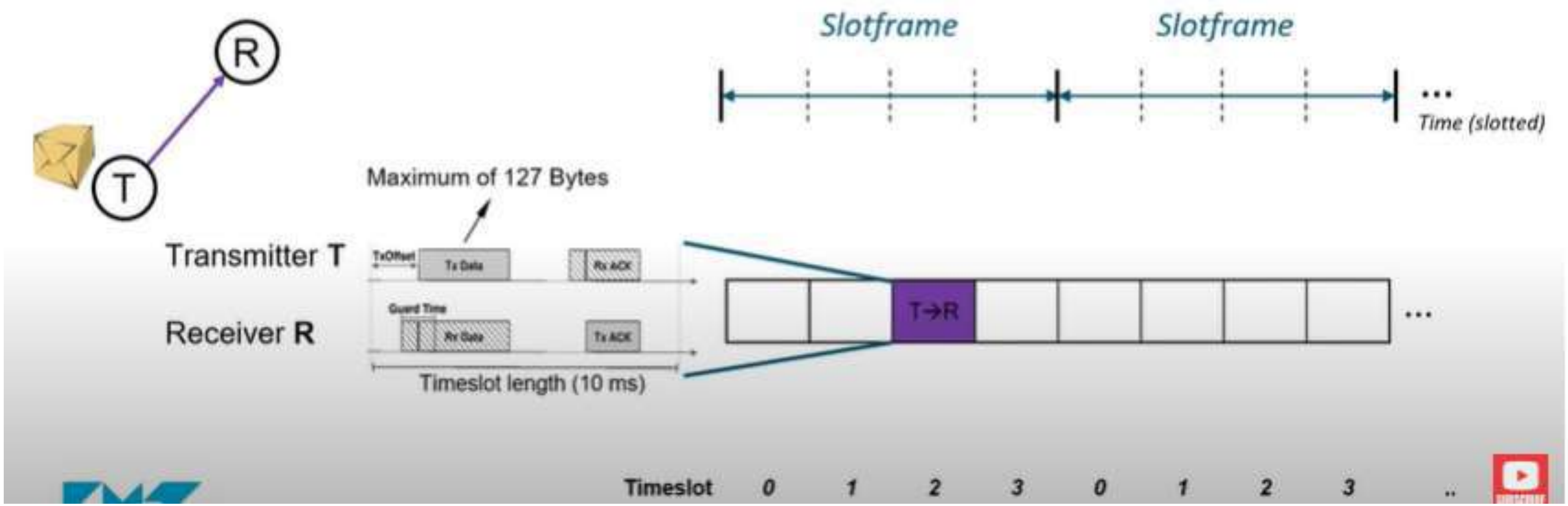
- Freedom of frequency band and physical layer
- Can be used across multiple communication platform
  - (ex. Ethernet / WiFi / 802.15.4 / Sub-1 GHz)
- Interoperability at the IP level

6TISCH (IPv6 over Time Slotted Channel Hopping)

- IEEE 802.15.4 Standard Targeting Low-Power, Low Data Rate & low-cost Wireless Meshes.
- It is a Layer 2 Protocol mainly working in MAC Layer of Data Link Layer.
- TSCH is designed for Reliable and deterministic communication & for low power operation.
- It is essentially a combination of TDMA & FDMA with Radio Hopping Channel Technique.



Time Slot & Slot Frame in TISCH MAC Protocol

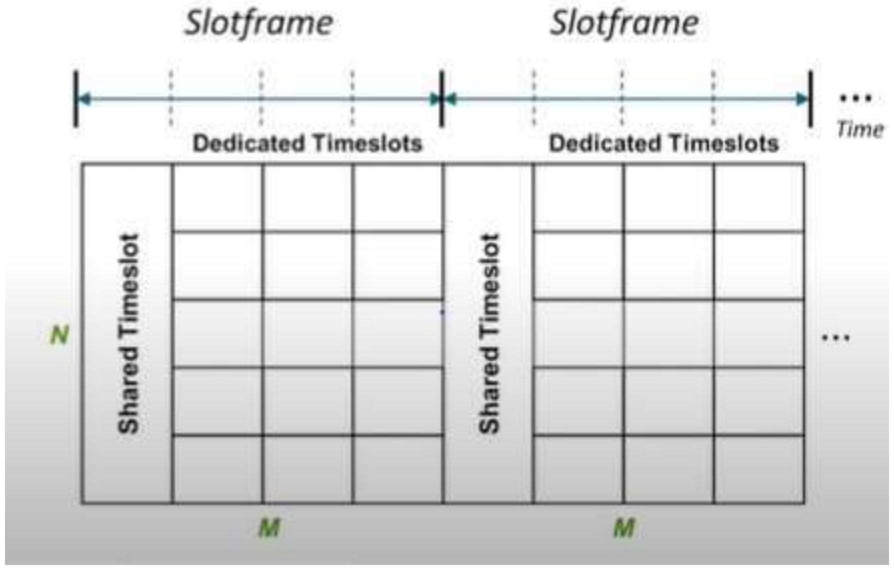


Schedule in TiSCH

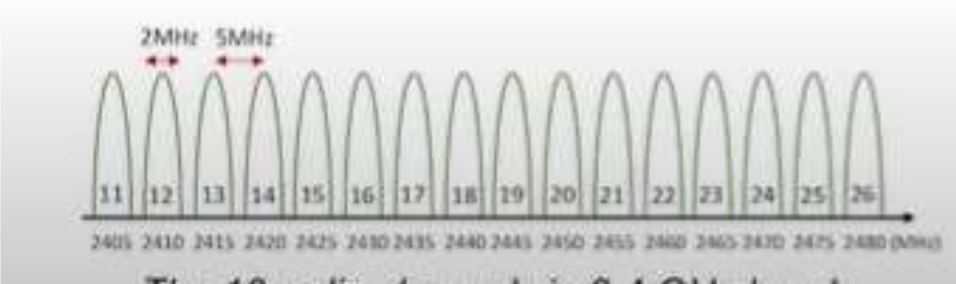
- A Schedule Orchestrates all possible combinations of the node with its neighbors and it is managed by Scheduling function. Indeed every actions on the node on each time slot determines the schedule of that node.

- At Each Time Slot each node knows if:-

-> It has the right to transmit a frame and to whom
-> It must stay “awake” to receive a frame.
-> It can sleep to save energy.



Schedule can be represented as M X N Matrix where M represents length of time slots and N Represents number of Radio channels to Hop.



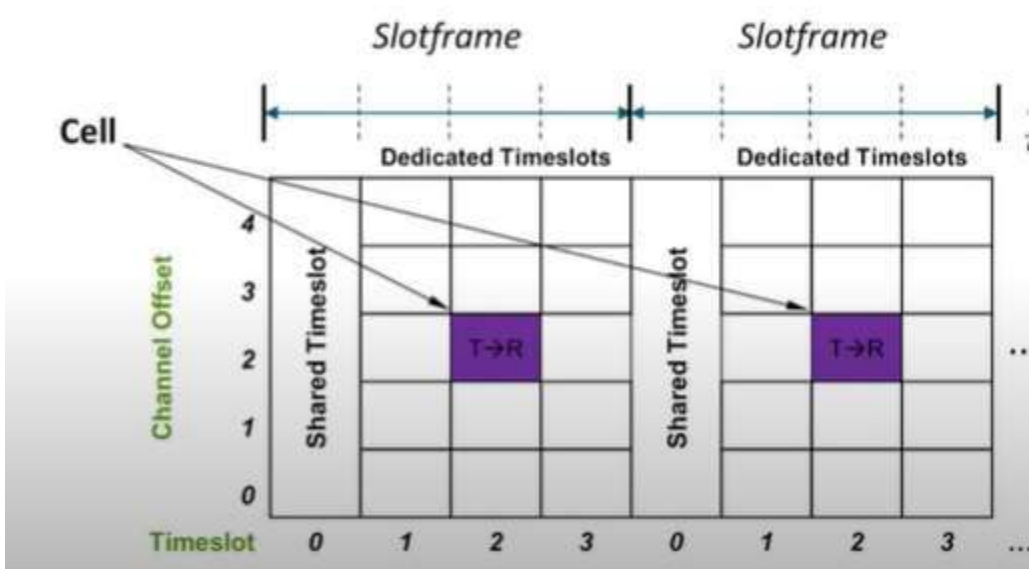
Channel Bandwidth – 2MHz
Channel Width – 5MHz
Number of Channels- 16

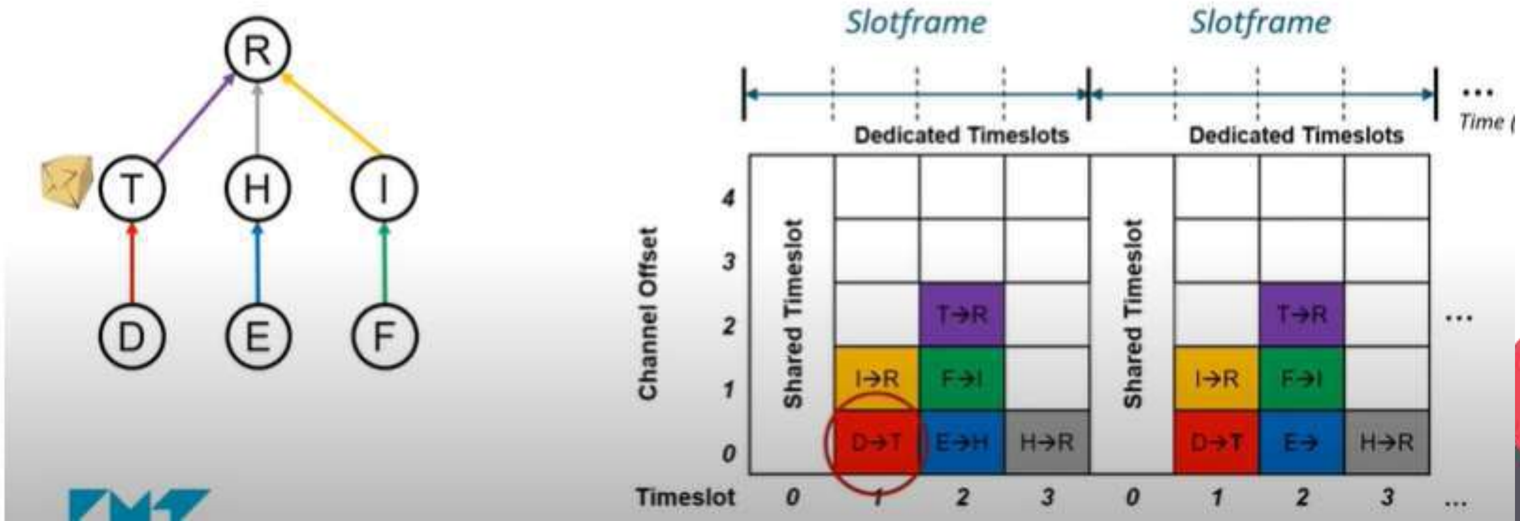
Cont’d

- An element is Scheduled Matrix is called a cell defined by Channel offset & Timeslot Pair.

-> Timeslot offset:- Determines Timslot position in Time domain.

-> Channel Offset:- Is an index which maps to Frequency or Radio channel that a node should tune to its radio receiver.





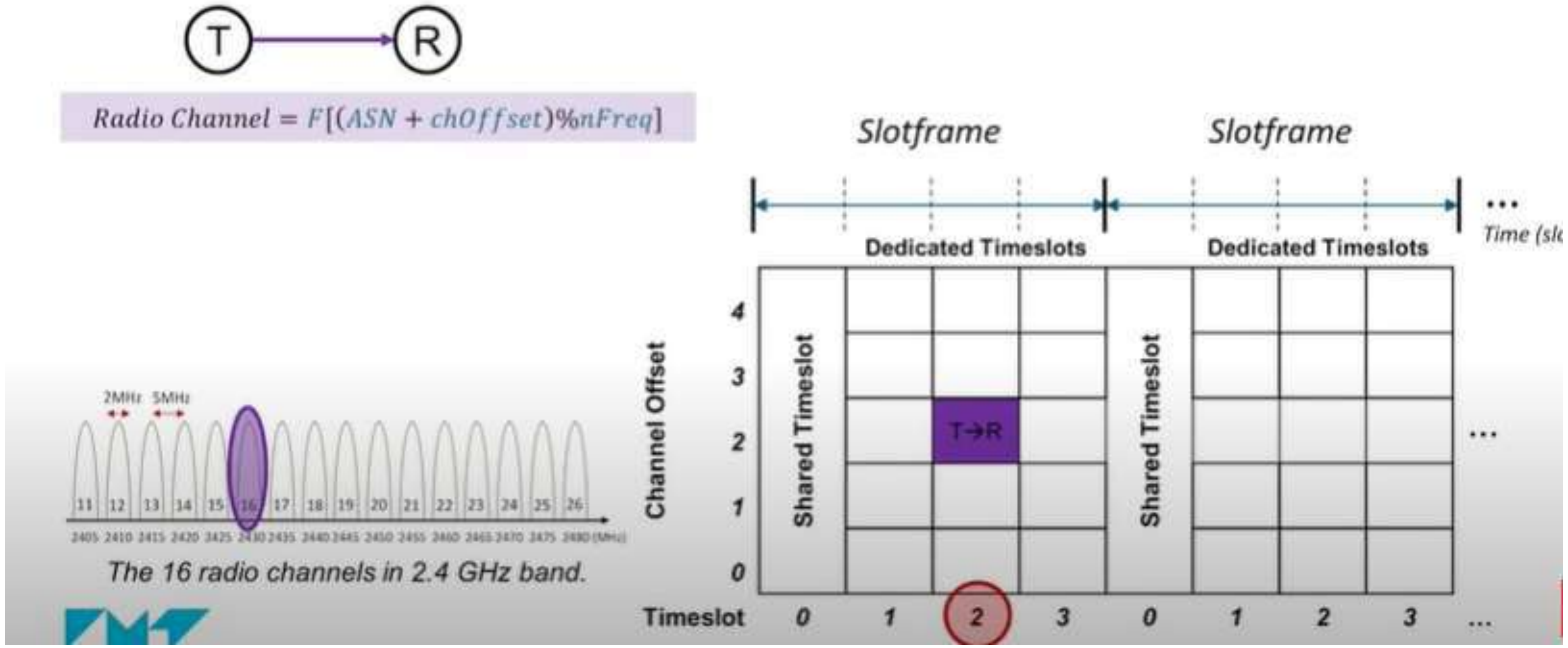
CHS (Channel Hopping Sequence)

- This Approach provides increased Reliability to Multiple Fading & External interference. It is handled through CHS, at each timeslot a Radio channel used by radio is computed using following equation.

ASN- Absolute Slot number represents number of time slots elapsed since network started.

nFreq:- Number of Available Phyical Radio channels

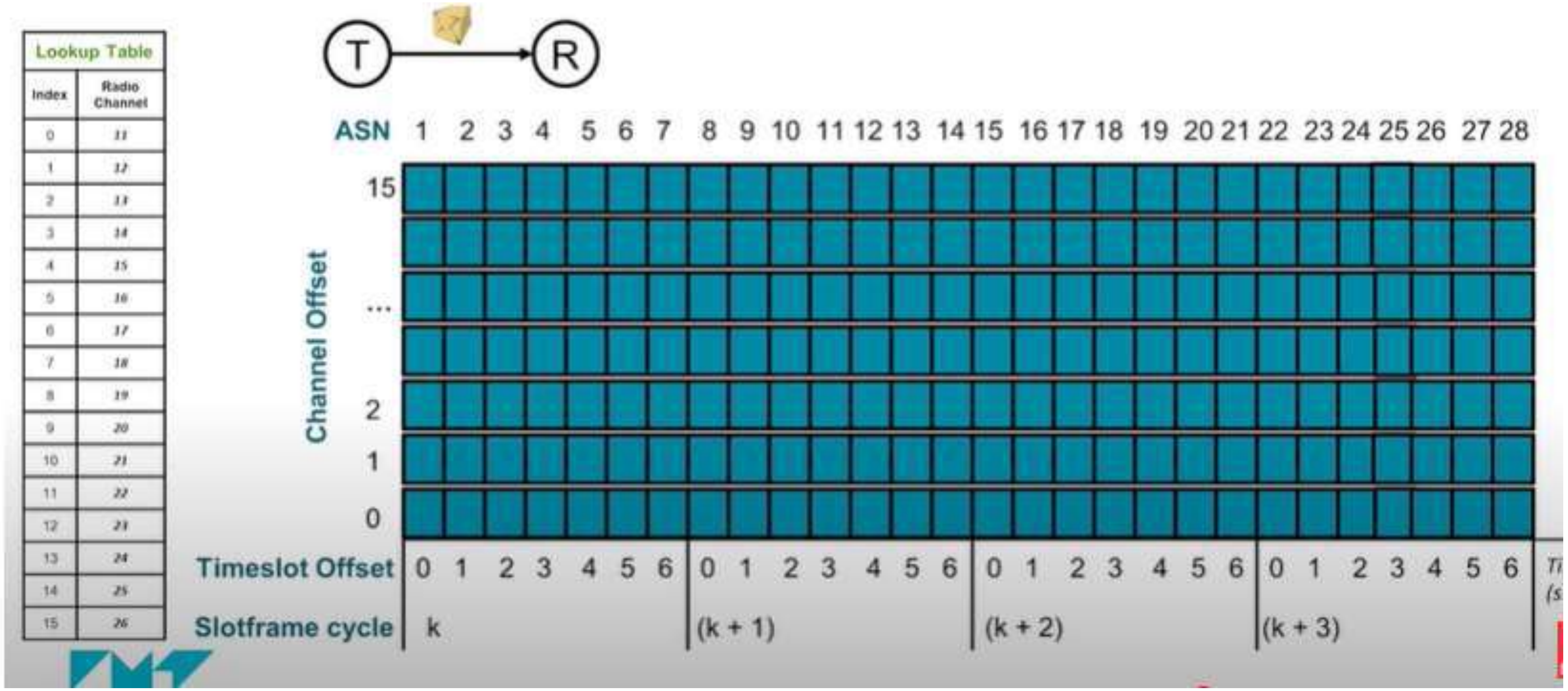
F:- Look up table, A Function



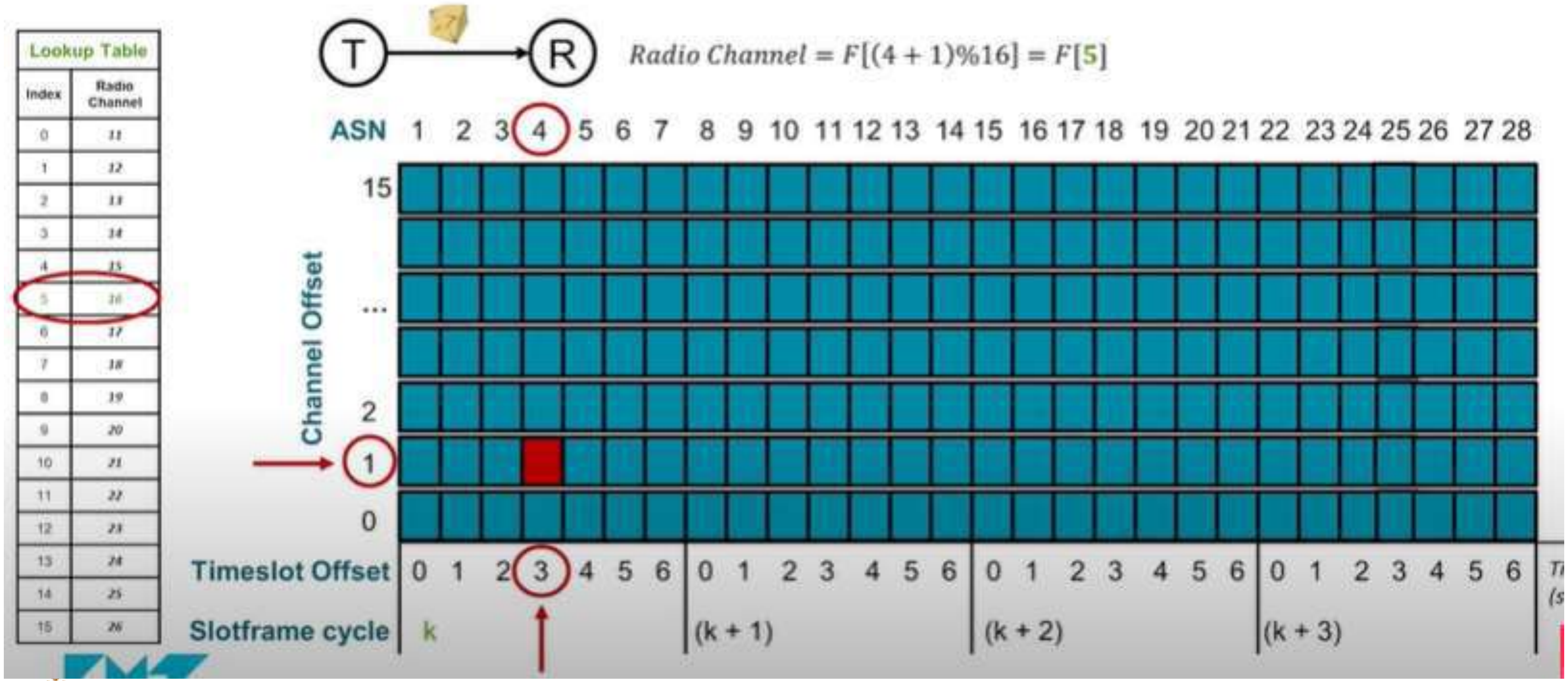
Example of CHS Operation

Lets Assume Node T as Transmitter & Node R as Receiver

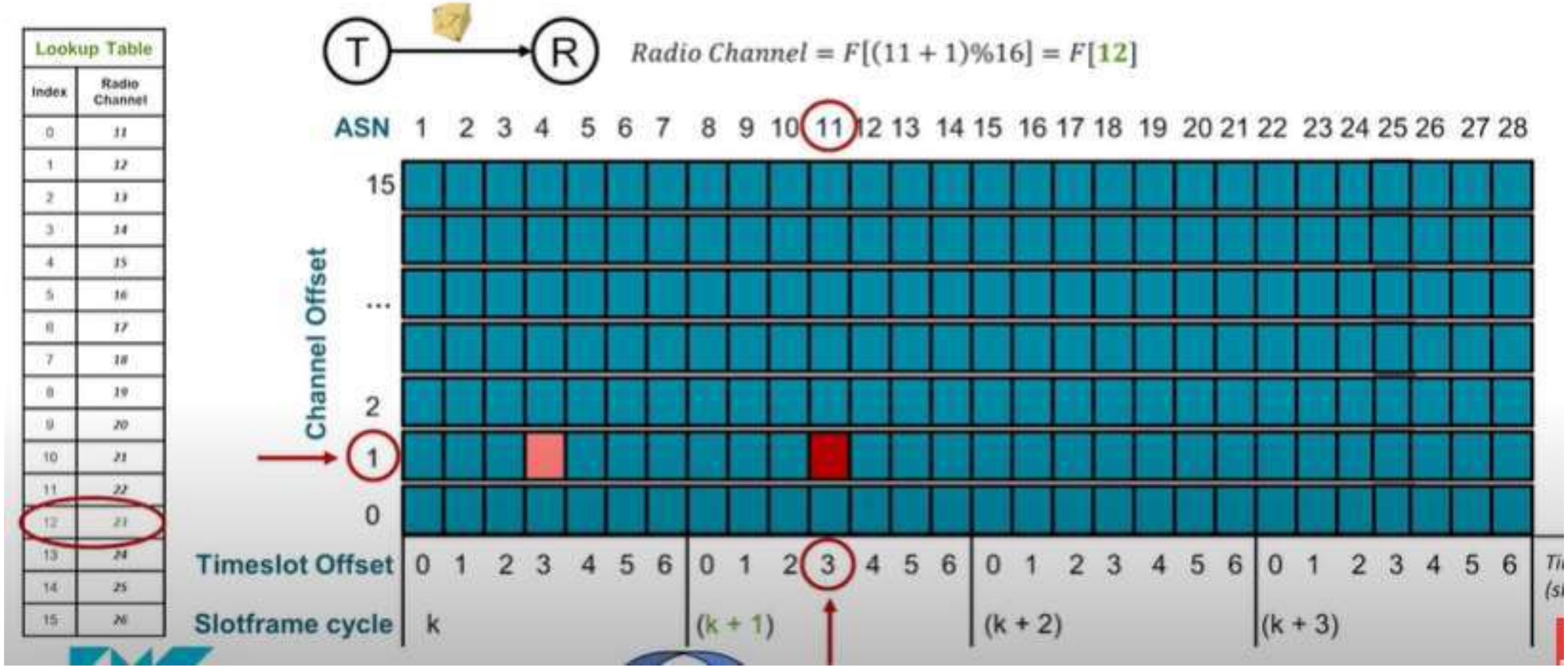
- X Axis- Timeslot offset- 1 Slot Frame (k) has 7 Timeslots
- Y Axis- Channel offset- Represents 16 radio channels.



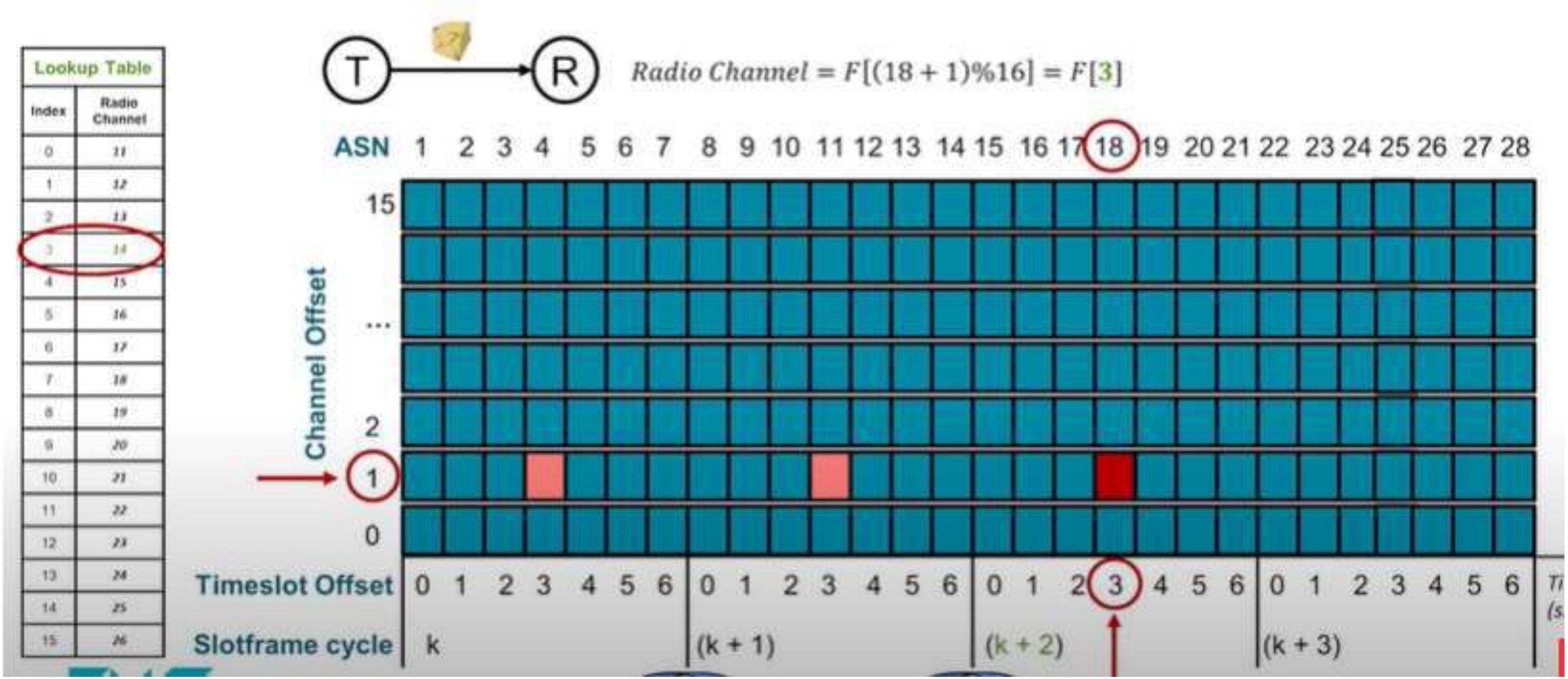
Find Radio Channel for 3rd Time slot of 1st Slot Frame (k) with channel offset of 1, Find out ASN, Compute Radio Channel as a Function and find the corresponding function value as per look up table



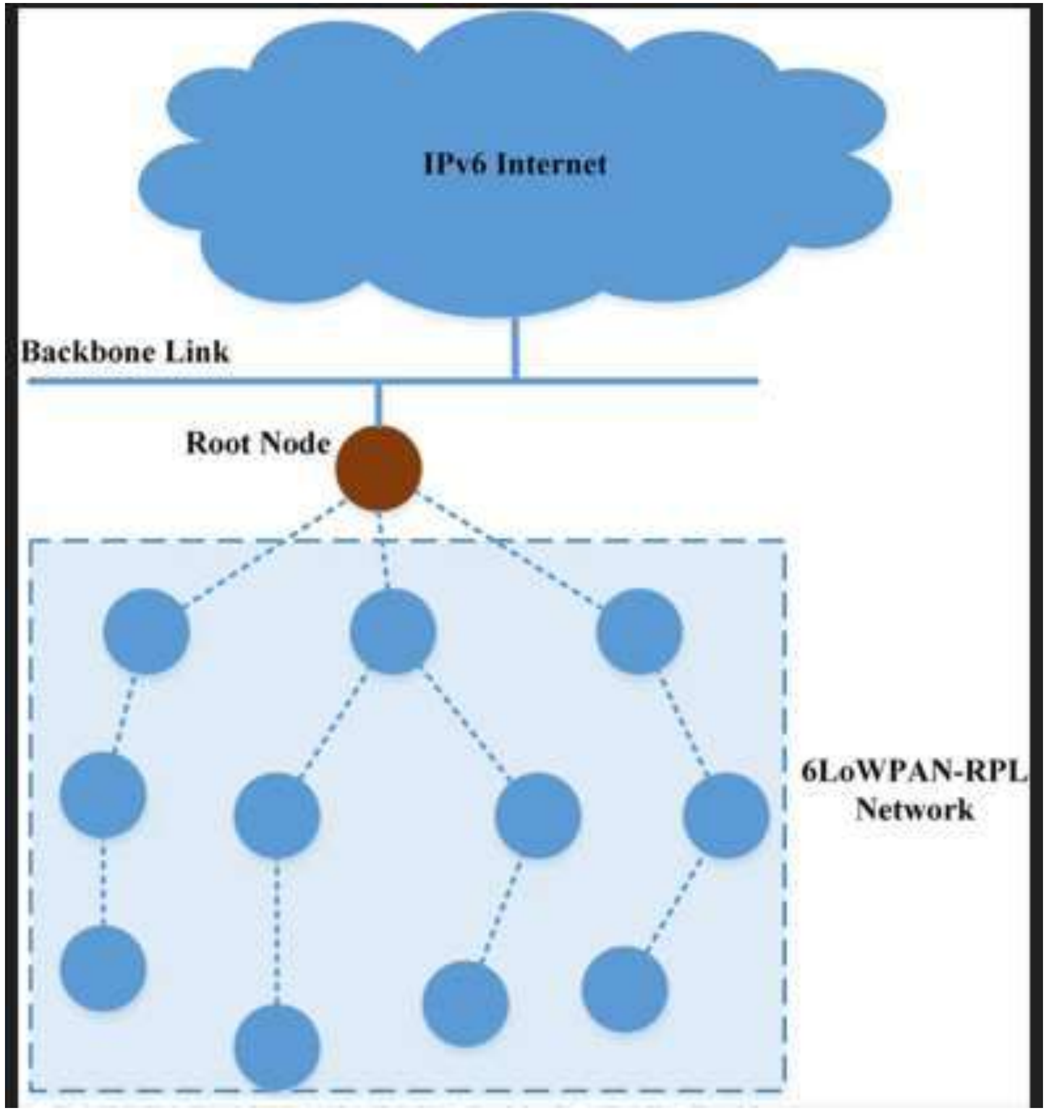
Find Radio Channel for 3rd Time slot of 2nd Slot Frame (K+1) with channel offset of 1, Find out ASN, Compute Radio Channel as a Function and find the corresponding function value as per look up table.



Find Radio channel Number for 3rd Slot frame, with Timeslot offset of 3 & Channel offset 1.



RPL: the IPv6 Routing Protocol for Low-power and Lossy Networks



What is RPL?

- IPv6 Routing Protocol for Low-Power and Lossy Networks (LLNs)
- Designed for IoT, wireless sensor networks, and industrial automation

Key Features of RPL

•Supports different traffic patterns: Point-to-Point (P2P), Point-to-Multipoint (P2MP), Multipoint-to-Point (MP2P)
•Uses **DODAG (Destination-Oriented Directed Acyclic Graph)**
•Supports multiple **RPL Instances** for different applications
•Provides **Loop Avoidance & Recovery**

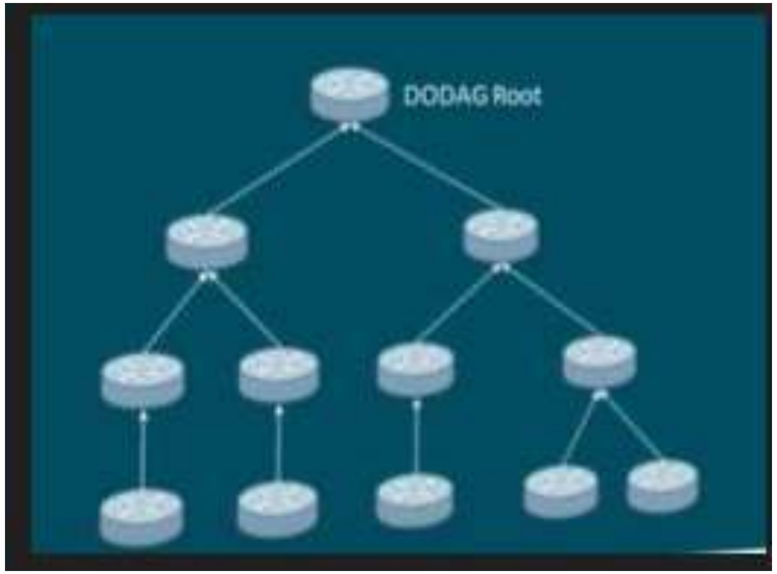
DODAG (Destination-Oriented Directed Acyclic Graph)

- A Tree Like structure where nodes route traffic toward the Root (DODAG Root).

-> DODAG Root:- The Main node, often a gateway.

->Rank:- Defines a node's position relative to the root (Node's distance from Root)

-> Parents & Children:- Nodes form Hierarchical Relationships



DODAG Flow

1) DODAG Root Initialization (Gateway Setup)

- The **Gateway (Root Node)** starts the network and assigns itself **Rank 1**.
- It broadcasts **DIO (DODAG Information Object) messages** to inform nearby nodes about its presence.

2) Node Rank Calculation & Parent Selection

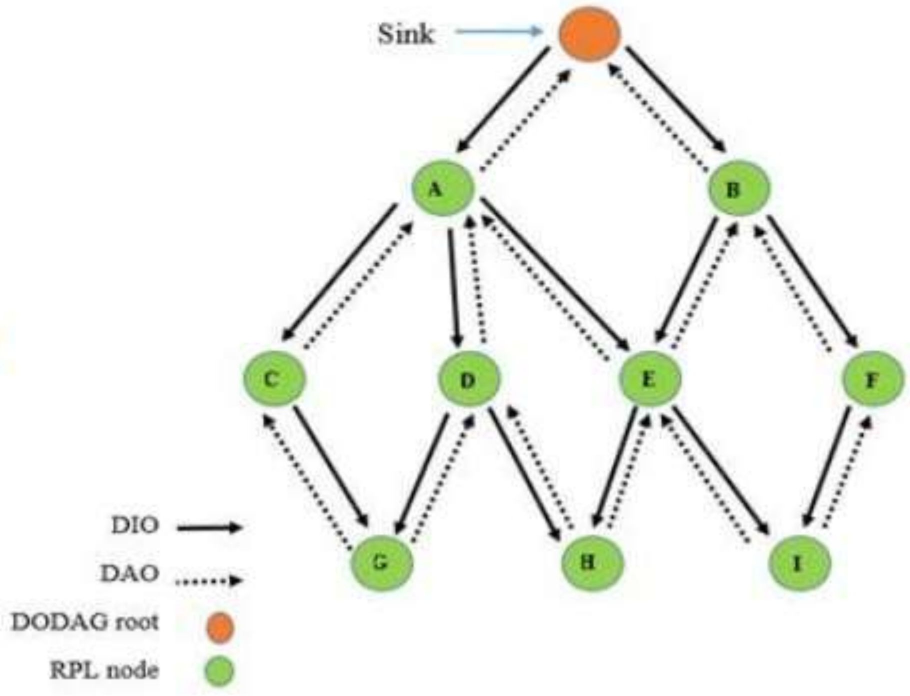
- Nodes receiving a DIO message calculate their **rank** (distance from the root).
- Nodes **select the best parent** based on:
  - **Lowest rank** (shortest path to the root).
  - **High link quality** (RSSI, ETX).
  - **Remaining energy** (optional for energy-aware networks).

3) Child Nodes Join the DODAG:- Other nodes repeat the process, selecting their parents and forming a tree-like structure.

DODAG Flow

3) Child Nodes Join the DODAG:-

- Use **DAO (Destination Advertisement object)** message to establish downward routes.
- DAO messages helps to advertise reachable destinations to their parents.
- This Downward Routes Enable Bi-directional communication (not just root-to-node, but also node-to-root).



DODAG Flow

4) Data Transmission Towards the Root

- Each node **forwards sensor data** towards its parent.
- The process repeats until the data **reaches the gateway**.

5) Network Maintenance & Self-Healing

- Nodes continuously monitor connectivity.
- If a parent node fails, child nodes **reselect a new parent** from available neighbors.
- New nodes can **join the network dynamically** by listening to DIO messages

Dodag Network Layout

```
Gateway (Root, Rank 1)
     /      |      \
  N1 (2)  N2 (2)  N3 (2)
  |        |       |
N4 (3)    N5 (3)  N6 (3)
```

- **DIO (DODAG Information Object)**:

The Gateway (Rank 1) sends **DIO messages** to Nodes **N1, N2, and N3 (Rank 2)**.
These Rank 2 nodes propagate the **DIO** messages downward to their respective child nodes (**N4, N5, and N6 (Rank 3)**).

- **DAO (Destination Advertisement Object)**:

Nodes **N4, N5, and N6 (Rank 3)** send **DAO messages** upward to their respective parent nodes (**N1, N2, and N3 (Rank 2)**).
Rank 2 nodes aggregate and forward **DAO** messages to the **Gateway**.

Problem Statement

- A smart farm deploys **10 sensor nodes** (temperature, humidity, and soil moisture sensors) across a large field. These nodes need to send data to a **gateway** for further processing and cloud storage. However:
- Some nodes are far from the gateway and cannot communicate directly.
- Nodes have limited battery life and should use energy-efficient paths.
- The network should automatically reconfigure if a node fails.

Solution using DODAG

1) Network Setup:

- The **gateway (root)** initializes the network.
- Each **sensor node** joins the network and selects a parent node based on the **best path** (low rank, high link quality).

2) Routing Formation:

- The gateway **sends DIO (DODAG Information Object) messages** to announce its rank (lowest rank = best).
- Nodes receive these messages and select the best **parent** based on:
    - Link quality (ETX, RSSI)
    - Hop count
    - Remaining energy

Cont’d

3) Nodes further send **DAO (Destination Advertisement Object) messages** to confirm their routing.

4) Data Transmission:

- Sensors forward data through multi-hop routing to the gateway.

If a parent node fails, its child nodes reselect a new parent.

Thank You